RESEARCH ARTICLE

A deep learning approach for lower back-pain risk prediction during manual lifting

Kristian Snyder[ID][1]*, **Brennan Thomas**[1], **Ming-Lun Lu**[2˚], **Rashmi Jha**[1˚], **Menekse S. Barim**[2‡], **Marie Hayden**[2‡], **Dwight Werren**[2‡]

1 Department of Electrical Engineering and Computing Systems, University of Cincinnati, Cincinnati, Ohio, United States of America, **2** National Institute for Occupational Safety and Health, Cincinnati, Ohio, United States of America

˚ These authors contributed equally to this work.
‡ These authors also contributed equally to this work.
* snyderks@mail.uc.edu

Abstract

Occupationally-induced back pain is a leading cause of reduced productivity in industry. Detecting when a worker is lifting incorrectly and at increased risk of back injury presents significant possible benefits. These include increased quality of life for the worker due to lower rates of back injury and fewer workers' compensation claims and missed time for the employer. However, recognizing lifting risk provides a challenge due to typically small data-sets and subtle underlying features in accelerometer and gyroscope data. A novel method to classify a lifting dataset using a 2D convolutional neural network (CNN) and no manual feature extraction is proposed in this paper; the dataset consisted of 10 subjects lifting at various relative distances from the body with 720 total trials. The proposed deep CNN displayed greater accuracy (90.6%) compared to an alternative CNN and multilayer perceptron (MLP). A deep CNN could be adapted to classify many other activities that traditionally pose greater challenges in industrial environments due to their size and complexity.

Introduction

Back pain, especially when occupationally-induced, is an extremely common ailment. 23.2% of the world's population is estimated to be affected in any given month [1]. It is considered the leading cause of job-related disability and missed work days, resulting in massive losses of productivity [2, 3]. Back pain is, in the United States, the largest contributor to total workers' compensation costs as of 2015, representing over 20% of costs and 13.7 billion USD annually [4].

The revised National Institute for Occupational Safety and Health (NIOSH) lifting equation [5] (RNLE) is currently considered the leading method of measuring back pain risk involved in both single and repeated lifting of objects in the workplace [6, 7]. Given the mass, relative source distance, relative destination distance (to the person) of an object, frequency of the lifting tasks, and the work-rest pattern, the RNLE determines the relative level of risk to the lifter. Taking measurements for using the RNLE in the field presents a challenge because the analyst

needs to interrupt work activity for measuring several lifting variables to recognize the characteristics of lifting tasks. In a world of activity recognition and wearable sensors becoming prominent [8–11], there is a distinct lack of research into automatically detecting lift risk, especially as it relates to back pain.

Real-time recognition of unsafe lifting would provide immediate feedback to the users, something currently impractical with traditional methods such as a specialist constantly monitoring the workers. Therefore, the ability to, in real-time, detect unsafe lifting behavior in an industrial setting would provide significant benefits. Closing the feedback loop allows for the wearer to be quickly alerted to any risk and help prevent further stress. A response time in seconds presents a significant improvement over typical pain feedback, which does not always present itself quickly and may be alongside debilitating injury [12]. Additionally, an automated approach is far more scalable and can be inexpensively rolled out to an entire set of workers for minimal cost compared to treatment and loss of productivity due to back injury.

Classifying a person lifting objects of various distance from themselves presents an inherent challenge because even to an observer, the different movements are quite similar. Additionally, sourcing data for specialized activities has its own challenges, given a dearth of existing datasets and the increased expense of independently collecting data. Consequently, datasets are typically small; the NIOSH lifting dataset consists of 720 total trials [13]. The challenge, then, is to develop a versatile model that can distinguish between very similar activities and operates on datasets possibly magnitudes smaller than for similar problems [14].

Significant advancements in deep learning approaches have been achieved in recent years, broadly categorized into video-based and sensor-based classification [15]. Video-based approaches have focused mostly on surveillance and gait recognition, although there is recent research that has successfully labeled live video data with a high degree of accuracy without assistance from manual body part labeling or tracking devices [15, 16]. While impressive, use of video classification in many workplaces requires a wide array of surveillance cameras placed to view all workers, which can be expensive and require complicated installation. In other areas, such as most construction and industrial sites, video is near-impossible due to the frequently temporary and constantly fluctuating workplaces. Sensors placed on the body do not require the person to be in a specific place and are relatively inexpensive and less privacy-intrusive than constant video capture [15], making them more practical for this use case.

Leading sensor-based approaches typically classify more distinct activities, such as standing, walking, and running with relative success [14, 17, 18]. However, classification of visually similar activities (such as walking up and down stairs) displays notably lower performance [17]. Additionally, other attempts to classify accelerometer data focused on $n > 500$ samples for each class and commonly in excess of thousands of samples [10, 14, 17, 19, 20]. The existence of other datasets with multiple accelerometers suggests a place for a model that can work on very detailed activities generating much more data with dedicated sensors than a smartwatch or smartphone [10].

In this paper, a model utilized to classify the above lifting activities is presented as a generalizable solution to the problem of classifying small datasets of very similar activity classes. The model uses 2-dimensional convolutional layers along with average pooling to classify the activities. Accelerometer and gyroscope data is first preprocessed with a Butterworth filter and manipulated into a 3D matrix resembling an image before being trained on. The structure of the model is examined by comparing it to variations in pooling, regularization, and complexity to display the theoretical underpinnings, increasing its adaptability to other problems. The network's various hyperparameters are also examined to justify their specific values. Finally, accuracy and other statistics of the proposed model are compared to traditional activity recognition models and their published results.

Expanding the list of activities to a wider subset of possible movement unlocks a multitude of possible benefits. In collaboration with the National Institute for Occupational Safety and Health (NIOSH), a deep convolutional neural network (CNN) model to classify relative risk level of lifting objects is developed. Such a model would assist in preventing serious, chronic back injury in workers and significantly improve their quality of life.

Materials and methods

Data collection

For purposes of analysis, data utilized for model development was sourced from a previous study by researchers at NIOSH to examine body motion for two-handed lifting tasks similar to those in the workplace [13]. Lifts were performed using the American Conference of Governmental Industrial Hygienists (ACGIH) Threshold Limit Values (TLV) for lifting, which defines 12 zones relative to the body in the sagittal plane shown in Fig 1. Save for Zones 1-3, 4, 7, and 10, all lifts began in the midpoint of these zones. In the aforementioned exceptions, initial points in the zone were altered to provide realistic motion for the subjects' ranges of motion. The object lifted consisted of a 36cm × 12 cm wire grid weighing 0.45 kg with two handles, simulating a crate or box. To prevent injury, the weight was kept small. The order of
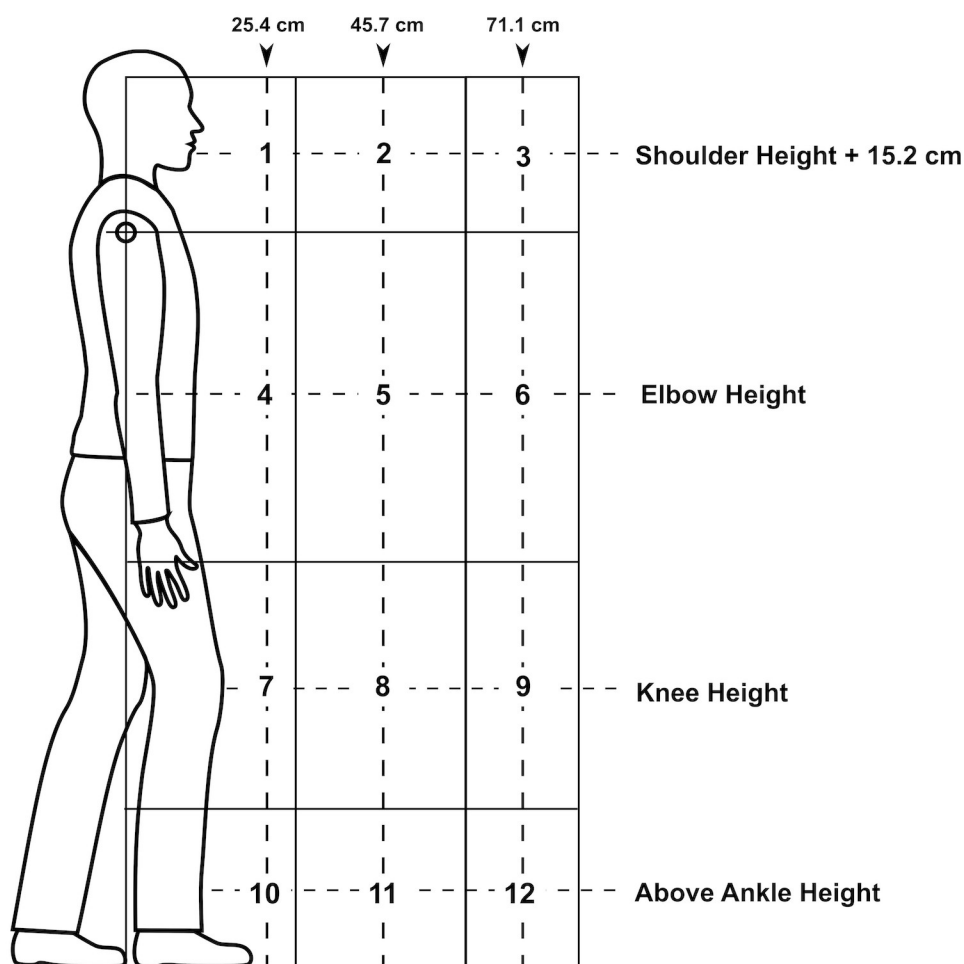

