Arctic HARE: A Machine Learning-based System for Performance Analysis of Cross-country Skiers

Tor-Arne S. Nordmo[1], Michael A. Riegler[2,1], Håvard D. Johansen[1], and Dag Johansen[1]

[1] UiT: The Arctic University of Norway
[2] SimulaMet, Norway
[3] University of Oslo, Norway
tor-arne.s.nordmo@uit.no

Abstract. Advances in sensor technology and big data processing enable new and improved performance analysis of sport athletes. With the increase in data variety and volume, both from on-body sensors and cameras, it has become possible to quantify the specific movement patterns that make a good athlete.

This paper describes *Arctic Human Activity Recognition on the Edge* (Arctic HARE): a skiing-technique training system that captures movement of skiers to match those against optimal patterns in well-known cross-country techniques. Arctic HARE uses on-body sensors in combination with stationary cameras to capture movement of the skier, and provides classification of the perceived technique. We explore and compare two approaches for classifying data, and determine optimal representations that embody the movement of the skier. We achieve higher than 96 % accuracy for real-time classification of cross-country techniques.

Keywords: Machine Learning · Activity Recognition · Distributed Systems · Embedded Systems · Ski Technique Classification

1 Introduction

Analyzing the performance of athletes is becoming practically feasible with the growth of technology. Performance analysis of athletes is the act of quantifying sports performance in order to develop an understanding that can inform the conscious or unconscious choices done by the athlete in order to enhance their performance [15]. Multiple products allow coaches and athletes to review their performance manually [23, 19]. Physiological data is also often obtained through on-body sensors [7, 4].

Endurance and technique are related aspects that are significant factors for quantifying the performance of an athlete. It is important to make systems that can be used outside of the laboratory environment in order to realistically capture these aspects. Real-time analysis can provide immediate feedback to an athlete while training. This allows them to quickly adjust their movement to improve their performance. Sports produce very large amounts of data that can be analyzed, however it can be difficult to make general analysis software due to the differences between the sports and differences between data sources.

We have developed a distributed system that can perform real-time performance analysis on the edge using on-body sensors. We have also explored a video-based approach to compare with the sensor-based approach. We apply this system to the field of cross-country skiing. This system allows ski athletes to receive real-time feedback while they are training.

The remainder of this paper is structured as follows: In section 2 we give a description of *Arctic HARE* and its subcomponents. In section 3 we briefly explain the acquisition and preprocessing of the data used. Then, section 4 provides an experimental evaluation of the system with a brief discussion of the results. Then, we give an overview of related work in section 5. Finally, we conclude the paper in section 6.

2 Related Work

Øyvind Gløersen and Gilgien [5] automatically detected cycle length (how far the skier moves during a cycle), cycle duration and sub-techniques (Offset/V1 skate (V1), V2/One skate (1SK), and V2 Alt/Two skate (2SK)) using Differential Global Navigation Satellite System (dGNSS) measurements of the head of the skier. They based the cycle on the lateral velocity of the skier. They achieved an accuracy between 98 % and 100 %, depending on how many skiers the model was trained. There are some drawbacks with the dGNSS approach. The need for stationary base stations that need time to calibrate, and need to be placed so all of them can communicate with each other [13]. Obstacles such as trees and buildings, and differences in elevation make this placement non-trivial.

Rindal et al. [17] use two Integrated Measurement Unit (IMU)s on the skier's arm and chest and a Multi-layer Perceptron (MLP) to do classification on the sensor data. The dataset they use consists of 10 skiers performing 6 techniques. A cycle detection method is used to split the sensor data based on the cycles. The split data is then interpolated or decimated, to ensure equal length on all splits. They achieved good results with ≈93 % accuracy on a relatively large dataset (over 8000 cycles/feature vectors). The Arctic HARE system utilizes the same technique for cycle detection, but uses different machine learning methods, and explores different sensor distributions on the body of the skier to maximize accuracy while minimizing the number of sensors.

