Article

Not peer-reviewed version

CLUMM: Contrastive Learning for Unobtrusive Motion Monitoring

Pius Gyamenah , Hari Iyer , Heejin Jeong , Shenghan Guo *

Posted Date: 27 November 2024

doi: 10.20944/preprints202411.2059.v1

Keywords: Self-supervised learning; Unobtrusive human sensing; Contrastive learning; In-situ monitoring; Motion recognition

Article

CLUMM: Contrastive Learning for Unobtrusive Motion Monitoring

Pius Gyamenah [1], Hari Iyer [2], Heejin Jeong [2] and Shenghan Guo [1,*]

[1] The School of Manufacturing Systems and Networks, Ira. A. Fulton Schools of Engineering, Arizona State University
[2] The Polytechnic School, Ira. A. Fulton Schools of Engineering, Arizona State University
* Correspondence: shenghan.guo@asu.edu

Abstract: Traditional approaches for human monitoring and motion recognition often rely on wearable sensors, which, while effective, are obtrusive and cause significant discomfort to workers. More recent approaches have employed unobtrusive, real-time sensing using cameras mounted in the manufacturing environment. While these methods generate large volumes of rich data, they require extensive labeling and analysis for machine learning applications. Additionally, these cameras frequently capture irrelevant environmental information, which can hinder the performance of deep learning algorithms. To address these limitations, this paper introduces a novel framework that leverages a contrastive learning approach to learn rich representations from raw images without the need for manual labeling. The framework mitigates the effect of environmental complexity by focusing on critical joint coordinates relevant to manufacturing tasks. This approach ensures the model learns directly from human-specific data, effectively reducing the impact of the surrounding environment. A custom dataset of human subjects simulating various tasks in a workplace setting is used for training and evaluation. By fine-tuning the learned model for a downstream motion classification task, we achieve up to 90 % accuracy, demonstrating the effectiveness of our proposed solution in real-time human motion monitoring.

Keywords: self-supervised learning; unobtrusive human sensing; contrastive learning; in-situ monitoring; motion recognition

1. Introduction

Human motion recognition is a task that identifies human motion from sensor data with significant implications across domains such as healthcare [1,2], sports [3], and manufacturing [4], where it is essential to understand human behavior from their motion patterns. For instance, in the manufacturing domain, it can provide valuable insights for understanding worker behavior, identifying potential safety risks, assessing workers, detecting ergonomic issues, and identifying areas where further training may be needed. Despite this potential, monitoring and analyzing human motion requires significant effort in sensing and analysis [5], especially in complex environments with many moving parts.

Most existing sensor-based human motion recognition efforts have utilized wearable sensors worn directly on body parts [6]. While wearable sensor technologies have advanced significantly, they are intrusive and may need to be adjusted to different user heights and sizes. Additionally, multiple wearable sensors must be used simultaneously in many cases, which can be cumbersome, cause discomfort, and reduce worker productivity [6–8].

Advancements in camera technology and image processing have paved the way for unobtrusive in-situ monitoring. Comparative studies from the literature show that camera-based approaches offer a less invasive and more comprehensive solution [9,10]. These camera-based approaches can capture detailed motion data without interfering with human activities, thus facilitating real-time monitoring and analysis. However, analyzing these large streams of real-time data comes with significant data-level challenges. Recent advances in deep learning (DL) have paved the way for more accurate

 doi:10.20944/preprints202411.2059.v1

human activity recognition from sensor data [8–12]. Deep learning approaches such as neural networks (NNs) can extract features directly from input data, thus providing end-to-end learning on raw sensor data without extensive preprocessing. Nevertheless, their performance heavily depends on large volumes of labeled data [13]. Unobtrusive camera-based sensors provide large data streams, but assigning activity labels individually to these large streams of sensor data is costly and labor-intensive and requires significant domain expertise, posing significant barriers to the efficiency and scalability of existing solutions. Thus, there is a need for label-efficient approaches to address this data-level bottleneck. Additionally, the dynamic environments in which human motion is monitored challenge the effectiveness of learning algorithms due to various factors such as occlusions, noise, varying lighting conditions, and irrelevant objects in the scene [7]. These factors can lead to the Clever Hans phenomenon [14], where a model performs well but relies on irrelevant data (spurious correlations) instead of learning the task of interest. This phenomenon can manifest in neural-network-based worker motion monitoring from image data. An algorithm may associate irrelevant features, such as background machinery and lighting conditions, with the motion task instead of focusing on human movements. This problem affects the generalizability of machine learning solutions to complex environments.

Recently, label-efficient approaches such as self-supervised learning (SSL), which learn representations directly from unlabeled data, have been proposed to overcome the limitations posed by the lack of labels [15,16]. SSL approaches such as contrastive learning (CL), which leverage instance discrimination to bring similar instances closer in the representation space while pushing dissimilar instances apart, have demonstrated superior performance across various modalities with robust generalizability compared to supervised approaches [16]. Additionally, they are less susceptible to spurious correlations and adversarial examples. Due to their label efficiency, robustness to variations, and generalizability, these approaches hold immense potential for unobtrusive human motion recognition tasks. However, to the best of our knowledge, they have not been explored enough for unobtrusive motion recognition. Existing self-supervised approaches to human motion recognition have focused on wearable sensors [17,18].

To enable label-efficient human motion recognition from unobtrusive sensing data, we introduce **Contrastive Learning for Unobtrusive Motion Monitoring (CLUMM),** a contrastive SSL-based framework for unobtrusive human recognition, leveraging the robust feature extraction and generalization capabilities of SSL to learn representations directly from unlabeled camera data. As shown in Figure 1, we extract skeletal coordinates from image frames of human videos and learn representations from them without manual data labeling. We show the effectiveness of the proposed approach by fine-tuning it on a small dataset of labeled human motion data. CLUMM addresses the data-level challenges in existing unobtrusive motion recognition methods as follows:

- **Joint tracking by Computer Vision (CV).** We remove the effect of the complex operational environment by directly extracting the coordinates of specific joints from the human body using CV techniques, specifically MediaPipe pose (MPP) [19]. MPP is an open-source framework developed by Google for estimating high-fidelity 2D and 3D coordinates of body joints. It uses BlazePose [20], a lightweight pose estimation network, to detect and track 33 3D body landmarks from videos or images. From the body landmarks identified by MPP, we select key landmarks based on the inputs of ergonomic experts to formulate the initial features significant to various motion types. This method of extracting joint information also preserves privacy by learning from joint coordinates instead of raw image data. Additionally, the MPP is scale and size invariant [19], which enables it to handle variations in human sizes and height.
- **SimCLR feature embedding.** We address the data bottleneck using a contrastive SSL approach to directly learn representations from camera data without requiring extensive manual labeling. We specifically use SimCLR [21] an SSL method that learns features by maximizing agreement between different augmented views of the same sample using a contrastive loss. We use SimCLR to learn embeddings and identify meaningful patterns and similarities within the extracted joint data depicting various motion categories. The learned representations are further leveraged in a downstream task to identify specific motion types.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 November 2024 doi:10.20944/preprints202411.2059.v1

- **Classification for motion recognition and anomaly detection.** Lastly, we leverage the learned representations from the CL training for a downstream classification task involving different motion categories. We train a simple logistic regression model on top of our learned representations to identify different motion categories in a few-shot learning [22] setting. This demonstrates the robustness and generalizability of our learned representations to downstream tasks. Additionally, we perform outlier analysis by evaluating the ability of our framework to identify out-of-distribution data. We introduce different amounts of outliers with varying deviations from the classes of interest and measure the ability of our framework to identify these outliers and the effect of outliers on the discriminative ability of our framework.

The proposed CLUMM contributes to the methodology and application of unobtrusive human motion monitoring. Methodologically, CLUMM will contribute a label-efficient machine learning approach to recognize motion types from unobtrusive human sensor data. CLUMM's ability to remove the effects of the complex environment will make it applicable to motion monitoring in such environments. Furthermore, CLUMM will contribute to anomaly detection and outlier analysis across various motion analysis tasks due to its robustness to outliers, demonstrated in our case study.