Fig 1. ACGIH lifting zone system depicting the relative areas collected for analysis. (Source: NIOSH).

lifting from each zone was randomized to prevent bias from ordering the lifts. A total of 720 trials with 6 trials for each subject in all zones were available as input data. Physical information about the trial participants, while collected in the previous study, was not utilized. This includes age, gender, height, and weight. While all of these factors can contribute to lower back pain, the focus of the model is not to predict risks based on inherent behavior but rather a pattern of behavior that spans these characteristics, linked to the physical risk zones that are the classes used by the model. These personal risk factors cloud the perception of the model as confounding variables.

Five male and five female subjects (mean and SD: 170 ± 7.4 cm for height and 85.7 ± 20.2 kg for weight) participated in the lifting process. All participants had six inertial measurement unit (IMU) sensors (Kinetic Inc.) attached to their bodies on the upper back (T12), each wrist, the dominant upper arm, waist, and the dominant thigh during all lifting. Each sensor consisted of a tri-axial gyroscope and accelerometer sampling at a rate of 25 Hz. All sensor data was calibrated and synchronized prior to data collection [13]. An example of this data is shown in Fig 2.
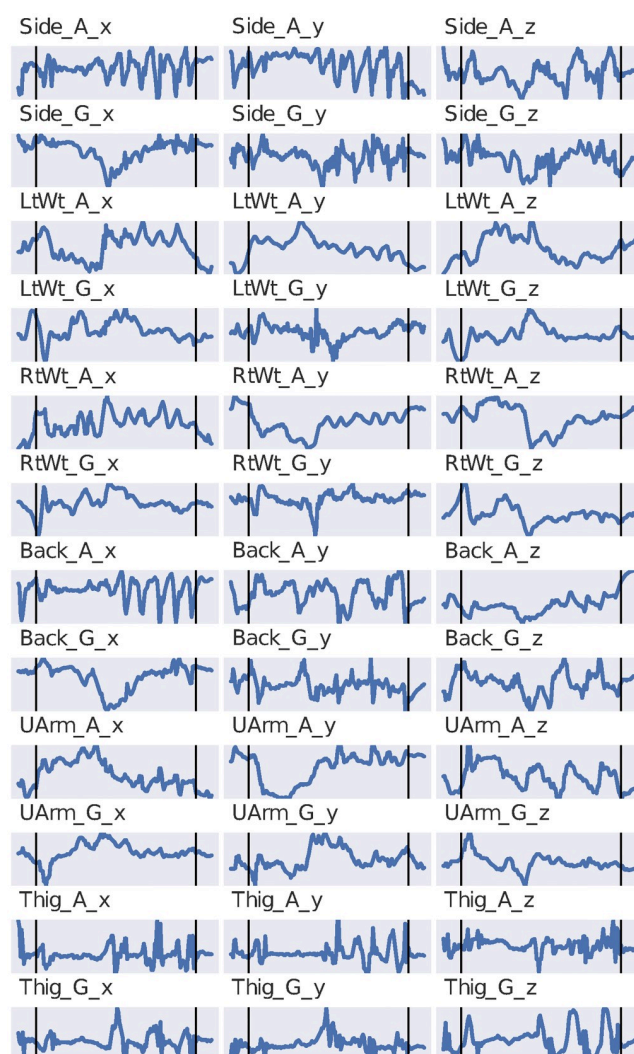
Fig 2. Plots of the accelerometer and gyroscope sensors for subject 1's first lift in zone 1 (high risk) from [13]. Not all data collected is shown; the two vertical black lines show the beginning and end of the actual lift.

Feature extraction

In training the model, 540 of the 720 total trials were utilized for training while the remaining 25% (180 trials) formed the test set. Training and testing subsets were sampled randomly without replacement. Features were the preprocessed lift, with zero-padding applied to lifts that did not reach the full time. All training utilized all of the 6 sensor areas, each containing an accelerometer and gyroscope, resulting in 18 total measurements for each point in time. The twelve ACGIH lifting zones were mapped to three risk levels: low, medium, and high-risk. The mapping is based on a slight modification to the Los Alamos National Laboratory recommendations to simplify the ACGIH zones; the zones mapped are listed in Table 1.

Validation, driving final statistics and hyperparameter tuning, was performed with a modified form of cross-validation. In creating the four folds (each with 75% training, 25% testing), each *ACGIH lifting zone* was individually sampled. Due to each zone only having 60 total trials, the probability that an entire zone would be absent from either the training or testing portion was unacceptably high. Sampling from each zone ensured that the model would both have examples of each activity and be tested on each activity.

Finally, in each case the data was scaled with normalization to the bounds [-1, 1] based on the training data, with the same scaling applied to the testing data. This is in direct interest of increasing performance of the neural network, which trains best on data normalized to these bounds. This resulted in 720 total 27,000-dimension class-labelled vectors for training and testing.

A maximum time window of 30 seconds was selected to train the model. Lifts that did not reach this length of time were zero-padded to reach the full time. Most trials did not reach this time period; the majority ended between 10 and 15 seconds. The long time slice length was selected to investigate dependence on alignment of the starting and ending times, ideally mitigating or eliminating any significant dependence. To reduce overall noise and drift, a Butterworth filter with order 2, lower bound of 2 Hz and upper bound of 12 Hz was applied to each dimension (X, Y, Z) of the gyroscope and accelerometer in each sensor, resulting in 36 total measurements for each point in time. A Butterworth filter is a signal processing filter that has maximally flat frequency response in the passband, preserving the original signal better than other filters [21]. Each dimension of each of the 12 sensors (an acceleromter and gyroscope each on the side, left wrist, right wrist, back, upper arm, and thigh) has 750 data points in each trial, resulting in 27,000 total features as shown in Eq 1.

$$
\begin{aligned}
& 6 \text{ IMU devices} \\
\times\ & 2 \text{ sensors per IMU (gyro and accelerometer)} \\
\times\ & 3 \text{ dimensions each} \\
\times\ & 30 \text{ sec max per trial} \\
\times\ & \frac{25 \text{ frames}}{\text{sec}} \\
=\ & 27{,}000 \text{ features}
\end{aligned}
\tag{1}
$$

Parameters and the filter itself were chosen based on previous research [8, 22, 23].

Table 1. Mapping of ACGIH lifting zones to relative risk levels.

Risk Level	Zones
Low	4, 5
Medium	6, 7, 8, 9
High	1, 2, 3, 10, 11, 12

To prepare the data for ingestion to the model, each 27,000-feature vector was reshaped to form a $95 \times 95 \times 3$ matrix. Specific reshaping is performed by stacking each sensor to form a $36 \times 750 \times 3$ matrix, placing each point of time into successive columns. The matrix is then line-wrapped to form the final image. See Fig 3 for a visualization of the process. Fig 4 displays an example of the final image after standardization.

This method produces a data format friendly to CNN models while preserving time locality of the data as much as possible. Convolutions, therefore, will be made more often between features occurring at similar times to help correct for an imprecise window mislabeling the start and end of the lift. Finally, the data is standardized for each sensor by scaling the data to a mean of zero and standard deviation of one.

Overall statistics

During testing and training of the proposed model, several class-specific statistics were collected to help measure its performance. In addition to the statistics below, two overall statistics were also collected: R_K (primarily utilized in hyperparameter tuning, see 3.1. Hyperparameter tuning) and overall accuracy.

Precision (shown in Eq 2) is defined as the proportion of instances belonging to the class (true positive or TP) over all instances (both TP and false positive or FP) classified as that class.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall (shown in Eq 3) is defined as the proportion of a elements in a particular class classified as that class over all elements belonging to that particular class (including TP and false negative or FN).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F-measure (shown in Eq 4) is the harmonic mean of precision and recall, used as an alternative to raw accuracy.