Rassem et al. [16] use deep-learning algorithms on 3D accelerometer data from cross-country skiing. They tested Convolutional Neural Network (CNN)s, different versions of Long Short-Term Memory (LSTM)s, and a MLP for classifying data from 1SK and 2SK skating. They segmented the accelerometer data by using a window over 1 second with 50 % overlap. The Arctic HARE system also utilizes deep-learning models for classifying the different sub-techniques, but explores different types of data from more classes and employs different preprocessing methods before training.

There is an issue with the papers by Øyvind Gløersen and Gilgien [5] and by Rindal et al. [17] due to either not describing their data fully or not having a uniform dataset. Having a non-uniform dataset and evaluating a model based on its accuracy can be deceptive due to the accuracy paradox [11]. We circumvent this because of our balanced datasets and we use f1-score as a metric as well.

3 Arctic HARE

Arctic HARE is a prototype we have devised that aims to provide real-time performance analysis for cross-country skiers. It consists of multiple components for gathering and processing activity data. Multiple on-body Inertial Measurement Unit (IMU) sensors and a camera is used for measuring body movement. The sensor data is processed on a mobile computation device, and the video data is processed on a cloud node. Either data source can be used for classifying the sub-technique performed by the skier. We have applied machine learning methods to achieve this. These models can be trained in a cloud system. An overview of the architecture can be seen in Figure 1. More details can be found in [14].

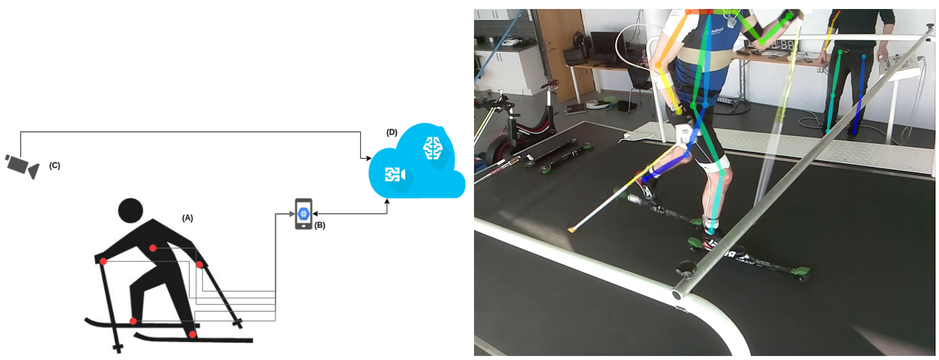

Fig. 1: Architectural overview of the *Arctic HARE* system. (A) illustrates the distribution of the sensors on the body of the user. They are connected to the mobile device (B) which, along with the static camera (C), communicates with the cloud (D). Also shows output of OpenPose result overlayed over video. Note that the head is outside of the view of the camera and the lack of points on the treadmill controller. This is discussed further in Section 6.2.

3.1 IMU Sensor Suit

A system was built to gather sensor data from the user's limbs. Five IMU sensors were distributed onto forearms, calves and chest of the user which were connected to a Raspberry Pi. The sensors themselves measure acceleration, magnetic field and orientation in three orthogonal directions, and are located on the limbs of the user. These sensors' values across the x, y and z directions comprise the total 46 features that the *Arctic HARE* system uses in its feature vectors, including a timestamp.

The sensor locations were chosen to be on the limbs in order to properly quantify the movement of the user. The sensor on the chest was added because it is commonly used [17, 3] and it captures the average overall movement of the user. It can also be used as a reference point for the other sensors to see how much they move with regards to the torso of the user.

3.2 Video System

A camera was mounted on the wall next to the user at a single stationary angle, which was used for the video data. The camera was connected to a black box embedded system for storage that would automatically save video data. The video data is at 42 FPS with a resolution of 1440×1080.

The video approach was chosen due to the recent increase in interest and success of computer vision both in industry and academia [9]. Exploring multiple methods of recording human movement can also have advantages for different kinds of movement and in multiple scenarios. Thus comparing or combining the video approach with the IMU-based approach seemed like an interesting area to explore.