Practically, CLUMM is highly useful for worker motion analysis. By integrating MPP as its joint feature extraction component, which is then connected with SimCLR and motion classification, CLUMM achieves improved feature extraction, adaptability to different downstream tasks, and reduced manual labeling effort. Additionally, CLUMM is robust to outliers and can capture other motion types inherent in the training data. Our approach to worker motion analysis presents a step toward effective and efficient human motion analysis. In a case study, we show the effectiveness of our solution by fine-tuning domain-specific data involving various task categories in a controlled laboratory environment. Using a few labeled examples, CLUMM outperforms transfer learning performance on a baseline ResNet model, demonstrating the superiority of the learned representations as a feature extractor for other tasks in a similar domain.

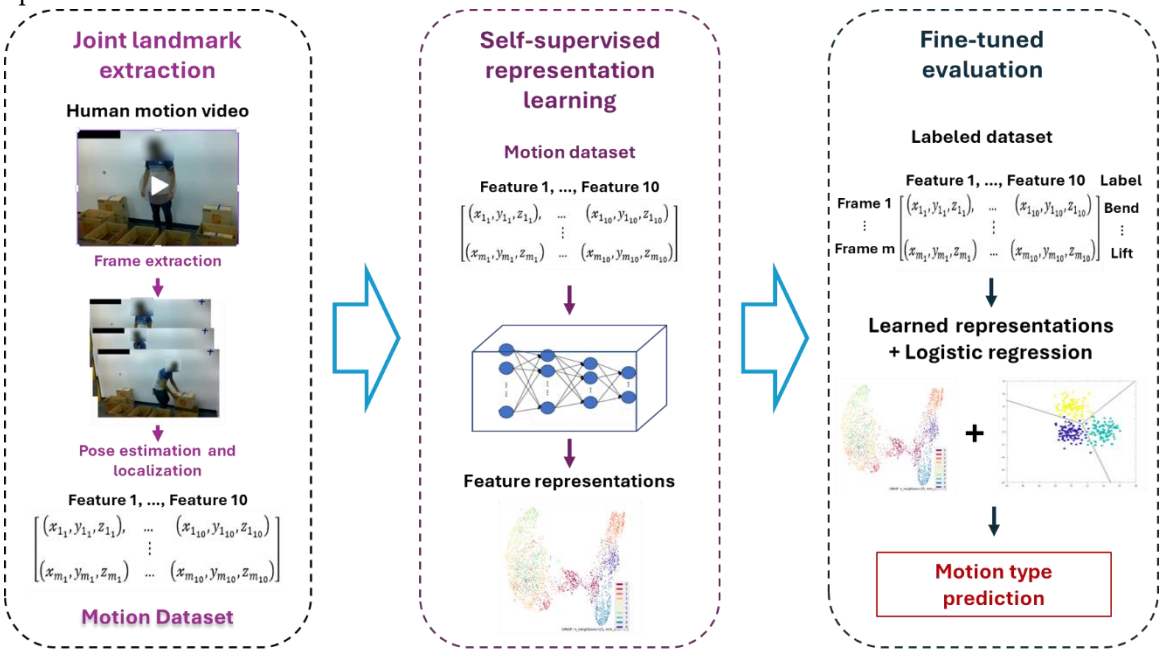

Figure 1. Architecture Pipeline for Human Motion Analysis.

The rest of this paper is organized as follows. Section 2 provides an overview of state-of-the-art literature. Section 3 elaborates on the technical details of our proposed methodology. Section 4 presents a case study using domain-specific data from manufacturing tasks in a laboratory environment. Section 5 provides a discussion of our results and highlights future directions. Finally, Section 6 concludes the work.

2. Literature Review

 doi:10.20944/preprints202411.2059.v1

Human motion recognition is a task that identifies human motion automatically from sensor data [12,23]. This rapidly growing field is significant to domains such as manufacturing, where it is necessary to understand human behavior, assess workers, provide additional training, and suggest ergonomic improvements [5]. Human motion recognition involves analyzing collected sensor data to identify and classify various motion categories [25,26]. Sensors used in the literature to collect human motion data are classified as either obtrusive or unobtrusive [6]. Obtrusive sensing involves wearable sensors such as gyroscopes, accelerometers, thermometers, etc. [24–26]. These wearable sensors are invasive, uncomfortable, and may affect productivity [10]. Additionally, users may forget to wear them [8,29]. On the contrary, unobtrusive sensing is carried out in a non-invasive manner without the need for wearable sensors. This is done by integrating non-wearable sensors in the environment as naturally as possible without direct contact with users [6].

Machine learning and deep learning approaches have been utilized to analyze sensor data using supervised and unsupervised methods. Supervised approaches require large volumes of annotated data, which is time and labor-intensive, prompting the need for label-efficient approaches [29] such as self-supervised learning [15,16]. This section reviews existing work in unobtrusive human motion sensing and self-supervised learning. It highlights the methodology gaps and provides justifications for our proposed methodology.

2.1. Unobtrusive Human Motion Monitoring

Unobtrusive sensing makes sensors as invisible as possible by blending sensors into the natural environment. This invisibility enables users to perform their activities unobtrusively and non-invasively [6]. Radio Frequency Identification (RFID) [4,29], thermal cameras [8,11,31], millimeter wave (mmWave) radar [31], LiDAR [20], and combinations of different non-wearable sensors with multimodal data [32] have been used in the literature for recognizing different motion categories across various domains. Beyond comfort, unobtrusive sensing using nonwearable sensors blended in the environment provides advantages such as broader coverage (cameras), consistency, and reduced signal noise [9,34,35]. Additionally, no user training on using the sensor is needed.

In recent years, motion sensor analysis has seen substantial improvements, reflecting various methodological approaches that leverage traditional rule-based approaches, classical machine learning, deep learning, and a mixture of preprocessing and deep learning.

Early approaches to human motion analysis primarily utilized rule-based mathematical and interpolation approaches based on domain expertise to analyze and classify various motion categories [35,36]. These methods involved the application of numerical algorithms to process and interpret sensor data. For example, Newaz et al. [35] used interpolation and mathematical measures to analyze data obtained from an array of low-resolution thermal sensors. Results from their study demonstrated the effectiveness of these rule-based approaches, showing superior performance to some machine learning (ML) approaches. In [36], the authors leveraged skeleton coordinates from a Microsoft Kinect sensor for classifying activities such as standing, sitting, lying, and falling. They performed handcrafted feature extraction using depth, height, velocity, acceleration, and angle between joints to classify various action categories. They set a threshold for determining various classes of activities and manually compute motion categories by comparing extracted features against the given thresholds for each action type. While these rule-based approaches have been proven to work with more straightforward datasets and tasks, they depend solely on predefined rules, which are prone to errors and may fail to capture complex patterns and relationships in complex sensor data where explicit rules may not be easily inferred. Additionally, they depend heavily on domain expertise, which may not always be readily available. Nevertheless, these approaches can be starting points for more efficient ML/DL approaches. For instance, the handcrafted features from [36] can serve as features for DL algorithms.

Several works in motion recognition have utilized classical machine learning approaches such as hidden Markov models, support vector machines (SVM), k nearest neighbors (KNNs), and ensemble methods such as random forests (RF) [16,18,26–29] to classify and predict motion patterns from sensor data. These approaches use manual feature extraction to extract relevant motion features

 doi:10.20944/preprints202411.2059.v1

and use them as inputs to train predictive models. In [39], the authors used a maximum entropy Markov model (MEMM) for activity recognition using a modified Viterbi algorithm to model the most probable activity sequences after preprocessing and feature extraction with optical flow and stepwise linear discriminant analysis. They modeled the activity states as states of the MEMM model and identified the next activity sequence based on video input. Liu et al. [4] use a similar Markov approach to predict the next probable motion sequence in a human-robot collaborative assembly task. KNNs, RFs, and SVMs are used in [30] on data from an array of thermopile sensors to classify 15 human motion categories. Their results demonstrate the effectiveness of classical ML approaches.