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{4}$$

Model design

The proposed model is based on the Visual Geometry Group Network (VGGNet, developed at the University of Oxford), a high-performing CNN model that is notable for its high depth and use of additional layers and small kernel size instead of fewer layers with a larger kernel size [24]. This helps to reduce the number of parameters of the network as well, especially important for small datasets which most models with many parameters struggle to converge on [25].

Specifically, the model is based on variation B of VGGNet (VGGNet B), with max pooling layers separated by groups of two convolutional layers with increasing filter count. Most notably, the filter count in each group of convolutional layers is smaller than for VGGNet B, ranging from 32 to 128 filters instead of 64 to 512. Additionally, the 2×2 max pooling layers are replaced with 2×2 average pooling layers. Max pooling is employed in most CNN models to increase contrast and preserve the most important information of an image while decreasing dimensionality of the input; however, contrast for the images is already quite high, with
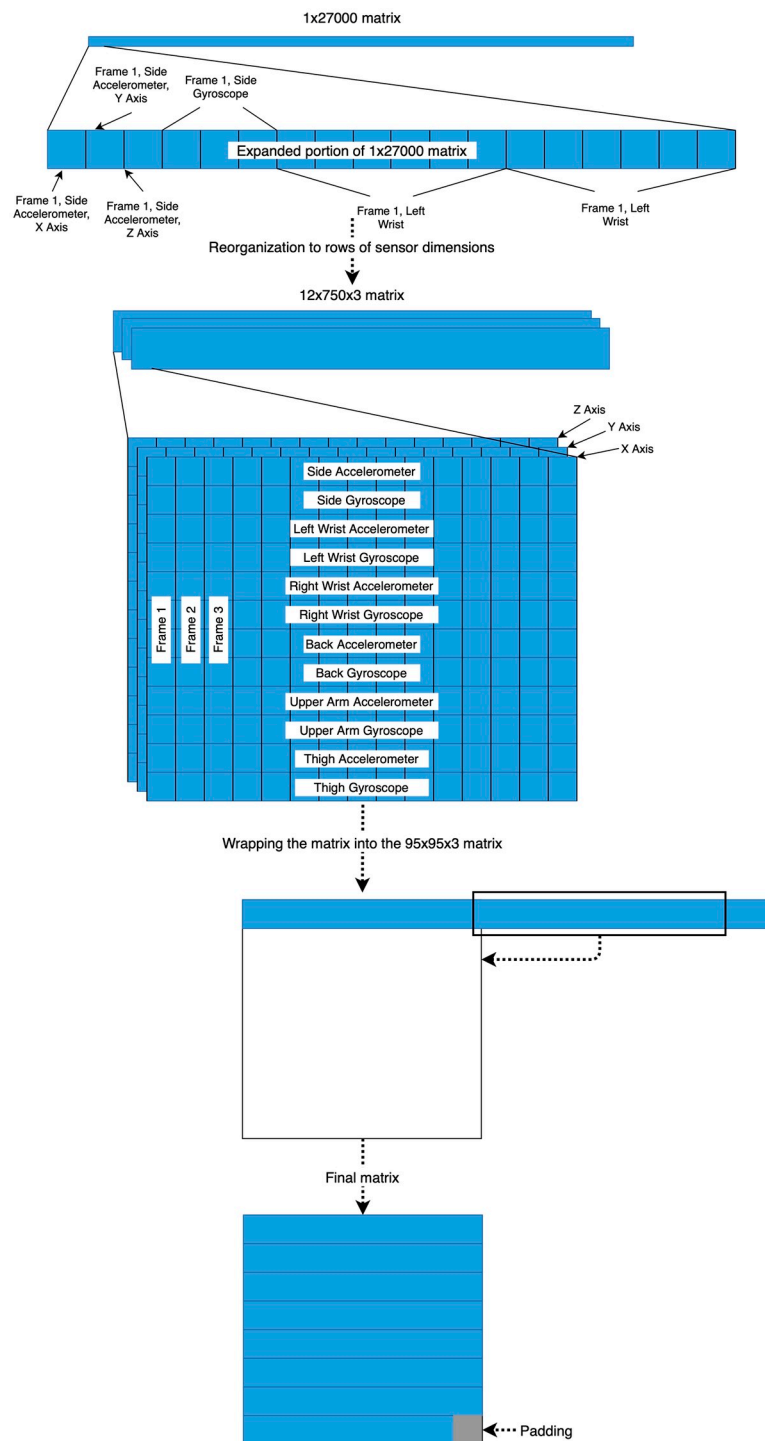
Fig 3. Process of ingesting data into the model for training. All figures not to scale. The resultant matrix could theoretically be any size; a square was selected for highest compatibility with existing CNN research.
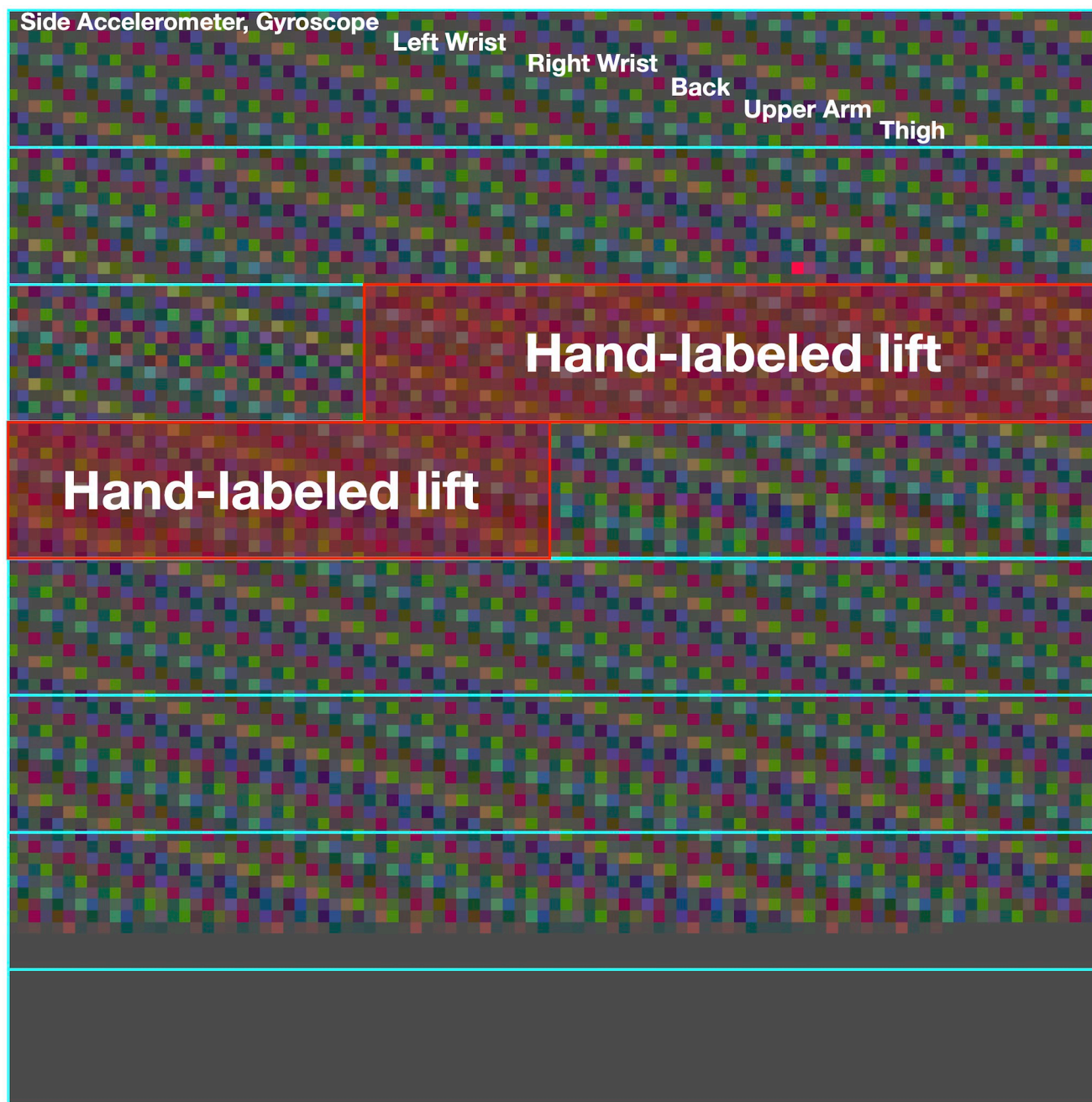
Fig 4. Example of an input image to the network. The image shown has had a Butterworth bandpass filter of order 2 and bounds 2 and 12 Hz applied to it in addition to a standardizing scaler. The grey block at the bottom represents padding to the model that makes all inputs the same size.

important features throughout the sample [26]. Average pooling retains more information from layer to layer because it incorporates all source pixels in the output compared to max pooling rejecting all but one of the pixels. When adaping VGGNet B to train on accelerometer data, this most significantly improved performance on the NIOSH lifting dataset. Table 2 contains a detailed description of the layers.