4 Data Acquisition and Preprocessing

During data acquisition a professional ski athlete wears the sensor suit and is filmed with a stationary camera while skiing with roller skis on a large treadmill. The data was recorded from seven young national-class male elite skiers skiing at their respective marathon speeds. The skiers' chosen speeds and inclines were relatively similar for the respective sub-techniques. The treadmill speed and incline was adjusted during transitions between sub-techniques. The incline also did not change dynamically as they would in the field. This causes a problem regarding the data width, i.e. the data cannot represent the full range of realistic speeds and inclines. However, due to the nature of the system, which allows for edge computing, it can easily be tested in the field and can acquire data from more realistic conditions.

The subset of classical and skating sub-techniques that we consider are:

Diagonal Stride (DIA) is a classical technique where the skier moves their arms and legs in opposition, similar to how you walk. It is used mainly on uphills [12]. It is the only one of these techniques where the arms move asymmetrically.

Double Poling (DP) is another classical technique used while going slightly downhill or at high speeds. It is done by only pushing against the snow with the poles at the same time, with very little movement of the legs [12].

Double Poling with Kick (DPK) is a classical technique similar to DP, but it also involves a kick. The kick alternates between the left and right foot.

This technique is used for traveling across rolling terrain for long distances when conditions are too fast for DIA, but too slow for DP [12].

V1 skate (V1) is a skating technique, though it is quite different from other skating techniques. It is regarded as the best way to go uphill. It is done in sequence by first pushing the poles down, then planting one ski before planting the other ski [12].

One skate (1SK) is another skating technique. It is called "one skate" because there is one poling action for every leg push. This technique is often used on gentle terrain. It is also known as gear 2 [12].

Two skate (2SK) is a skating technique, named similarly to the one above because the is one poling action for every other leg push. This is a high speed technique. It is also known as gear 3 [12].

4.1 Preprocessing

Sensor Data The IMU sensor data is used to determine cycle length by detecting peaks in the data. The method used is similar to what is described in [17]. The z-axis data of the gyroscopic sensor on the right arm is filtered using a gaussian low-pass filter over 15 samples to remove high-frequency noise. The peaks of the signal are then detected using a first-order difference approximation of the derivative to find where the slope is zero. The indices of these peaks are then saved in a file to be used for splitting both the IMU and video data into sequences. The length of the longest cycle is stored in the configuration file described in Section 5 in order for new data sequences to be padded to the appropriate length.

After this, multiple data sets were generated, one for each of the sensor distributions. Each distribution was identified by a 5 bit code, where a 1 or a 0 indicated whether IMU sensor data from that specific sensor was present in the respective data set. These datasets were used for the comparison in Section 6.1.

Cycle length changes based on which sub-technique is used. Measuring the sub-techniques for five minutes each is not a guarantee for a completely balanced dataset, however the dataset was approximately uniform. The total amount of feature vectors in the training dataset was approximately $8,000$.

Video Data The raw video files are stored as multiple MPEG transport stream (`.ts`) files that are then concatenated together to one MPEG-4 (`.mp4`) file for each skier. It was then split up into the respective classes based on the IMU sensor data splits. The video sequences are then split into individual frames which are resized to 256 by 192 pixels with 3 channels (R,G,B) and concatenated as sequences of $256 \times 192 \times 3$-tensors where the sequences correspond to cycles similar to what is done with the IMU data. The frames are resized to fit into the Inception-v3 model. This dependency on the IMU sensor data to create the video cycles can be circumvented by for example using OpenPose for detecting cycles instead, or using a set interval duration as a sliding window. Before training of the neural networks the tensor sequences are padded with zero-tensors (which