Despite their efficacy, these classical ML approaches often need manually extracted handcrafted features from sensor data and activity labels as inputs to classifiers. They require extensive domain knowledge for feature engineering and may struggle with high-dimensional data and complex motion patterns [12]. Unlike classical machine learning approaches, deep learning approaches such as neural networks can extract features directly from input data, thus providing end-to-end learning on raw sensor data without extensive preprocessing. Additionally, deep learning approaches are scalable to large volumes of sensor data and adaptable to different scenarios, making them suitable for the motion recognition task [41]. Rezaei et al. [8] show the effectiveness of deep learning approaches by comparing the results of applying ML and DL algorithms to data from an array of low-resolution infrared cameras. The infrared cameras were mounted on an experimental room's side wall and ceiling. The data in stereo images were used as ground truths for ML and DL algorithms. Their results showed the superiority of deep learning methods over traditional ML approaches. Additionally, they show the advantage of sensor fusion in this domain by comparing the results from a single sensor and a fusion of multiple infrared sensors at different locations. Multiple Comparative studies [31,42] have also shown the superiority of DL approaches for motion recognition. Convolutional neural networks (CNNs) and LSTMs have particularly demonstrated remarkable performance and have been used extensively in recent work [7–11,30,41–44], have been used extensively in prior work with impressive recognition performance showing their ability to capture spatiotemporal dependencies in motion data.

Recent advancements have explored hybrid approaches that combine preprocessing techniques with deep learning models. These approaches employ traditional preprocessing steps such as denoising, voxelization, and dimensionality reduction to enhance sensor data quality before feeding them into deep learning models. For instance, Singh et al. [31] used a mixed approach using a preprocessing approach and ML and DL approaches. They process the raw data from an mmWave radar using a sliding window to gather point clouds and convert them into voxels. These voxels are used as inputs for different classifiers. They evaluate the performance of various classifiers such as SVMs, Multi-layer Perceptron (MLP), Bi-Directional LSTM, and Time-Distributed CNN + Bi-directional LSTM on the processed inputs and find that the Time-Distributed CNN + LSTM performs the best. Similarly, Yu et al. [22] used DBSCAN to remove noise from point clouds. They conducted voxelization and augmented the voxelated input before feeding it into a dual-view CNN for end-to-end learning. These integrations demonstrate the effectiveness of harnessing the strengths of preprocessing methods and advanced deep-learning algorithms to achieve motion recognition performance.

These hybrid approaches have been particularly transformative in camera-based motion analysis. In this approach, information about the positions and movement of body parts extracted from pose estimation models is input into deep learning models. Utilizing this joint information allows for a more coherent representation of motion dynamics as it captures the intricate relationships between body parts and their movements over time [46]. In [47], the authors classify activities using the Euclidean distance of 3D joint coordinates in consecutive frames as input to a CNN for activity classification.

A comprehensive comparative study by Açıs et al. [42] sought to evaluate training effectiveness using raw RGB data from a Kinect sensor versus utilizing pose coordinates. The study assessed the performance of an LSTM feature extractor and compared it with feature extraction using a CNN on joint coordinates. Additionally, the study compared the effectiveness of training a CNN from scratch

 doi:10.20944/preprints202411.2059.v1

and using three transfer learners (Densenet201 [48]], Xception [49], and Resnet50 [50]) on raw RGB images. The comparative evaluation results indicated that utilizing joint coordinates and an LSTM feature extractor yielded the best accuracy. Specifically, joint data with a CNN from scratch resulted in 72% accuracy. In comparison, joint data with LSTM achieved 98% accuracy, and the best transfer learner on raw RGB images produced an accuracy of 60%. Thus, the study concluded that utilizing joint information for training yielded significantly better results than training on raw RGB images. Inspired by the results from [42], this study utilizes pose information as input for feature extraction.

Despite the advantages of deep learning-based approaches to the motion recognition task, their performance heavily depends on large volumes of labeled data [13], which is time-consuming and challenging to annotate. We seek to address this data-level challenge by learning representations directly from unlabeled data without relying on class labels. This is crucial as labeling motion data is labor-intensive and time-consuming, especially for diverse and complex motions. Additionally, human-subject data is expensive to gather in practice [1], requiring remunerations and IRB approvals in some cases. These limit the data available for training these DL algorithms for new sensor data. Thus, there is a need for label-efficient and adaptive approaches to learning robust representations that can be used as transfer learners to fine-tune small sensor data. We address this by leveraging SimCLR, a contrastive self-supervised learning framework, to learn representations directly from unlabeled data. The framework can learn robust representations that can be used for downstream tasks in human motion-related tasks on sensor data.

2.2. Self-Supervised Learning

As discussed in Section 2.1, existing approaches to human motion recognition face significant challenges due to their reliance on large volumes of labeled data, which is expensive to collect and requires a substantial manual labeling effort and domain expertise. Self-supervised learning (SSL) emerges as a strong alternative to address this data bottleneck.

Self-supervised Learning (SSL) is a machine learning paradigm that learns representations directly from unlabeled data without relying on corresponding class labels [15]. SSL addresses the data bottleneck of supervised learning approaches, thereby alleviating the reliance on large, labeled datasets. SSL has recently received much attention and contributed significantly to advances in natural language processing (NLP) and CV [16]. Recent work has shown the effectiveness of SSL approaches across different data modalities such as text, audio, video, and time series [46–48]. SSL leverages unlabeled inputs to define a pretext task, which it uses to learn representations [54]. These representations capture the internal relationship between inputs, which can be fine-tuned for downstream tasks [54]. In contrast to supervised learning approaches, where representations are task-specific, the representations learned by SSL are task-agnostic [15], making the learned representations adaptable to other tasks. Additionally, SSL boasts of better generalizability and robustness to spurious correlations and adversarial attacks than supervised learning [15,16]

Existing approaches to SSL employ different strategies to accomplish the pretext task, which involves learning supervisory signals from unlabeled data. Context-based approaches [15] utilize the intrinsic contextual relationships within the inputs, such as color and rotations, as supervisory signals. These approaches train models to understand the positions and orientations of objects within a scene [16]. On the other hand, generative approaches [15] focus on reconstructing input data or generating new ones. These approaches involve masked image modeling (MiM) [55], where large portions of images are masked for the network to repaint. This approach focuses on local views and pays particular attention to internal information.

CL approaches build on instance discrimination, bringing similar instances closer in the representation space while pushing dissimilar instances apart. CL approaches offer better discriminative power and more robust feature representation learning than other approaches, as they are trained to differentiate between different data instances [21]. They do not require complex reconstruction tasks, making them simple to train and implement. They are also scalable to large data, unlike generative models. A range of methods have been used in literature to contrast data samples.

 doi:10.20944/preprints202411.2059.v1

Negative example-based CL treats views originating from the same sample as positive pairs while treating views from different instances as negative pairs. These methods maximize the proximity between positive pairs and the separation between negative pairs [15]. This method is used in MoCo and SimCLR [21] for 2D image CL.

Self-distillation-based methods like Bootstrap Your Own Latent (BYOL), Simple Siamese Networks (SimSIAM), and DINO eliminate the need for negative pairs. They feed two different views of an input sample to two similar neural network encoders with the same architecture but different weights. These networks are then mapped to each other using a predictor. These methods maximize the similarity between positive pairs while employing diverse strategies to prevent mode collapse [15,16].

Feature decorrelation-based CL methods like Barlow Twins are designed to learn decorrelated features. These approaches generate distorted views of the same image using a distribution of data augmentations and employ strategies to encourage similarities within the embeddings of distorted views while minimizing redundancy between the features. Akin to self-distillation approaches, methods are also used to prevent collapse [15].

Chen et al. [21] introduced SimCLR, a straightforward negative example-based contrastive learning framework for learning feature representations. SimCLR simplifies contrastive learning using a straightforward approach that reduces the need for high memory. It achieves this by contrasting transformed views of images in the same batch and allows for flexible batch sizes [1]. SimCLR is architecturally simple. It utilizes well-known augmentations, simple projection heads, and encoders, making it scalable and adaptable to other architecture and data modalities. It is also model agnostic, allowing for easy application to sensor data by adjusting the encoder to the specific sensor data [12]. Beyond its simplicity and flexibility, SimCLR achieves state-of-the-art performance on benchmark tasks. We use SimCLR as a representation learner in our proposed model owing to its simplicity, robustness, and scalability to different data modalities.