Table 2. Detailed specification of the layers involved in the proposed model. All 2D convolution layers contain a ReLU activation layer.

Model configuration	
Layer type	**Parameters**
Input (95x95x3 matrix)	
2D convolution	32 filters, 3x3 kernel
Average pooling	2x2 cell
Dropout	25%
2D convolution	64 filters, 3x3 kernel
2D convolution	64 filters, 3x3 kernel
Average pooling	2x2 cell
Dropout	25%
2D convolution	128 filters, 3x3 kernel
2D convolution	128 filters, 3x3 kernel
Average pooling	2x2 cell
Dropout	25%
Flatten	
Fully connected	1024 units
Batch normalization	
Dropout	25%
Softmax output	

https://doi.org/10.1371/journal.pone.0247162.t002

Pretraining—training a neural network on similar data to improve weight initialization and then fine-tuning on the final dataset—of the model was considered as a possibility prior to training on the previously mentioned dataset. However, this approach results in issues that impact both the model's feasibility and applicability to other challenges. Firstly, pretraining requires a similar dataset to that which is being studied. For situations such as image recognition, these are plentiful and robustly tested. In the case of activity recognition, especially multi-sensor activity recognition, previous research is limited in scope and so presents a challenge of needing to pretrain on data that does not exist. Additionally, this limits the model's usefulness in other research; requiring robust knowledge of the field and pretraining limits the model to experts who can draw on their own experience to develop a more tailored model.

Examining the dataset's size, 720 total samples, 540 of which used in training, is a relatively small quantity compared to other applications of CNNs. However, the field of activity recognition presents a challenge for data collection; save [14], the other referenced manuscripts contain datasets within an order of magnitude of that used in this model. While a challenge, working with small amounts of training data has significant precedent and so was not rejected as an approach.

In addition, [27] suggests that the generalization—applicability of a model beyond its training and testing data—of neural networks trained on small datasets is approaching performance provided by large, well-studied datasets like MNIST. Most importantly, [27] observed this for small datasets with a significant amount of noise, which the collected dataset contains even after some cleanup.

In addition to the proposed model, a separate model (CNN+LSTM) was developed as an alternative approach, utilizing a network of 1-dimensional convolutional layers and long short-term memory (LSTM) layers, based on DeepConvLSTM by Ordóñez et al [28]. This approach produces a far less complex network and treats the dataset as a time series instead of a 3D matrix thanks to the LSTM layers. LSTM layers utilize the current state of the network

with their memory units, building direct relationships between the currently analyzed data and previously analyzed data. This makes it especially effective on temporally organized data, such as accelerometer and gyroscope signals [29].

The CNN+LSTM model also uses a slightly data format. Instead of the 95x95x3 matrix, the 12x750x3 matrix is utilized. Features are extracted by sliding a 12x1x3 window (essentially a column of the matrix) over the trial, reducing the dimensionality of the data from 3D to 2D and making it compatible with the LSTM layers. See [28] for additional details on the structure of LSTM and 1D convolutional layers.

Three other models were developed strictly for comparison: a simpler CNN model with lower depth, a max pooling VGGNet B variation, and a multi-layer perceptron (MLP) model. The simpler CNN model does not utilize any form of pooling or a dense layer, consisting only of convolutional layers and a softmax output layer. The max pooling VGGNet B model is identical to the proposed model in all ways save the usage of max pooling layers instead of average pooling.

Models were all trained in the same manner, utilizing ADAM [30] as a gradient descent estimator, categorical cross-entropy as a loss function and, in place of a specific number of epochs, utilizing early stopping to halt training. The early stopping module monitored loss with a min delta of 0 and patience of 10 epochs. After loss failed to decrease, the best-performing (according to loss, not accuracy) weights of the last 10 epochs were selected for testing.

To develop the models, Keras v2.2.4 was used as an interface to TensorFlow v1.15.0. Data manipulation and standardization was performed with Scikit-learn v0.21.3.

Hyperparameter tuning

A difficult portion when developing machine learning models is tuning the various hyperparameters, which are parameters for the model that are statically set before training begins. These parameters can affect training and testing results as significantly as alterations to the model structure and so deserve their own discussion.

In developing the model, three hyperparameters were focused on due to their great effect on accuracy: L2 regularization importance (λ), dropout percentage, and learning rate (α). Searches were not performed for global minima due to the extreme effort involved and were tested with both single-value variation and multiple-value variation to evaluate any effects dependent on multiple hyperparameters.

All hyperparameters were tuned *solely on cross-validated results*. Training/testing loss and other statistics were not used in tuning hyperparameters.

λ configures the importance of the L2 regularizer, which is applied solely to the final softmax dense layer in both the activity and output portions. Regularization in general is the practice of adding a loss function to the *complexity* of the layer and incentivizing training of a sparser and less complex model. This helps to reduce the prevalence of overfitting to the training data [31]. Importance is the weight given to this regularizer over the basic loss function used in training; a higher value signifies higher importance. Most values for λ are between 0.1 and 10^{-10}.

Dropout percentage determines the portion of the weights at each layer that, for each epoch, are temporarily removed from the model and inserted after training the epoch. Dropout helps to further reduce overfitting by forcing the network to learn the same features multiple times as the portions of the model that previously trained are randomly removed [32]. Increased dropout can further prevent overfitting but also introduce instability when training, justifying tuning this parameter.

α configures the degree to which the optimizer tweaks the weights at each layer to descend the gradient. A higher value results in typically faster descent but increased instability. Overshooting the target, increasing the loss, and having to backtrack is more common. Lower α can help to prevent this, but typically requires a far longer training time and may lead to the optimizer converging poorly.

When comparing these models performance, a single value was desired to prevent subjectivity. However, traditional measures like F-measure and accuracy can fail to sufficiently capture performance, especially for multiclass and imbalanced datasets as present here. To correct for this and utilize a more descriptive statistic, R_K correlation was utilized [33]. R_K is a generalization of Pearson and Matthews correlation for multiclass problems. It has been shown to be superior to Cohen's kappa and standard metrics like f-measure (the latter specifically due to its inclusion of true negatives) as a single-metric comparison tool [34].

Eq 5 displays a discretized version of R_K that allows for its usage on a single confusion matrix [33]. C_{lk} represents elements of the matrix.

$$R_K = \frac{\sum_{klm} C_{kk} C_{lm} - C_{kl} C_{mk}}{\sqrt{\sum_k \sum_l C_{kl} \sum_{\substack{l' \\ k' \neq k}} C_{k'l'}} \sqrt{\sum_k \sum_l C_{lk} \sum_{\substack{l' \\ k' \neq k}} C_{l'k'}}} \quad (5)$$

Results

720 total trials were used for training and testing. Table 3 lists the number of trials for each class label.
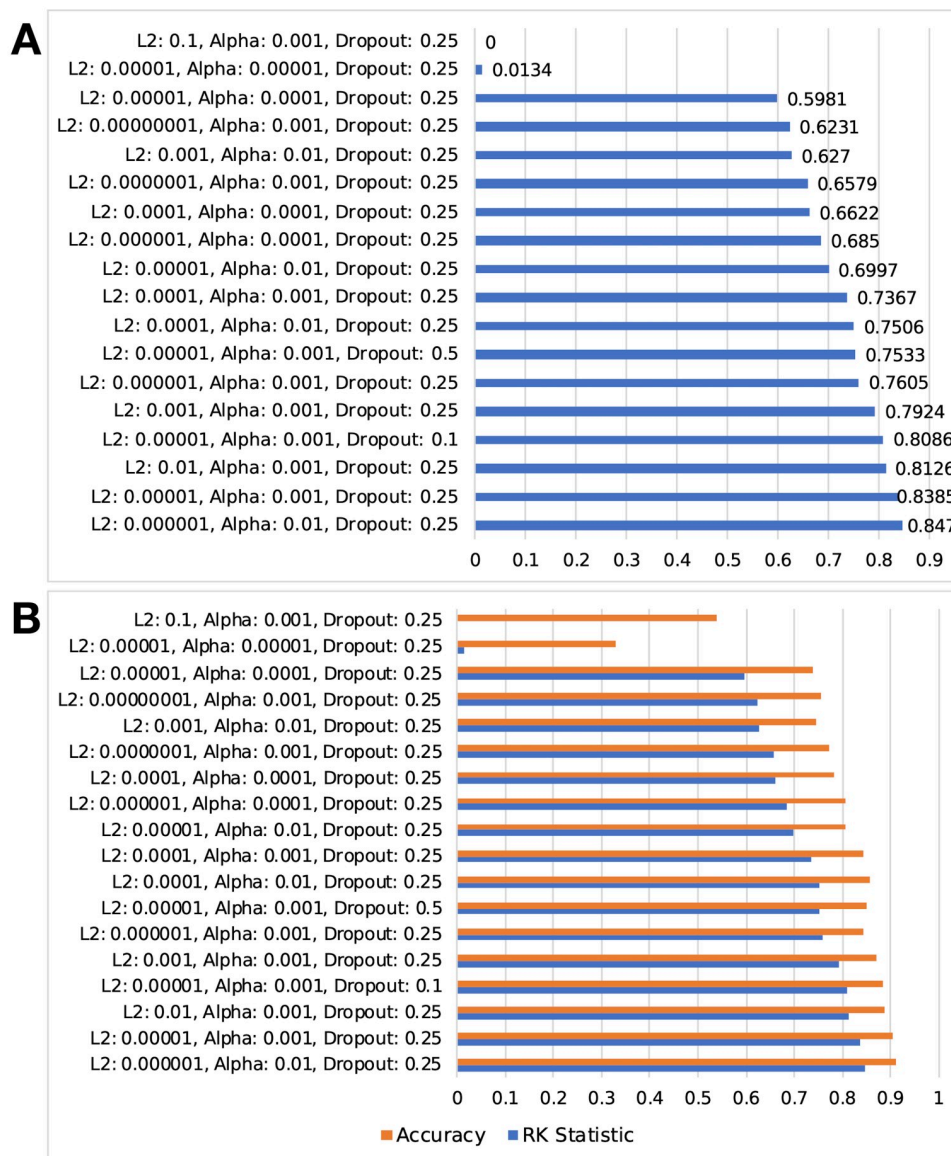
Hyperparameter results