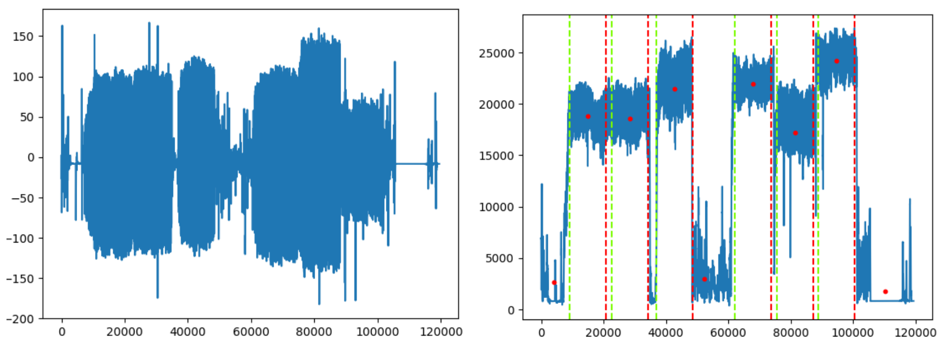

Fig. 2: The raw output of the z-axis of the gyroscope on the right arm, followed by the same signal after it has been through the annotation process. The red dots represent the cluster means and the green and red dashed lines represent the beginning and end of each class respectively.

corresponds to completely black images) so that all sequences are of the same length. The splitting procedure is very parallelizable, therefore each step was parallelized over the video sequences by using the `multiprocessing` module of `Python`.

The video data was considerably smaller with only 4 sub-techniques performed by 2 skiers. This reduced the amount of possible training data by a considerable amount to approximately $2,000$ feature vectors. It was therefore important to explore methods for reducing the dimensionality of the data, as was discussed above.

4.2 Data Annotation

Annotation of the IMU and video data is a crucial part in creating the training dataset. It is possible to do this by visual inspection, as can be seen in Figure 2, where there are six segments with different mean amplitudes. However, this process can be slow and the classes can be difficult to discern from each other, so alternative methods were explored. The goal was to split and annotate the data automatically.

The method we ended up using was the unsupervised clustering method K-means on the sensor data after convolving the absolute value of it with a 1-vector with 100 entries. This effectively scales the amplitudes of the signal outputs during technique performances, but not between them. We tried to locate 9 clusters; the six sub-techniques and three pauses located at the beginning, between the classical and skating parts, and at the end. The interval of a certain sub-technique is determined to be at the mean of each cluster: $\text{Interval}_k = \mu_k \pm \frac{L}{2}$, where L is the total duration of a sub-technique performance in number of measurements ($\approx 12,000$). If the pauses are too long it can interfere with the clustering, but these can be trimmed or avoided during data acquisition by starting and stopping the program at times closer to the technique exercises. The final result can be seen in the second image in Figure 2. The reason this works is because we know the order and duration of the techniques that are performed. We can therefore visually inspect whether the clusters make sense.

5 Machine learning methods

All of the models for the IMU sensor data and video data were created using *Keras*[1] [24], a neural network framework running on top of *TensorFlow*[2] [1]. *Keras* provides an API for constructing neural networks with a layer abstraction. Dropout was used to prevent overfitting. The sequences were padded to the same length so that they could be batched uniformly.

5.1 On IMU Sensor Data

The choice of using an LSTM-based network for classification of the IMU sensor data is based on good results on time series data [6]. The IMU data was concatenated based on a cycle detection method described in Section 4.1 and padded before being classified by an LSTM network. Datasets for each sensor configuration was created and fed through the network to determine the best configuration (see Section 6.1). The model we used has two LSTM layers followed by a fully-connected layer and then a Dropout layer. The number of neurons of the LSTM layers was set to 50 while testing the different sensor configurations as a initial experimental setup.

After a suitable sensor configuration was found, grid search was used to determine optimal hyperparameters while trying to make the model as small as possible, which was a requirement in order to make training and classification more efficient. The number of units in the LSTM layers were varied between 10 and 128 units and the dropout layer's rate varied between 0.1 and 0.7.

5.2 On Video Data

Convolutional Neural Networks (CNN) have achieved state-of-the-art results on image data [20] and are widely used in industry today [10]. Combining this with RNNs was therefore expected to yield good results.

Two different CNN-based methods were used for feature extraction; one based on extracting general features of the frames and one that uses pose estimation from the frames to determine the locations of the body parts of the skier.