2.3. Section Summary

This section has explored various unobtrusive human motion analysis approaches, including mathematical and rule-based approaches, classical machine learning, deep learning, and hybrid approaches. Although each method has contributed unique strengths and driven advancements toward human motion analysis, they all share a common challenge: the need for large volumes of labeled data, which is costly and requires skilled human annotation with sufficient domain expertise. This data-level challenge is pertinent in human motion analysis, where multiple sensors can easily obtain large data streams. However, assigning activity labels individually to these large streams of sensor data is costly and labor-intensive, posing significant barriers to the efficiency and scalability of existing solutions. We address this gap in this study by harnessing the ability of self-supervised learning to learn supervisory signals directly from unlabeled sensor data. We introduce a hybrid approach combining data preprocessing with SimCLR, a straightforward contrastive learning framework to learn robust representations from video streams. Our learned representations can be used to fine-tune models for downstream tasks. By addressing this data-level gap, we alleviate the financial and logistical burden associated with data labeling and enhance the scalability and robustness of unobtrusive human motion recognition.

3. Method Development

In this section, we outline the methodological details of CLUMM. As illustrated in Figure 1, the proposed CLUMM framework first extracts human pose landmarks from image frames captured in human motion videos (Section 3.1.1). These extracted landmarks are then used to construct a dataset for self-supervised feature representation learning (Section 3.1.2). To evaluate the robustness of CLUMM against outliers and its applicability to anomaly or outlier detection, we deliberately introduce outliers into the training dataset (Section 3.2).

3.1. CLUMM Training

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 November 2024 doi:10.20944/preprints202411.2059.v1

We begin by extracting image frames from human motion videos to enable human motion recognition on raw camera data without manual data labeling. We then use MPP to identify the spatial locations of 33 body joints, as shown in Figure 2. Of the 33 landmarks detected, 10 landmarks of interest are selected based on careful evaluation by ergonomics experts and evidence from existing work to construct a dataset for motion recognition (Section 3.1.1). We utilize a modified version of SimCLR, a contrastive SSL framework for image representation learning, to learn robust representations from the constructed dataset (Section 3.1.2). SimCLR identifies intrinsic motion types in the dataset by grouping similar representations in the feature space and pushing dissimilar ones apart. Finally, we train a simple multiclass logistic regression model on top of the learned representations in a few-shot learning setting, using transfer learning on a small subset of labeled data to validate the generalizability of the representations on downstream motion recognition tasks (Section 3.1.3)

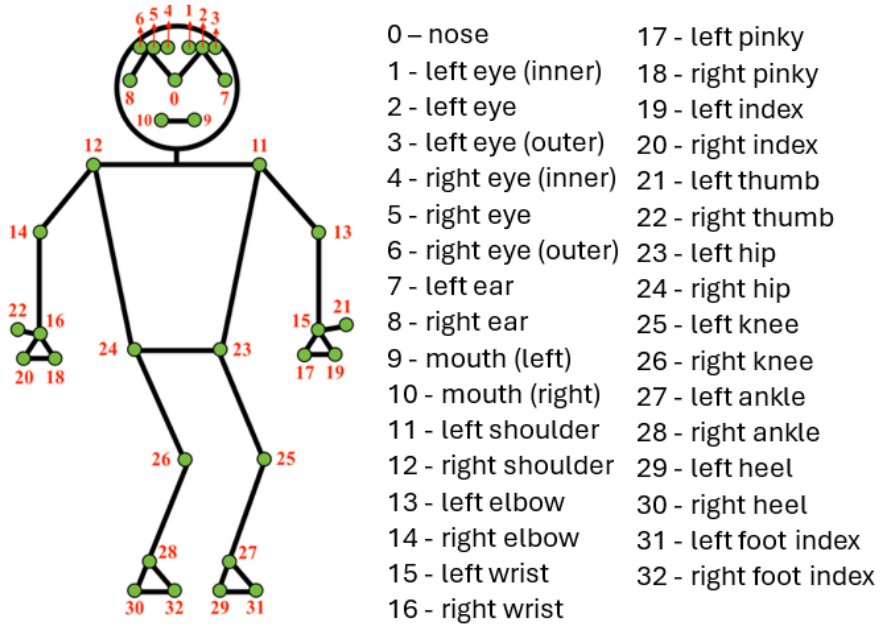

Figure 2. Human Body Pose Landmarks, adapted from Mediapipe [20].

3.1.1. Pose Landmark Extraction

We employ MPP [19] to estimate poses and track landmark locations from image frames obtained from motion videos. MPP is an open-source framework created by Google to estimate high-fidelity 2D and 3D coordinates of body joints. MPP uses BlazePose [20], a lightweight pose estimation network, to detect and track 33 3D body landmarks from videos or images, as depicted in Figure 2. These landmark positions approximate the location of each identified body part in either image or world coordinates. Research in [42] has shown that using joint coordinates as inputs for DL training leads to better results than using raw RGB images. To avoid learning spurious correlations [56] and optimize the performance of our learned representations, we aim to reduce the influence of environmental factors like lighting, contrast, and irrelevant features that might affect the deep learning network and conceal the features of interest. We achieve this by using MPP to extract pose landmarks pertinent to the specific activities of interest, ultimately revealing the position of various joints in the human body. BlazePose can also handle occlusions, thus making our approach robust to occlusions in front of the detected human. We extract 10 landmarks from the upper and lower limbs (Table 1) relevant to the motion tasks we study in this paper, denoted as an index set I. These landmarks were chosen based on evidence from prior studies that identified and used specific landmarks relevant to the task they investigated [5,46]. This approach helps to focus on the human

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 November 2024 doi:10.20944/preprints202411.2059.v1

movement and avoid learning background information. We obtain pose landmarks in normalized image coordinates to make our extraction agnostic to body shape or type.

We extract the normalized X, Y, and Z coordinates of each image frame's ten landmarks of interest and quantize them into a feature vector of 30 elements. Missing coordinates are replaced by 0 to indicate their absence. We finally construct a dataset, a matrix of each frame's feature vectors (see Figure 2).

Table 1. Pose Landmarks of Interest, i.e., elements in I.

Pose number	Representation
11	Left Shoulder
12	Right Shoulder
13	Left Elbow
14	Right Elbow
15	Left Wrist
16	Right Wrist
23	Left Hip
24	Right Hip
25	Left Knee
26	Right Knee

Assuming $L_i = (x_i, y_i, z_i)$ are the cartesian coordinates of the ith landmark. For each image frame $\boldsymbol{D}_j, j = 1, 2, \dots, m$, the feature vector $\boldsymbol{V}_j$ extracted is $\boldsymbol{V}_j = [\boldsymbol{L}_{j_1}, \boldsymbol{L}_{j_2}, \dots, \boldsymbol{L}_{j_{10}}] = [(x_{j_1}, y_{j_1}, z_{j_1}), \dots, (x_{j_{10}}, y_{j_{10}}, z_{j_{10}})]$. The entire dataset is shown in Eq. (1).

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{V}_1 \\ \boldsymbol{V}_2 \\ \vdots \\ \boldsymbol{V}_m \end{bmatrix} = \begin{bmatrix} (x_{1_1}, y_{1_1}, z_{1_1}) & \cdots & (x_{1_{10}}, y_{1_{10}}, z_{1_{10}}) \\ (x_{2_1}, y_{2_1}, z_{2_1}) & \cdots & (x_{2_{10}}, y_{2_{10}}, z_{2_{10}}) \\ \vdots & \ddots & \vdots \\ (x_{m_1}, y_{m_1}, z_{m_1}) & \cdots & (x_{m_{10}}, y_{m_{10}}, z_{m_{10}}) \end{bmatrix} \quad (1)$$

Where m is the number of images.