Fig 5 displays the results of training the proposed models with different values for the stated hyperparameters. Fig 5 displays how R_K can report different results due to its consideration of true negatives compared to accuracy, making it a more well-rounded single metric for comparing various models.

However, when selecting the resulting hyperparameters, a qualitative selection of the second-best performing parameters was made. This is due to the behavior of the optimizer at such a high learning rate; it behaved in a more unstable pattern and terminated at a relatively high categorical cross-entropy compared to a slightly lower α. Fig 6 displays the behavior involved. Therefore, the parameters utilized for the proposed model and variant with max pooling are $\lambda = 10^{-5}$, $\alpha = 10^{-3}$, dropout percentage = 25%.

Table 3. Number of trials for each class of lift.

Class	Count
Low Risk	120
Medium Risk	240
High Risk	360

https://doi.org/10.1371/journal.pone.0247162.t003

Fig 5. Comparison of performance for various hyperparameters set on the proposed model. A: comparison of R_K statistics. B: R_K statistic compared with accuracy. Accuracy ranges from 0 to 1 while R_K ranges from -1 to 1. Values from -1 to 0 are not shown due to no results in that range.

https://doi.org/10.1371/journal.pone.0247162.g005

Classification results

Although various model structures were used, all relied on the same basic layer unit: the 2D convolutional layer. Models varied in complexity and depth, with some containing LSTM layers in an attempt to formally capture time-based data.

Table 4 compares precision, recall, and f-measure for the proposed model, a typical simplified CNN model, and a fully-connected multilayer perceptron network.

The modified VGGNet B model performed the best out of all models on medium and high-risk lifts and only slightly underperformed the max pooling variant in low-risk lifts. It also significantly outperformed the simplified CNN and multilayer perceptron in f-measure for all classes and in overall accuracy. Fig 7 displays the distribution of predictions for the proposed
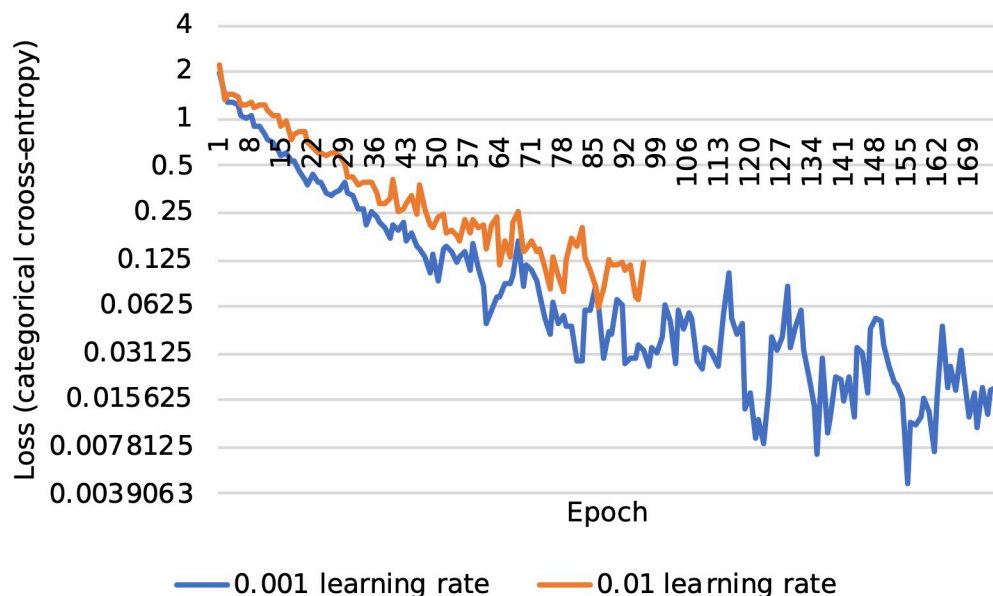
Fig 6. Loss gradient descent for α of 0.01 and 0.001. Each point represents the training categorical cross-entropy at the completion of each epoch.

model on the testing data. The results for the proposed model are displayed in more detail in Fig 8. A detailed confusion matrix of the results for the proposed model (de-normalized) is shown in Table 5.

Discussion

Small datasets are notoriously difficult to train on for machine learning models and that is borne out here, with typical CNN and MLP models barely outperforming randomness for some guesses and delivering no more than a 2/3 accuracy. The proposed VGGNet B variant significantly outperformed the other two models and max pooling variation, both in R_K (0.862) and accuracy (90.6%). Notably, it also displayed the best f-measure for every single class as well, although it improved the least in low-risk classification.

As R_K is typically only considered satisfactory above an 0.7 correlation, the proposed model and CNN+LSTM models were also the only methods that displayed acceptable performance.

Various other models were also considered as benchmark comparisons. This includes ImageNet, ResNet, DenseNet, and Microsoft's very deep image recognition network. However, these models were not deemed suitable due to their high complexity. They are primarily utilized on very large datasets with millions of samples, making them far too complex for a small dataset like the one examined.

Table 4. Summary of classification results for proposed model and alternatives.