The first method is done by sending the processed video data sequences through Inception-v3 [20], which is a CNN trained on the ImageNet dataset [18]. We use all the layers except the final layers which are fully-connected and used for classification. The frames reduced from $256 \times 192 \times 3$ tensors to 2048-dimensional vectors. The vectors are combined into sequences similar to what is done with

[1] https://keras.io/

[2] https://www.tensorflow.org/

IMU data. Then these output sequences are run through an LSTM network which classifies the sequences into the different sub-technique classes.

Openpose [8], which is another CNN-based application, was the second method used for feature extraction. Openpose extracts the positions of the limbs of the skier in the video frames. These positions, which can be seen in the second image in Figure 1, are used as features in a 36-dimensional vector which are combined into sequences, similar to what is done above.

The LSTM architectures were also tuned by a grid-search, but were not constrained by trying to minimize the network size as what was done in Section 5.1. Due to the fact that the features of the training data are different and the feature vectors have a different dimensionality, one needs to determine which architecture works for the given data.

6 Evaluation

To collect data and perform experiments we used a Raspberry Pi 3 model B connected to five ADXL345 IMUs [22] as our mobile computation device. The Raspberry Pi ran Raspbian 4.4 and the server ran Ubuntu 16.04.4 LTS. A Raspberry Pi was chosen due to having similar hardware limitations, size and architecture as modern smartphones, but with a simple interface for connecting the wired IMU sensors. Two different cloud instances were used, one rack server with 16 GB of memory and two quad-core Intel Xeon X5355 CPUs running at 2.66 GHz and a 1 Gbit / s ethernet network interface for training the sensor-based model. A tower computer with quad-core Xeon E5-1620 CPU running at 3.70 GHz and a Titan Xp GPU with 12 GB RAM and 3584 CUDA cores running at a base rate of 1417 MHz was used for training and classification with the video-based model.

We apply repeated stratified k-fold cross-validation in order to evaluate the predictive ability of the models, with $k = 7$, giving a training/test split of approximately 85%/15%. The repetition is done the get more stable results and to avoid bias of the estimator [2].

6.1 Comparing Sensor Distributions

We wanted to investigate an optimal distribution of sensors on the body of a ski athlete that minimizes the number of sensors. This is because we avoid the problem with the *curse of dimensionality* [21], and we wanted to make the sensor suit easier to apply and less interfering for the skier.

First, we performed an exhaustive search over the entire grouped feature space. The space is grouped because subsets of the features belong to different IMU sensors. As was defined in Section 4.1, the different sensor distributions are represented by a 5-bit code.

The $(50, 50)$ layer configuration used for the results above contains a total of $34,306$ parameters that need to be trained. The top 5 sensor configurations that

Sensor Conf.	Mean Acc.	Std. Dev.
00110	96.94 %	2.71 %
11000	93.71 %	5.18 %
01100	93.28 %	4.94 %
10010	91.94 %	6.20 %
01001	91.20 %	6.87 %

Table 1: Table of the top 5 sensor configurations with the best accuracy.

Hyperparameters	Accuracy	F1-Score
20,10,0.1	79.25 %, 13.95 %	70.92 %, 27.34 %
32,64,0.2	82.45 %, 10.99 %	78.52 %, 29.17 %
50,50,0.1	96.11 %, 3.98 %	96.33 %, 4.29 %
64,64,0.2	95.87 %, 5.48 %	92.49 %, 3.31 %
128,64,0.4	96.77 %, 2.26 %	94.74 %, 4.02 %

Table 2: Table of the top hyperparameter choices with means and standard deviations of accuracy and f1-score.

contain less than 4 IMU sensors, with corresponding accuracies can be seen in Table 1.

The best sensor configuration seems to be 00110, which corresponds to the IMU sensors on the left leg and on the right arm. As a result, this configuration will be explored further. The grid search space is between 10 and 128 for the two LSTM layers, and between 0.1 and 0.7 for the dropout rate.