3.1.2. Contrastive Self-Supervised Representation Learning

We use the SIMCLR contrastive learning framework [21] to learn robust representations in the pose data constructed in Section 3.1.1. SimCLR is a straightforward contrastive SSL framework that maximizes and minimizes the agreement between positive and negative pairs [21]. Positive pairs are generated by applying random augmentations on the same sample. SimCLR is made up of four key components:

- **Random augmentation module**: This module applies random data augmentations on input samples $\boldsymbol{V}_i, i = 1, 2, \dots, m$. It performs two transformations on a single sample, resulting in two correlated views $\widetilde{\boldsymbol{V}}_{i_1}$ and $\widetilde{\boldsymbol{V}}_{i_2}$ treated as a positive pair.
- **Encoder module** $f(\cdot)$: This module uses a neural network to extract latent space encodings of the augmented samples $\widetilde{\boldsymbol{V}}_{i_1}$ and $\widetilde{\boldsymbol{V}}_{i_2}$. The encoder is model agnostic, allowing various network designs to be used. The encoder produces output $\boldsymbol{h}_{i_1} = f(\widetilde{\boldsymbol{V}}_{i_1})$ and $\boldsymbol{h}_{i_2} = f(\widetilde{\boldsymbol{V}}_{i_2})$, where f is the encoder network.
- **Projector head** $g(\cdot)$: This small neural network maps the encoded representations into a space where a contrastive loss is applied to maximize the agreement between the views [16]. A multi-layer perceptron is used, which produces output $\boldsymbol{\psi}_{i_1} = g(\boldsymbol{h}_{i_1})$ and $\boldsymbol{\psi}_{i_2} = g(\boldsymbol{h}_{i_2})$, where g represents the projector head and $\boldsymbol{h}_i$ represents the output of the encoder module.
- **Contrastive loss function:** This learning objective maximizes the agreement between positive pairs.

SimCLR was initially designed for images. Therefore, we make the following modifications to accommodate our transformed pose landmark data.

 doi:10.20944/preprints202411.2059.v1

A. Data Augmentation

We apply two random transformations on our input data to compose the augmentations for our SimCLR training.

- **Random Jitter** $t_1(\cdot)$: We apply a random jitter on the input samples using a noise signal drawn from a normal distribution with a mean of zero and a standard deviation of 0.5, i.e., $\varepsilon_i \sim N(0, 0.5^2)$. Thus, for each input sample $\boldsymbol{V}_i$, we obtain $\widetilde{\boldsymbol{V}}_{i_1} = t_1(\boldsymbol{V}_i) = \boldsymbol{V}_i + \varepsilon_i, i = 1, 2, \dots, m$.
- **Random scaling** $t_2(\cdot)$: We scale samples with a random factor drawn from a normal distribution with a mean zero and a standard deviation of 0.2, i.e., $\eta_i \sim N(0, 0.2^2)$. Therefore, for each input sample $\boldsymbol{V}_i$, we obtain $\widetilde{\boldsymbol{V}}_{i_2} = t_2(\boldsymbol{V}_i) = \boldsymbol{V}_i \cdot \eta_i, i = 1, 2, \dots, m$.

These transformations, shown in the literature to improve the feature representation performance of contrastive learning models [57–59], are composed to form the augmentation module of our modified SimCLR.

B. Encoder Module

We use a pre-trained ResNet model as our encoder to obtain latent space representations of our input samples. We modify the first convolutional layer of the ResNet to accommodate our data, which has a single channel instead of the expected three channels for RGB Images. The ResNet model [50] was introduced by He et al. to address the vanishing gradient problem associated with deep networks by introducing the residual or skip connection, which directly adds the input $\boldsymbol{\xi}$ to the outputs $F(\boldsymbol{\xi}, \boldsymbol{W})$ of a network. Thus, the residual block is represented as $\boldsymbol{\phi} = F(\boldsymbol{\xi}, \boldsymbol{W}) + \boldsymbol{\xi}$, where $\boldsymbol{W}$ is the weight matrix of a layer. We leverage this in our encoder to learn robust representations without degradations.

C. Projector Head

We use a simple multilayer perceptron (MLP) to transform the learned representations from the encoder module into a space where we apply a contrastive loss to maximize the agreement between positive pairs. Our projection head is a three-layer MLP consisting of a linear layer followed by a ReLU activation function and a final linear layer.

D. Contrastive Loss Function

CLUMM uses the Normalized Temperature-scaled cross entropy (NT-Xent) loss [60] from SimCLR. Given two different augmentations $\widetilde{\boldsymbol{V}}_{i_1}$ and $\widetilde{\boldsymbol{V}}_{i_2}$ of an input sample, NT-Xent aims to bring these views closer together in the feature space while pushing views from different samples apart. Assuming $\boldsymbol{\psi}_{i_1}$ and $\boldsymbol{\psi}_{i_2}$ are the embeddings of $\widetilde{\boldsymbol{V}}_{i_1}$ and $\widetilde{\boldsymbol{V}}_{i_2}$ respectfully, after passing through the projector head, NT-Xent is defined as:

$$\mathcal{L}_{i_1,i_2} = -\log \frac{e^{\frac{\mathrm{CoSim}(\boldsymbol{\psi}_{i_1}, \boldsymbol{\psi}_{i_2})}{\tau}}}{\sum_{k=1}^{2b} \mathbf{1}_{[k \neq i]} e^{\frac{\mathrm{CoSim}(\boldsymbol{\psi}_{i_1}, \boldsymbol{\psi}_{i_2})}{\tau}}} \tag{2}$$

Where:

CoSim is the cosine similarity between the two embeddings.

b is the batch size.

τ is the temperature parameter that controls the sharpness of the distribution.

$\mathbf{1}$ is the indicator function that evaluates to 1 when $k \neq i$.

Algorithm 1 summarizes the representation learning process.

Algorithm1: CLUMM feature representation learning

Input: Human Motion Videos $\boldsymbol{\mathcal{D}_1}, \boldsymbol{\mathcal{D}_2}, \dots, \boldsymbol{\mathcal{D}_n}$, landmark extractor $\boldsymbol{M}$ (MediaPipe in this context), landmark indices I, constant τ, batch size b, structure of f, g, and augmentation functions t_1, t_2

#step 1: pose estimation and dataset construction

for each video $\boldsymbol{\mathcal{D}_j}, j = 1, 2, \dots, m$:

Extract image frames from video $\boldsymbol{\mathcal{D}_j} = \{\boldsymbol{D_{j1}}, \boldsymbol{D_{j2}}, \dots, \boldsymbol{D_{jm}}\}$

 doi:10.20944/preprints202411.2059.v1

for each frame $\boldsymbol{D}_{ji} \in \boldsymbol{v}_j$
Extract landmarks indexed by I
$\boldsymbol{L}_{ji} = \boldsymbol{M}(\boldsymbol{D}_{ji}) = \{\boldsymbol{L}_{ji_1}, \boldsymbol{L}_{ji_2}, \ldots, \boldsymbol{L}_{ji_{10}}\}$
Concatenate the extracted landmarks into a feature vector $\boldsymbol{V}_i$
end
Construct a feature matrix from feature vectors $\boldsymbol{V}_{ji}, i = 1, 2, \ldots, m$

$$\boldsymbol{v}_j = [\boldsymbol{V}_{j1}, \boldsymbol{V}_{j2}, \ldots, \boldsymbol{V}_{jm}]^T$$

end
Step 2: SimCLR training
for sampled minibatch in $\{\boldsymbol{V}_k\}_{k=1}^{b} \in \boldsymbol{v}_j, j = 1, 2, \ldots, n$
for each sample $k \in \{1, \ldots, b\}$
Apply the two augmentation functions to each sample

$$\widetilde{\boldsymbol{V}}_{k_1} = t_1(\boldsymbol{V}_k), \widetilde{\boldsymbol{V}}_{k_2} = t_2(\boldsymbol{V}_k)$$

Pass augmented samples through encoder f
$\boldsymbol{h}_{k_1} = f(\widetilde{\boldsymbol{V}}_{k_1})$, $\boldsymbol{h}_{k_2} = f(\widetilde{\boldsymbol{V}}_{k_2})$
Pass encoded representations through projection head g
$\boldsymbol{\psi}_{k_1} = g(\boldsymbol{h}_{k_1}), \boldsymbol{\psi}_{k_2} = g(\boldsymbol{h}_{k_2})$
end
for each pair $k_1, k_2 \in \{1, \ldots 2b\}$
Compute pairwise similarity

$$S_{k_1,k_2} = \textbf{CoSim}\,(\boldsymbol{\psi}_{k_1}, \boldsymbol{\psi}_{k_2})$$