| Risk Level	Proposed Model	CNN+LSTM	VGGNet B (Max Pool)	CNN	MLP
Low F-measure	0.776	0.8	0.746	0.444	0.367
Medium F-measure	0.862	0.783	0.692	0.635	0.500
High F-measure	0.964	0.923	0.815	0.765	0.667
Accuracy	**0.906**	0.861	0.766	0.667	0.567
RK	**0.839**	0.77	0.65	0.53	0.28
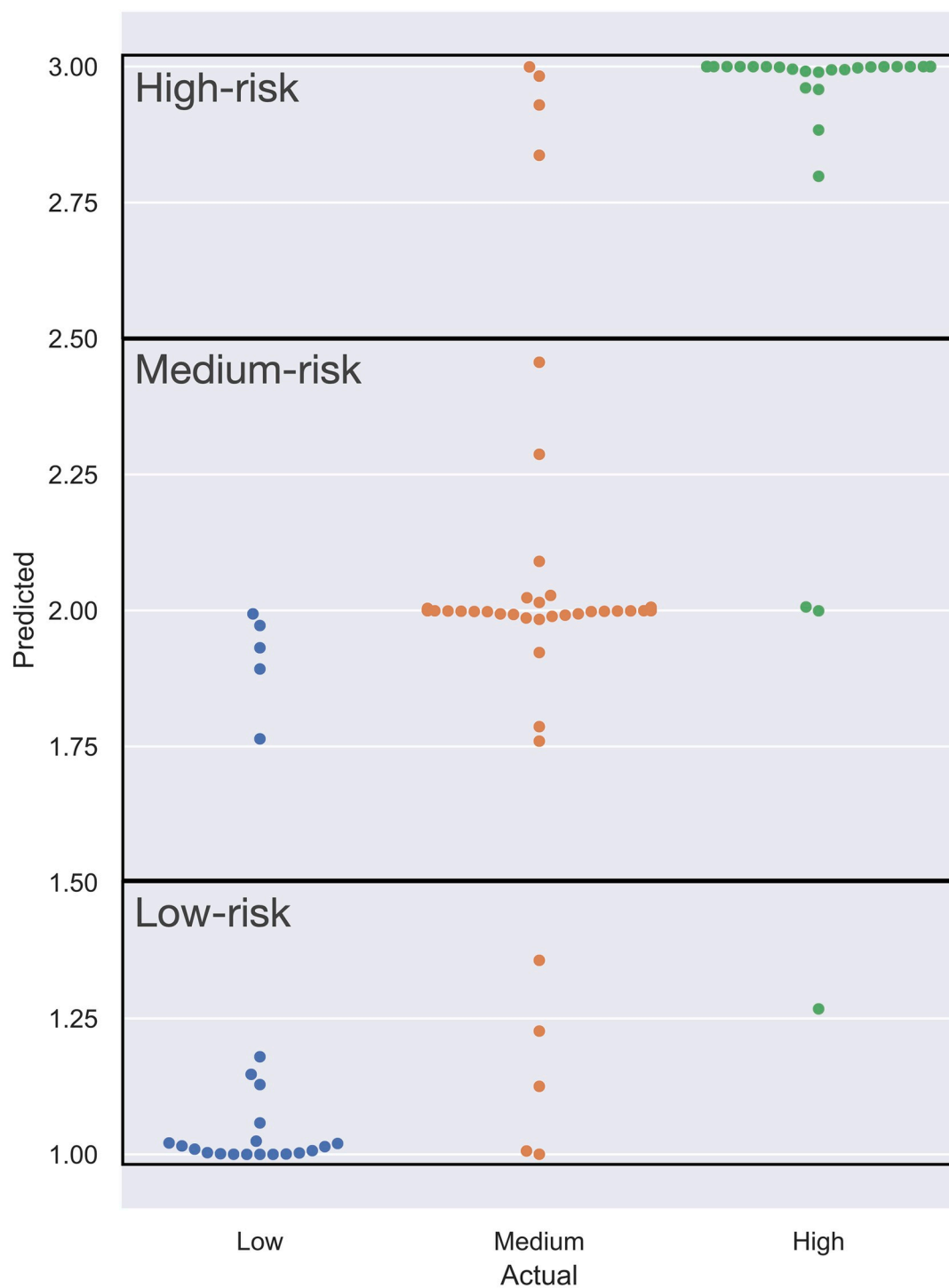
Fig 7. Swarm plot of the testing results by the proposed model. The x-axis represents the true labeling and the y-axis the model output. The y-axis is divided into three zones that define the resulting class for each value, labeled in the top-left of each box.

https://doi.org/10.1371/journal.pone.0247162.g007
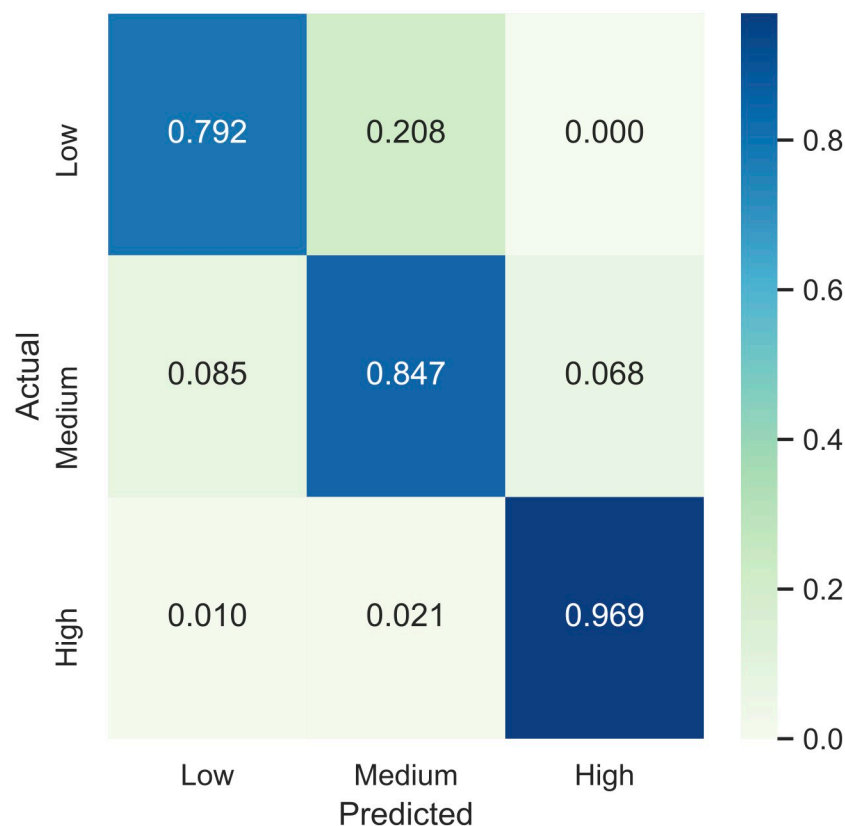
Fig 8. Heatmap plot of the testing results by the proposed model. Each row has been normalized so that each class has the same color scale.

Notably, the data trained on (see Table 3) is an example of an imbalanced dataset, with significantly varying numbers of samples in each risk level. This typically poses a challenge for machine learning models that automatically become biased toward the larger class [25]. This is obvious from examining the f-measures for all models, which increase with the number of trials for the class.

While low-risk lifts were classified more poorly than other classes, the proposed model improves upon all alternatives and helps to avoid issues such as over and under-sampling, which can lead to worse performance when testing [25].

Average pooling

Average pooling, other than parameter changes, is the major departure from VGGNet B for the model. It resulted in a 5.3 basis point increase in accuracy with major benefits to both

Table 5. Confusion matrix of the de-normalized results shown in Fig 8. The results have been scaled to a single testing set, but all folds were used in these results.

		Predicted Class		
		Low	Medium	High
Actual Class	**Low**	**19**	5	0
	Medium	5	**50**	4
	High	1	2	**94**

medium and high-risk accuracy. It is theorized that, in this case, average pooling outperforms max pooling because it passes more information to the next layer by using all 4 cells in the pool instead of selecting the highest value. This could increase generalizability of the model to testing data. However, max pooling is typically the selected model for successful CNN-based classifiers [24].

These models typically are trained on largely unprocessed image data that may have low contrast and significant redundancy in a given region, reducing the information loss of max pooling. Max pooling picking a blue pixel from the sky in an image is still representative. Average pooling is similar to various image downsampling methods, albeit simplified in its attempts to preserve information. As accelerometer data, especially at low frequencies (the dataset was recorded at 25 Hz), is vulnerable to loss of information, preserving this information may have resulted in the increase in accuracy. Use of max pooling would simulate a very rough downsampling of the data, which could clip many of the important fluctuations in rotation and acceleration from the sensor data. Average pooling smoothes out this downsampling and, while it still eliminates information, performs it less severely than max pooling.

Saliency mapping

Saliency mapping is the process of determining the input features that the model recognizes as the most significant to the output class. First defined by Simonyan et. al. in [35], saliency is a multistep process. Given a final score matrix $S_c(I)$ for a class c selected by the model, the linear score can be represented in Eq 6 as

$$S_c(I) = w_c^T I + b_c \qquad (6)$$

where I is the pixels of the image, w_c is the weights for the class, and b_c is the bias of the class. As the model is non-linear in nature due to the activation functions and so could not be easily computed, the approximation is computed instead in Eq 8.

$$S_c(I) \approx w^T I + b \qquad (7)$$

w^T is the derivative of the score matrix S_c to the image matrix I at the point I_0 (the image itself), shown in Eq 8 [35].

$$w = \left. \frac{\delta S_c}{\delta I} \right|_{I_0} \qquad (8)$$

This approximation determines how much each pixel, if changed, would affect the class score and assigns a value to them.

Here, the primary purpose was to determine whether the model was truly recognizing portions of the input as the lift or simply tweaking the weights to fit on noise, which is a possibility. However, this is not the behavior displayed in the saliency plots obtained from the proposed model.