It is important to note how the hyperparameters scale the number of weights that need to be trained and used for calculating predictions; a $(20, 10)$ network consists of $4,426$ trainable parameters, a $(50, 50)$ network consists of $34,306$ trainable parameters, and a $128, 128$ network consists of $207,622$ trainable parameters. We want to reduce the size of the model by minimizing the hyperparameters as well. Based on the results in Table 2, we want to keep the $50, 50$ configuration due to the high accuracy and f1-score and the fact that it is a relatively small network. Results from 5 times repeated 7-fold cross-validation.

Finally, we will test the classification rate using the final sensor distribution. This efficacy test will be done to determine if the IMU-based approach can handle real-time feedback. The live classification application fills a buffer corresponding to approximately 5.3 seconds before detecting the peaks and concatenating the feature vectors. The time from getting the first feature vector in the buffer to classification was determined to be $5.32\,\mathrm{s} \pm 0.11\,\mathrm{s}$. Thus the time it took to process the feature vectors is inconsequential to the rate of classification and the true bottleneck lies in the fact that classification cannot occur before the buffer is full.

The right leg appeared in multiple configurations that got the best accuracies. The reason for this might be that the skiers most likely are all right-handed, and therefore probably favor the right leg as well. This can be significant, particularly for less symmetric movements such as V1. The best configurations include IMU sensors on a leg and an arm. Combining the IMU sensors on a leg and an arm makes sense considering most movements are highly symmetrical and those limbs encompass the entire movement well.

6.2 Comparison of the Video-Based Methods

Both of the methods tested on the video data are examples of transfer learning. We wanted to determine which method would produce the best features to use in order to classify the video data correctly. It is important to extract the relevant information from the video so that the models are trained correctly. This is related to the curse of dimensionality. We do not want the models to learn irrelevant particularities about the video data. Therefore we conjectured that OpenPose will give a better result, given that Inception-v3 gives a more general representation of the entire frame.

First we determined the best LSTM layer configuration for the respective feature extraction methods. This was done via a grid search as explained in Section 5.2. The number of the units in the LSTM layers ranged between 32 and 256, and the Dropout rate between 0.1 and 0.7.

Based on the results above we chose $(256, 32)$ and $(64, 128)$ as the number of units respectively for the LSTM layers after Inception-v3 and OpenPose feature extraction. The accuracy and F1-scores using both feature extraction methods is given in Table 3. We also evaluate efficacy of the two methods.

Feature extraction	**Accuracy**	**F1-score**	**Processing Time per File**
Inception-v3	94.33 %, 4.90 %	93.32 %, 7.18 %	20.55 s ± 0.08 s
OpenPose	86.69 %, 13.13 %	87.94 %, 17.23 %	10.44 s ± 0.12 s

Table 3: Accuracy and F1-scores with corresponding standard deviations using the different feature extraction methods. Average time for processing a 2 second video file is also shown.

7 Conclusion

We have designed, implemented and experimentally evaluated *Arctic HARE*, a machine learning-based system for performance analysis of cross-country skiers. In this work we wanted to explore two different approaches to automatic performance analysis of cross-country skiers. We wanted to minimize the number of IMU sensors while achieving acceptable accuracy and see whether the video-based approach is viable by exploring feature extraction methods.

The system was built with this mind by creating the sensor suit with five IMU sensors distributed across the body, and using multiple feature extraction methods on the video data. We also implemented several applications for data acquisition and live classification in order to test the system. The novelty of our work is two-fold. Firstly, the IMU-based part of the system is located on the body of the skier and gives feedback in real-time which allows for in-field usage. Secondly, our exploration of video-based activity recognition with possible applications.

We determined from the results of the experimental evaluation of the system that, for the task of performance analysis of cross-country skiers, it is possible to achieve high classification accuracy using only two IMU sensors. Also, based on the processing time needed for the video data and the fact that the IMU sensors can easily be used in multiple environments, the IMU-based approach is more fitting for the task compared to video. However, the video-based approach can be used for specific areas, such as post-race analysis of the finish line.

Despite cross-country skiing being our domain of research in this text, numerous other sports share similar periodic patterns. Therefore, we can apply *Arctic HARE* to these as well, and we are currently exploring other fields such as football and running.