Compute NT-Xent loss in Eq (2)
end
Update networks f and g to minimize the loss
end
return encoder f and throw away g

E. Finetuning with SoftMax Logistic Regression

We assess the generalization performance of our learned representations by fine-tuning a multinomial logistic regression model [61] in a transfer learning setting. Multinomial logistic regression extends the standard logistic regression [62] for classification problems with more than two classes. Given a dataset $\{(\boldsymbol{V}_i, \mathcal{y}_i)\}_{i=1}^{n}$ where $\boldsymbol{V}_i \in \mathbb{R}^d$ is the ith input feature vector and $\mathcal{y}_i \in \{1,2 \ldots, K\}$ is the corresponding class label with K representing the number of classes, multinomial logistic regression computes the probability of a sample $\boldsymbol{V}_i$ belonging to class k using the SoftMax [63] represented as Eq. (3)

$$P(\mathcal{y}_i = k|\boldsymbol{V}_i) = \frac{e^{(\phi_k(\boldsymbol{V}_i))}}{\sum_{j=1}^{K} e^{(\phi_j(\boldsymbol{V}_i))}} \tag{3}$$

where ϕ_k is the raw calculated score for each class given by

$$\phi_k(\boldsymbol{V}_i) = \boldsymbol{w}_k^T \boldsymbol{V}_i + \mathcal{b}_k \tag{4}$$

where $\mathcal{b}$ is the bias term, $\boldsymbol{w}_k$ is a weight vector associated with a class k.

Given a single training instance $(\boldsymbol{V}_i, \mathcal{y}_i)$, the loss function is represented as

$$\ell(\boldsymbol{W}, \mathcal{b}) = -\log P((\mathcal{y}_i \mid \boldsymbol{V}_i) \tag{5}$$

with $\boldsymbol{W} = [\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, \ldots, \boldsymbol{w}_K^T]$, which generalizes across all training samples as

$$J(\boldsymbol{W}, \mathcal{b}) = -\frac{1}{n}\sum_{i=1}^{n} \log P((\mathcal{y}_i \mid \boldsymbol{V}_i) \tag{6}$$

The parameters $\boldsymbol{W}$ and $\mathcal{b}$ are learned by minimizing the loss function using an optimization technique such as gradient descent or its variants [64].

 doi:10.20944/preprints202411.2059.v1

We manually categorize a subset of our dataset into three activity classes, *idle*, *lift*, and *bend*, and randomly split the dataset into a training and a testing set (distinct from those used for pre-text training). We fix the encoder weights of CLUMM for transfer learning and train only the linear layer connected to the encoder. We conduct 5-fold cross-validation to prevent selection bias and ensure a dependable evaluation of the performance of our logistic model.

3.2. Impact of Outliers on CLUMM Performance

Using ML/DL methods for human motion recognition requires training data. For the proposed CLUMM, training data containing several fixed motion types (classes) are used to train SimCLR to obtain the feature embeddings. Introducing unexpected motion types, in addition to the fixed ones, would result in "outliers" in the training data, potentially impacting SimCLR's feature extraction performance and, thus, the classification accuracy. We investigate the effects of these outliers on our proposed model by first clustering the features of interest and calculating the distance from outliers to inliers. We leverage the K-Means clustering algorithm to cluster the extracted coordinates (features). The K-Means algorithm minimizes the within-cluster sum of squared distances (WCSS), defined as:

$$WCSS = \sum_{i=1}^{c} \sum_{\boldsymbol{V} \in C_i} |\boldsymbol{V} - \boldsymbol{\mu}_i|^2 \tag{7}$$

Where:

c is the number of clusters,

C_i is the set of points belonging to cluster i,

$\boldsymbol{\mu}_i$ is the centroid of cluster i,

$\boldsymbol{V} = [V_1, V_2, \dots, V_{30}]$ is a data point (feature vector of length 30 here),

$|\boldsymbol{V} - \boldsymbol{\mu}_i|^2$ is the squared Euclidean distance between $\boldsymbol{V}$ and $\boldsymbol{\mu}_i$.

In our case, the coordinates are clustered into $c = 3$ clusters and the centroids are updated to minimize WCSS. After performing K-Means clustering, the Euclidean distance between each point $\boldsymbol{V}$ and the closest centroid $\boldsymbol{\mu}_i$ is computed using:

$$d(\boldsymbol{V}, \boldsymbol{\mu}_i) = \sqrt{(V_1 - \mu_{i1})^2 + (V_2 - \mu_{i2})^2 + \cdots + (V_{30} - \mu_{i30})^2} \tag{8}$$

where $V_1, V_2, \dots, V_{30}$ are the coordinates of the point $\boldsymbol{V}$ (where we use V to represent the $\boldsymbol{V} = [(x_1, y_1, z_1), \dots, (x_{10}, y_{10}, z_{10})]$ defined in Eq. (1)), $\mu_{i1}, \mu_{i2}, \dots, \mu_{i30}$ are the coordinates of the centroid $\boldsymbol{\mu}_i$.

Outliers are data points whose distance from the nearest centroid exceeds a threshold. The threshold is calculated as:

$$t = \mu_d + 2\sigma_d \tag{9}$$

where μ_d is the mean of the distances from points to their nearest centroid (see Eq. 10), and σ_d is the standard deviation of those distances.

Points with $d(\boldsymbol{V}, \boldsymbol{\mu}_i) > t$ are considered outliers. For each cluster, the following metrics are computed:

1. The average distance from all points in a cluster i to their respective centroid

$$\mu_d^{(i)} = \frac{1}{|C_i|} \sum_{\boldsymbol{V}_j \in C_i} d(\boldsymbol{V}_j, \boldsymbol{\mu}_i) \tag{10}$$

Where:

$\boldsymbol{\mu}_i$ is the centroid of the cluster, which is the mean of all points in the cluster in terms of their coordinates.

$\mu_d^{(i)}$ is the average distance between each point in the cluster and the centroid $\boldsymbol{\mu}_i$. It quantifies how tightly clustered the points are around the centroid.

$d(\cdot,\cdot)$ is the distance between the data point and the centroid, typically computed using the Euclidean distance formula.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 November 2024 doi:10.20944/preprints202411.2059.v1

$|\cdot|$ is the cardinality of a set measuring the size (or the number of data points) of the set [65].

2. The maximum distance of any point in a cluster i to the centroid:

$$M_d^{(i)} = \max_{\boldsymbol{V}_j \in C_i} \left(d(\boldsymbol{V}_j, \boldsymbol{\mu}_i) \right) \tag{11}$$

M_d indicates how spread out or far away the most distant point is from the cluster's center.

Since our framework relies on contrastive learning, where the performance of the loss is dependent on the selection of positive and negative pairs [66], the presence of outliers in the training data is likely to distort the process by presenting false positive pairs or hard negatives, causing the model to emphasize irrelevant features which will affect the quality of the learned representations and the generalization of the model to downstream tasks. Given the contrastive loss in Eq. (2), the presence of outliers introduces noisy or irrelevant negative pairs, increasing the denominator and making it harder to distinguish true positive pairs. Additionally, false positives introduced by outliers distort the similarity metric $\text{CoSim}(\boldsymbol{\psi}_{i_1}, \boldsymbol{\psi}_{i_2})$ by lowering the similarity and subsequently increasing the loss. Assuming a binary indicator $\omega_i \in 0,1$ where $\omega_i = 1$ indicates the presence of an outlier and $\omega_i = 0$ as a regular sample, we can rewrite the contrastive loss Eq. (2) with outliers as:

$$\mathcal{L}_{i_1,i_2} = -\log \frac{e^{\frac{\text{CoSim}(\boldsymbol{\psi}_{i_1}, \boldsymbol{\psi}_{i_2})}{\tau}}}{\sum_{k=1}^{2b} 1_{[k \neq i]} e^{\frac{\text{CoSim}(\boldsymbol{\psi}_{i_1}, \boldsymbol{\psi}_{i_2})}{\tau}} + \lambda \delta \omega_i} \tag{12}$$

Where:

λ is the mean distance of an outlier to the cluster centroid.