The low and medium saliency plots (Fig 9) are nearly identical but this is not an indicator that they cannot be differentiated. Instead, the model considers the same parts of the lifting motion as important for each class. For high-risk lifts, shown in Fig 9, the model is drawing on two very specific regions, with very low weighting for the rest of the image. We theorize that it may have determined two specific points of the lift: when the object is lifted and when the lifter accelerates back to a neutral, upright position. These regions are also present on the other two saliency plots but are not as clearly delineated. This difference in quality is the clearest indicator that further improvements are possible beyond simply examining classification accuracy.
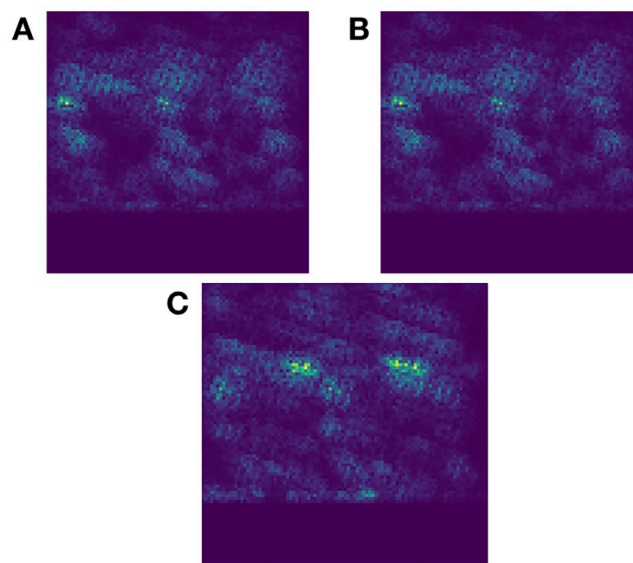
Fig 9. Saliency plots for final softmax layer of network. A: low-risk saliency. B: medium-risk saliency. C: high-risk saliency. Bright green/yellow represents the highest weighting; dark purple represents the lowest weighting.

https://doi.org/10.1371/journal.pone.0247162.g009

However, this would most likely require alterations to the structure of the model and not simple hyperparameter tuning.

Producing these plots also provides a separate benefit: examining what sensors contribute most effectively to the results. While more sensor data can assist in classification, many of the sensors have difficult or impractical placement, such as those on the thigh and upper back. On the high-risk classification, the two hot areas center around the wrist and back sensors. Side, upper arm, and thigh sensors contributed to the classification but, as shown in Fig 9, are not as bright and so are candidates for possible removal in future research.

A significant advantage of the CNN+LSTM model is, due to the data input shape, saliency analysis is far clearer here. Fig 10 displays the saliency for high-risk lifts as well, with the various sensors labeled. Here, it is far clearer that the back and wrist sensors are the most significant. While all sensors contribute to the prediction (in at least one dimension), removing the thigh, upper arm, and side sensors may still leave enough information to sufficiently classify lifting risk level.

Additionally, we see two general peaks here as well, suggesting that lifting behavior may have an initial acceleration and final acceleration as general features. As both models appear to be examining the same region of the data, it is increasingly likely that the data is sufficiently separable and contains true features instead of simple noise.

Feature extraction

Many other examples of accelerometer classification employ manual feature extraction with various measurements including means, variances, zero-crossing rates, and various other statistics [36].

This method only requires a signal filter and standardization before reshaping and feeding to the model. This could be performed very quickly and benefits from significant previous efforts in signal processing to allow streamed and real-time transformations. While many more features are utilized, handcrafted features typically require significant domain experience
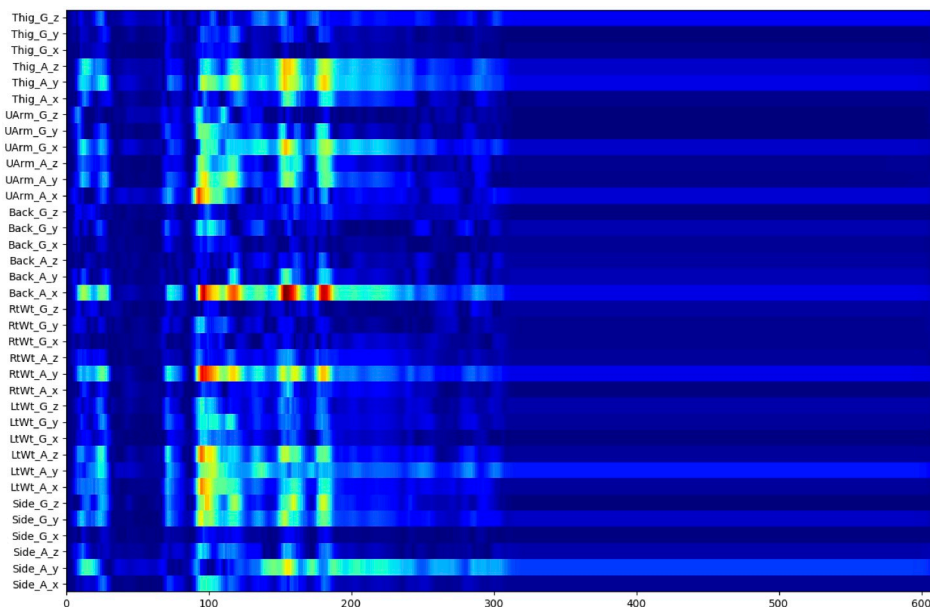
Fig 10. Saliency plot for high-risk lift trials obtained from the CNN+LSTM model. Scale ranges from deep blue as the lowest significance and deep red as the highest significance. The x-axis is frames of the input data and y-axis is the sensor data, where A/G is accelerometer or gyroscope and x, y, z are the dimensions for the sensor.

and tuning for high performance. The preprocessing proposed could easily be applied to many different datasets with minimal alterations.

Possible improvements and future research

90.6% accuracy, while a significant improvement over other models, still provides opportunity for further advancement. It is doubtful that hyperparameter tweaking could significantly increase accuracy, as Fig 5 displays only small increases once 85% accuracy was reached. Model alterations are most likely necessary to reach overall 95% testing accuracy. However, high-risk classification, considered most important to ensure worker safety, is excellent, with 96.9% accuracy.

One possible region of interest is configuring the number of convolutional layers between the average pooling layers. Deeper into the network, additional convolutional layers may provide high-level feature extraction. Nearer to the input layer, more layers typically increase recognition of granular details. However, as stated in the introduction, a high number of parameters for a small dataset such as this can result in the optimizer failing to converge. Therefore, simply adding layers may fail to significantly increase accuracy and require alterations to the general network structure as well.

The presence of a fully-connected layer at the end of the network is also a point of interest. This layer especially provides a significant number of parameters and so is a target for optimization. Altering the number of units and configuring a regularizer on that layer may assist, especially on improving low-risk lifts due to optimizing for a less complex model.

For real-world use, minimizing the number of sensors will significantly advance the practicality, reducing cost and eliminating the awkward placement of several sensors. Ideally, one sensor would provide sufficient data to classify lifting. However, the two wrist sensors and a side sensor are simple enough to attach that they may be an acceptable alternative. The strong emphasis, unfortunately, by both models on the back sensor may indicate that this sensor is

necessary. Redundancies may nonetheless exist in the sensor data and may provide sufficient separability for a similar model.

Finally, based on the saliency maps generated, the window size for the lift could be significantly reduced with, in all likelihood, minimal reduction in accuracy. This would increase the speed of activity recognition in real-world use simply due to the reduced input size. A reduction in input size also decreases the number of parameters in the network by eliminating features, possibly allowing for further depth in the network. However, overzealous input size reduction could cut off significant parts of lifts and so needs to be performed carefully.

Dataset applicability

Data collection of the set used to train the proposed model is further described in [13]. Notably, the object lifted—an 0.45 kg wire grid—was designed to minimize the impact of any possible injury from lifting incorrectly. Incorrectly, here, is lifting from *any* of the medium- or high-risk zones. In the case of high-risk zones, there is no method of lifting significant weight that will not contribute to injury. Therefore, the tasks are not perfectly realistic due to it only approximating the real-world lifting behavior seen in the workplace.

This is a limitation of the dataset and could be improved with additional variation in data collection with future studies. Additionally, the generalizability of the model could be further studied with additional variation and more trials to better determine its adaptability to new data.

Conclusion

Classifying accelerometer data is traditionally difficult due to the requirement of most machine learning models requiring large datasets. Therefore, much of the existing research focuses on typical activity and exercise classification that can draw on existing datasets or be compiled from many subjects. Specialized activities with lesser impact, then, have been neglected due to the difficulty involved in compiling enough data for traditional models.

The proposed model was able to quickly and accurately (90.6% accuracy, 0.839 R_K) classify a small accelerometer dataset provided by NIOSH with minimal feature extraction and significantly greater performance than other models tested. Specifically, the usage of a CNN that would normally classify images along with the alteration to use average pooling over max pooling provided the greatest benefit. Hyperparameter tuning was also shown to have significant effects on the performance, but in lower magnitudes. It is very likely that a similar model could be trained on other small and/or unbalanced datasets to make their classification feasible where other models have failed. The proposed technique provides excellent monitoring of risks involved in the various types of manual lifts in an industrial setup to provide timely interventions.

Supporting information

S1 Data.
(ZIP)

Acknowledgments

Author Contributions

Conceptualization: Kristian Snyder, Ming-Lun Lu, Rashmi Jha.