Acknowledgements

This work is partially funded by the Research Council of Norway project number 274451 and Lab Nord-Norge ("Samfunnsløftet").

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)

[2] Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of k-fold cross-validation. Journal of machine learning research 5(Sep), 1089–1105 (2004)

[3] Casale, P., Pujol, O., Radeva, P.: Activity Recognition from Single Chest-Mounted Accelerometer. https://www.researchgate.net/publication/260425987_Activity_Recognition_from_Single_Chest-Mounted_Accelerometer (03 2014)

[4] FitnessKeeper, I.: Runkeeper. https://runkeeper.com/ (2018), online; Accessed 21.05.2018

[5] Øyvind Gløersen, Gilgien, M.: Classification of Ski Skating Techniques using the Head's Trajectory for use in GNSS Field Applications. https://www.researchgate.net/publication/311735263_Classification_of_Ski_Skating_Techniques_using_the_Head's_Trajectory_for_use_in_GNSS_Field_Applications (2016)

[6] Graves, A.: Supervised sequence labelling. In: Supervised sequence labelling with recurrent neural networks, pp. 5–13. Springer (2012)

[7] Halvorsen, P., Sægrov, S., Mortensen, A., Kristensen, D.K., Eichhorn, A., Stenhaug, M., Dahl, S., Stensland, H.K., Gaddam, V.R., Griwodz, C., et al.: Bagadus: an integrated system for arena sports analytics: a soccer case study. In: Proceedings of the 4th ACM Multimedia Systems Conference. pp. 48–59. ACM (2013)

[8] Hidalgo, G.: OpenPose: Real-time multi-person keypoint detection library for body, face, and hands estimation. https://github.com/CMU-Perceptual-Computing-Lab/openpose (2017)

[9] Kishor, N.: Top 8 Technology Trends for 2018 You Must Know About. http://houseofbots.com/news-detail/2653-4-top-8-technology-trends-for-2018-you-must-know-about (2018), online; Accessed 15.05.2018

[10] Lambert, F.: Tesla's new head of AI and Autopilot Vision comments on his new role. https://electrek.co/2017/06/21/tesla-ai-autopilot-vision/ (2017), online; Accessed 25.05.2018

[11] Lao, R.: Machine Learning — Accuracy Paradox. https://www.linkedin.com/pulse/machine-learning-accuracy-paradox-randy-lao (2017), online; Accessed 16.05.2018

[12] McKenney, K.: Cross-country ski techniques with video examples. http://crosscountryskitechnique.com/ (2014)

[13] Navipedia: Differential gnss (2014), https://gssc.esa.int/navipedia/index.php?title=Differential_GNSS&oldid=13309, [Online; accessed 3-May-2018]

[14] Nordmo, T.A.S.: Arctic HARE: A Machine Learning-based System for Performance Analysis of Cross-country Skiers (2018), Master's Thesis

[15] O'Donoghue, P.: Research Methods for Sports Performance Analysis (2010)

[16] Rassem, A., El-Beltagy, M., Saleh, M.: Cross-country skiing gears classification using deep learning. arXiv preprint arXiv:1706.08924 (2017)

[17] Rindal, O.M.H., Seeberg, T.M., Tjønnås, J., Haugnes, P., Sandbakk, Ø.: Automatic classification of sub-techniques in classical cross-country skiing using a machine learning algorithm on micro-sensor data. Sensors 18(1), 75 (2017)

[18] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115(3), 211–252 (2015)

[19] Software, Q.: Quintic software: Biomechanical 2D Video Analysis. https://www.quinticsports.com/software/ (2018), online; Accessed 21.05.2018

[20] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)

[21] Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press, 4 edn. (2009)

[22] www.analog.com: Adxl345. http://www.analog.com/en/products/mems/accelerometers/adxl345.html (2013)

[23] www.hudl.com: Hudle: One platform to help the whole team improve. https://www.hudl.com/products/hudl (2018), online; Accessed 21.05.2018

[24] www.keras.io: Keras: The python deep learning library. https://keras.io/ (2018), accessed 10.03.2018