δ is a weight controlling the impact of the outlier on the denominator.

This modification of the contrastive loss shows that outliers $\omega_i = 1$ increase the denominator and consequently the overall loss, affecting the ability of the network to learn good representations.

3.3. Summary of CLUMM

As shown in Figure 3, our proposed CLUMM framework first extracts frames from videos. Then, it performs a pose extraction step (Section 3.1.1) to produce quantized input samples. These are sent into a contrastive learning step (Section 3.3) to learn feature representations directly from the unlabeled input. To demonstrate the robustness of the proposed framework, we finetune a simple multinomial logistic regression model on the learned representation using a small set of manually labeled data. The result of this experiment is shown in Section 4.

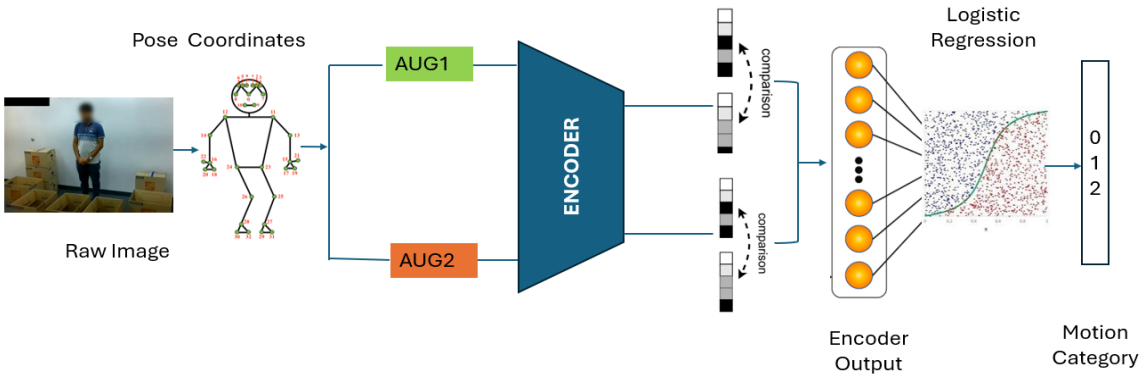

Figure 3. CLUMM Framework.

3.4. Note to Practitioners

In this paper, we approach the human motion recognition problem from a self-supervised approach using skeletal coordinates of various motion types across various application domains. Findings from our experiments show that our approach can effectively identify intrinsic motion types in the data, which is a benefit of self-supervised learning. Our framework can detect different motion

 doi:10.20944/preprints202411.2059.v1

types on the fly as they are introduced to the training data. This ability to capture intrinsic motions from the data makes our approach applicable to motion recognition, outlier analysis, and anomaly detection by fine-tuning on labeled data with transfer learning.

4.0. Case Study

To evaluate CLUMM's effectiveness in motion recognition, we conducted experiments on a custom dataset simulating various tasks in a workplace setting. The dataset comprises videos captured in a laboratory environment using an optical camera. The experimental design involved human subjects performing tasks with boxes of varying sizes. These tasks included lifting a box, inserting a box into another box, and placing a box on a surface. The actions were performed either randomly or systematically using a standardized numbering system. Video recordings of participants (age 27 ± 1.41 years, arm length 0.70 ± 0.014 meters, height 1.74 ± 0.014 meters) were recorded at 30 frames per second with standardized resolution and dimensions. To enhance the dataset and eliminate blind spots, videos were recorded from three angles (left, center, and right), providing multiple perspectives. The recorded videos were time-synchronized and denoised to ensure frame consistency across the dataset.

The experimental procedure was approved by the Institutional Review Board of Arizona State University. Each participant reviewed and signed an IRB-approved informed consent prior to participation.

4.1. Data Preparation

We extracted image frames from three different videos of a human subject performing multiple tasks. These frames contained distinct actions repeated across each task. The unlabeled image frames were shuffled and preprocessed for CLUMM training, followed by the pose extraction pipeline described in Section 3.1.1. This process eliminated background effects, such as the presence of boxes and variations in lighting, as well as differences in the size and height of the human subject. These steps ensured that the representations learned from the data were generic, invariant to scale and lighting conditions, and generalizable across diverse operational environments. Notably, no labels were assigned to the dataset at this stage.

4.2. Human Motion Recognition with CLUMM

We trained the modified SimCLR (Section 3.1.2) to learn representations directly from the unlabeled pose dataset. For the encoder, we experimented with two pre-trained models, ResNet18 and ResNet50, and employed a three-layer MLP with an output dimension of 128 as the projector head (Section 3). The AdamW optimizer was used with an initial learning rate of 0.005 and a cosine learning rate decay schedule for pretraining over 500 epochs. Due to limited data availability, we used a small batch size of 64. This training step was designed to learn robust representations that could serve as initial weights for downstream motion classification. Leveraging SimCLR, we grouped features representing the same motion types, while features of different motion types were pushed apart. This process effectively captured the intrinsic variability of motion types in the dataset.

To explicitly categorize the motion types, we manually labeled a subset of the dataset into three activity classes: *idle*, *lift*, and *bend*. The labeled dataset was then randomly split into a training set of 1,500 frames and a test set of 916 frames (distinct from the frames used for pre-text training). Since the learned representations from the modified SimCLR are feature embeddings rather than class labels, we trained a simple multinomial logistic regression model on the labeled dataset to identify motion categories and assess the efficiency of the learned representations. This was done in a transfer learning setting where the encoder weights of the SimCLR component were frozen, and only the logistic regression layer was trained. Thus, SimCLR functioned as a feature extractor for the logistic regression model. The logistic regression model was trained with 5-fold cross-validation for 100 epochs using the Adam optimizer with a learning rate of 0.01 and a multistep learning rate decay. We provide a comparison of the fine-tuning performance with a baseline ResNet model in Table 2.

 doi:10.20944/preprints202411.2059.v1

Table 2. Comparison of fine-tuned CLUMM with baseline ResNets.

Network	Accuracy	Precision	Recall	F1 Score
ResNet18 Baseline	79.6%	0.801	0.797	0.796
ResNet50 Baseline	77.3%	0.782	0.773	0.771
CLUMM (ResNet50 Backbone)	**83.5 %**	**0.833**	**0.835**	**0.831**
CLUMM (ResNet18 Backbone)	**90.0 %**	**0.899**	**0.90**	**0.899**

4.3. Baseline Evaluation

We compare the motion recognition performance of CLUMM using ResNet18 and ResNet50 backbones to the performance of baseline ResNet18 and ResNet50 transfer learners on the same dataset; the initial layers of the ResNets are frozen, with only the last fully connected layer being trained. We keep everything else the same as the fine-tuning process with CLUMM. Table 3 compares the performance of our fine-tuned CLUMM and baseline ResNet models (the best performances have been bolded). CLUMM outperformed the baseline models with either ResNet18 or ResNet50 backbones, indicating the enhanced feature extraction by CLUMM due to its self-supervised learning capability.

Table 3. Results from outlier analysis using a ResNet18 backbone.