Data curation: Kristian Snyder, Ming-Lun Lu, Menekse S. Barim, Marie Hayden, Dwight Werren.

Formal analysis: Kristian Snyder, Menekse S. Barim, Marie Hayden, Dwight Werren.

Funding acquisition: Ming-Lun Lu, Rashmi Jha.

Investigation: Kristian Snyder, Brennan Thomas, Menekse S. Barim, Marie Hayden, Dwight Werren.

Methodology: Kristian Snyder, Brennan Thomas.

Project administration: Ming-Lun Lu, Rashmi Jha.

Resources: Menekse S. Barim, Marie Hayden, Dwight Werren.

Software: Kristian Snyder, Brennan Thomas.

Supervision: Ming-Lun Lu, Rashmi Jha.

Validation: Kristian Snyder, Brennan Thomas, Ming-Lun Lu, Rashmi Jha.

Visualization: Kristian Snyder, Brennan Thomas.

Writing – original draft: Kristian Snyder.

Writing – review & editing: Kristian Snyder, Ming-Lun Lu, Rashmi Jha.

References

1. Hoy D, Bain C, Williams G, March L, Brooks P, Blyth F, et al. A systematic review of the global prevalence of low back pain. Arthritis and Rheumatism. 2012; 64(6):2028–2037. https://doi.org/10.1002/art.34347 PMID: 22231424

2. Bernhard BP. A critical review of epidemiological evidence for work-related musculoskeletal disorders of the neck, upper extremity and low back. Musculoskeletal disorders and workplace factors. 1997;.

3. Council NR, Others. Musculoskeletal disorders and the workplace: low back and upper extremities. National Academies Press; 2001.

4. Liberty Mutual Research Institute for Safety. Liberty Mutual Workplace Safety Index. Liberty Mutual Insurance; 2018. Available from: https://business.libertymutualgroup.com/business-insurance/Documents/Services/WorkplaceSafetyIndex.pdf.

5. Waters T, Putz-Anderson V, Garg A. Applications Manual for the Revised NIOSH Lifting Equation. DHHS (NIOSH). 1994; 94-110:1–164.

6. Waters TR, Lu ML, Piacitelli LA, Werren D, Deddens JA. Efficacy of the revised NIOSH lifting equation to predict risk of low back pain due to manual lifting: Expanded cross-sectional analysis. Journal of Occupational and Environmental Medicine. 2011; 53(9):1061–1067. https://doi.org/10.1097/JOM.0b013e31822cfe5e

7. Dempsey PG, McGorry RW, Maynard WS. A survey of tools and methods used by certified professional ergonomists. Applied Ergonomics. 2005; 36(4 SPEC. ISS.):489–503. https://doi.org/10.1016/j.apergo.2005.01.007

8. Music J, Stancic I, Zanchi V. Is it possible to detect mobile phone user's attention based on accelerometer measurment of gait pattern? Proceedings—International Symposium on Computers and Communications. 2013; p. 522–527. https://doi.org/10.1109/ISCC.2013.6754999

9. Liu Y, Nie L, Liu L, Rosenblum DS. From action to activity: Sensor-based activity recognition. Neurocomputing. 2016; 181:108–115. https://doi.org/10.1016/j.neucom.2015.08.096

10. Bao L, Intille SS. Activity recognition from user-annotated acceleration data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2004; 3001:1–17. https://doi.org/10.1007/978-3-540-24646-6_1

11. Preece SJ, Goulermas JY, Kenney LPJ, Howard D, Meijer K, Crompton R. Activity identification using body-mounted sensors—A review of classification techniques. Physiological Measurement. 2009; 30 (4). https://doi.org/10.1088/0967-3334/30/4/R01 PMID: 19342767

12. Mayer JM, Mooney V, Matheson LN, Erasala GN, Verna JL, Udermann BE, et al. Continuous Low-Level Heat Wrap Therapy for the Prevention and Early Phase Treatment of Delayed-Onset Muscle Soreness of the Low Back: A Randomized Controlled Trial. Archives of Physical Medicine and Rehabilitation. 2006; 87(10):1310–1317. https://doi.org/10.1016/j.apmr.2006.07.259 PMID: 17023239

13. Barim MS, Lu ML, Feng S, Hughes G, Hayden M, Werren D. Accuracy of An Algorithm Using Motion Data Of Five Wearable IMU Sensors For Estimating Lifting Duration And Lifting Risk Factors. Proceedings of the Human Factors and Ergonomics Society Annual Meeting. 2019; 63(1):1105–1111. https://doi.org/10.1177/1071181319631367

14. Weiss GM, Lockhart JW, Pulickal TT, McHugh PT, Ronan IH, Timko JL. Actitracker: A smartphone-based activity recognition system for improving health and well-being. Proceedings—3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016. 2016; p. 682–688. https://doi.org/10.1109/DSAA.2016.89

15. Cook D, Feuz KD, Krishnan NC. Transfer learning for activity recognition: a survey. Knowledge and Information Systems. 2013; 36(3):537–556. https://doi.org/10.1007/s10115-013-0665-3

16. Jalal A, Kim YH, Kim YJ, Kamal S, Kim D. Robust human activity recognition from depth video using spatiotemporal multi-fused features. Pattern Recognition. 2017; 61:295–308. https://doi.org/10.1016/j.patcog.2016.08.003

17. Kuspa K, Pratkanis T. Classification of Mobile Device Accelerometer Data for Unique Activity Identification; 2013. Available from: http://cs229.stanford.edu/proj2013/PratkanisKuspa-ClassificationOfMobileDeviceAccelerometerDataforUniqueActivityIdentification.pdf.

18. Kwapisz JR, Weiss GM, Moore SA. Activity recognition using cell phone accelerometers. ACM SIGKDD Explorations Newsletter. 2011; 12(2):74. https://doi.org/10.1145/1964897.1964918

19. Maurer U, Smailagic A, Siewiorek DP, Deisher M. Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. In: International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06). IEEE; 2006. p. 113–116. Available from: http://ieeexplore.ieee.org/document/1612909/.

20. Hammerla NY, Halloran S, Plötz T. Deep, convolutional, and recurrent models for human activity recognition using wearables. IJCAI International Joint Conference on Artificial Intelligence. 2016; 2016-Janua:1533–1540.

21. Erer KS. Adaptive usage of the Butterworth digital filter. Journal of Biomechanics. 2007; 40(13):2934–2943. https://doi.org/10.1016/j.jbiomech.2007.02.019

22. Kwon S, Lee J, Chung GS, Park KS. Validation of heart rate extraction through an iPhone accelerometer. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE; 2011. p. 5260–5263.

23. Mayagoitia RE, Nene AV, Veltink PH. Accelerometer and rate gyroscope measurement of kinematics: An inexpensive alternative to optical motion analysis systems. Journal of Biomechanics. 2002; 35 (4):537–542. https://doi.org/10.1016/S0021-9290(01)00231-7

24. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings. 2015; p. 1–14.

25. Kotsiantis S, Kanellopoulos D, Pintelas P. Handling imbalanced datasets: A review. Science. 2006; 30 (1):25–36. https://doi.org/10.1007/978-0-387-09823-4_45

26. Nagi J, Ducatelle F, Di Caro GA, Cireşan D, Meier U, Giusti A, et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition. 2011 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011. 2011; p. 342–347.

27. Olson M, Wyner AJ, Berk R. Modern neural networks generalize on small data sets. Advances in Neural Information Processing Systems. 2018; 2018-December(NeurIPS):3619–3628.

28. Ordóñez FJ, Roggen D. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors (Switzerland). 2016; 16(1). https://doi.org/10.3390/s16010115 PMID: 26797612

29. Sak H, Senior A, Beaufays F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. INTERSPEECH-2014. 2014; p. 338–342.

30. Kingma DP, Ba JL. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings. 2015; p. 1–15.

31. Cortes C, Mohri M, Rostamizadeh A. L2 Regularization for Learning Kernels. Optimization. 2012;.

32. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. 2012; p. 1–18.

33. Gorodkin J. Comparing two K-category assignments by a K-category correlation coefficient. Computational Biology and Chemistry. 2004; 28(5-6):367–374. https://doi.org/10.1016/j.compbiolchem.2004.09.006

34. Delgado R, Tibau XA. Why Cohen's Kappa should be avoided as performance measure in classification. PLoS ONE. 2019; 14(9):1–26. https://doi.org/10.1371/journal.pone.0222916

35. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. 2nd International Conference on Learning Representations, ICLR 2014 —Workshop Track Proceedings. 2014; p. 1–8.

36. Arif M, Kattan A. Physical activities monitoring using wearable acceleration sensors attached to the body. PLoS ONE. 2015; 10(7):1–16. https://doi.org/10.1371/journal.pone.0130851