Number of Outlier images	Percentage of outlier landmarks	Mean outlier distance	Max outlier distance	Accuracy	Precision	Recall	F1-Score
None	-			**90.00 %**	**0.899**	**0.90**	**0.899**
100	3.5 %	0.30	0.80	86.76 %	0.867	0.867	0.865
200	3.1 %	0.30	0.81	84.48%	0.845	0.844	0.843
500	3 %	0.31	0.83	84.64	0.845	0.846	0.846

4.5. Results from Outlier Analysis

We manually introduced outliers into the training data to investigate the effects of outliers on our proposed model. We introduced different motion types from the three motion categories used in this study. We examined 100, 200, and 500 outliers and reported their impact on the performance of CLUMM with ResNet18 backbone (See Table 3). Since outliers may have specific landmark positions similar to the regular motion, we performed outlier analysis for each landmark using the methodology described in Section 3.3. Keeping all our parameters and hyperparameters constant, we performed CLUMM training on the new training datasets with outliers and fine-tuned testing data. As hypothesized in Section 3.3, we observed a decrease in the generalization capacity of CLUMM on the downstream task with the introduction of outliers in the training data. However, we observed that increasing the number of outliers from 200 to 500 did not affect performance by a significant margin. This proves the robustness of the framework to outliers. We show the results of our analysis in Table 3. Figure 4 shows a 2D representation of the embedding space, where the axes are the first and second principal components, demonstrating the grouping of the various motion types intrinsic to the data. The majority of the data instances have been well separated from other motion types.

 doi:10.20944/preprints202411.2059.v1

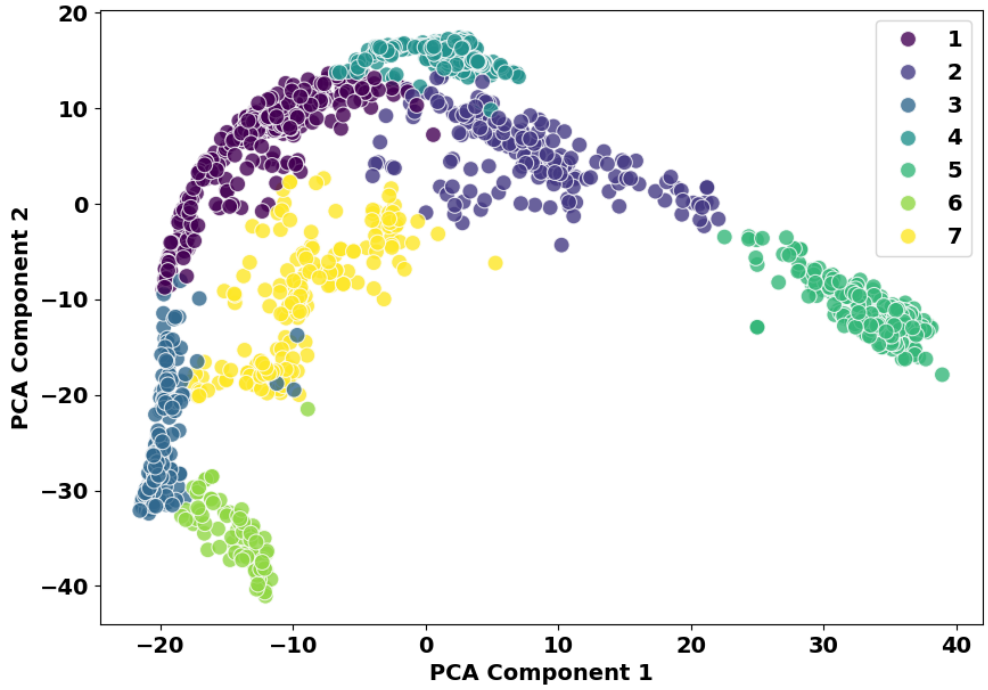

Figure 4. 2D Visualization of embedding space of learned representations showing the intrinsic motion types in the data.

5. Conclusion and Future Work

This paper presents a comprehensive framework for human motion analysis, bridging gaps in the implementation of efficient and unobtrusive human motion recognition to promote better human well-being. Our proposed solution leverages MediaPipe and a contrastive learning framework for feature extraction, addressing data-level challenges by streamlining the data labeling and feature extraction processes. Results from fine-tuning domain-specific data demonstrate the effectiveness of our approach in achieving accurate and scalable motion analysis, which can be adapted to other motion-related tasks in complex environments. Our fine-tuned model outperforms a baseline supervised model, showcasing the potential of self-supervised learning to significantly reduce manual labeling efforts.

Although this study achieved impressive results, there are limitations that we plan to address in future work. Firstly, the dataset was collected in a controlled laboratory environment, which may not fully reflect the dynamics of real-world operations. In our future research, we aim to collect data in natural environments to validate the scalability of our proposed methodology.

Additionally, we plan to extend this work to include multi-camera human sensing and adaptive multi-sensory data fusion. Our future efforts will specifically focus on integrating multiple camera perspectives to capture more comprehensive data and applying fusion technologies to learn simultaneously from various angles. We also intend to investigate how incorporating new data streams can enhance the performance of our representation learning module and enable continuous learning of new motion types.

Funding: This work is supported by the Arizona State University startup grants. This work was also supported by the National Institute for Occupational Safety and Health (NIOSH) under Grant T42OH008672

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Institutional Review Board Statement: This research adhered to the American Psychological Association Code of Ethics and received approval from the Institutional Review Board at Arizona State University (STUDY00016442).

Data Availability Statement: The data are not publicly available, due to privacy.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. C. I. Tang, I. Perez-Pozuelo, D. Spathis, and C. Mascolo, "Exploring Contrastive Learning in Human Activity Recognition for Healthcare," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.11542
2. Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," Dec. 15, 2019, *Elsevier Ltd*. doi: 10.1016/j.eswa.2019.04.057.
3. D. V. Thiel and A. K. Sarkar, "Swing Profiles in Sport: An Accelerometer Analysis," *Procedia Eng*, vol. 72, pp. 624–629, 2014, doi: 10.1016/j.proeng.2014.06.106.
4. H. Liu and L. Wang, "Human motion prediction for human-robot collaboration," *J Manuf Syst*, vol. 44, pp. 287–294, Jul. 2017, doi: 10.1016/j.jmsy.2017.04.009.
5. H. Iyer, N. Macwan, S. Guo, and H. Jeong, "Computer-Vision-Enabled Worker Video Analysis for Motion Amount Quantification," May 2024, [Online]. Available: http://arxiv.org/abs/2405.13999
6. J. M. Fernandes, J. S. Silva, A. Rodrigues, and F. Boavida, "A Survey of Approaches to Unobtrusive Sensing of Humans," Mar. 01, 2023, *Association for Computing Machinery*. doi: 10.1145/3491208.
7. C. Pham, N. N. Diep, and T. M. Phuonh, "e-Shoes: Smart Shoes for Unobtrusive Human Activity Recognition," IEEE, 2017.
8. A. Rezaei, M. C. Stevens, A. Argha, A. Mascheroni, A. Puiatti, and N. H. Lovell, "An Unobtrusive Human Activity Recognition System Using Low Resolution Thermal Sensors, Machine and Deep Learning," *IEEE Trans Biomed Eng*, vol. 70, no. 1, pp. 115–124, Jan. 2023, doi: 10.1109/TBME.2022.3186313.
9. C. Yu, Z. Xu, K. Yan, Y. R. Chien, S. H. Fang, and H. C. Wu, "Noninvasive Human Activity Recognition Using Millimeter-Wave Radar," *IEEE Syst J*, vol. 16, no. 2, pp. 3036–3047, Jun. 2022, doi: 10.1109/JSYST.2022.3140546.
10. M. Gochoo, T. H. Tan, S. H. Liu, F. R. Jean, F. S. Alnajjar, and S. C. Huang, "Unobtrusive Activity Recognition of Elderly People Living Alone Using Anonymous Binary Sensors and DCNN," *IEEE J Biomed Health Inform*, vol. 23, no. 2, pp. 693–702, Mar. 2019, doi: 10.1109/JBHI.2018.2833618.
11. K. A. Muthukumar, M. Bouazizi, and T. Ohtsuki, "A Novel Hybrid Deep Learning Model for Activity Detection Using Wide-Angle Low-Resolution Infrared Array Sensor," *IEEE Access*, vol. 9, pp. 82563–82576, 2021, doi: 10.1109/ACCESS.2021.3084926.
12. K. Takenaka, K. Kondo, and T. Hasegawa, "Segment-Based Unsupervised Learning Method in Sensor-Based Human Activity Recognition," *Sensors (Basel)*, vol. 23, no. 20, Oct. 2023, doi: 10.3390/s23208449.
13. I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," Nov. 01, 2021, *Springer*. doi: 10.1007/s42979-021-00815-1.
14. S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K. R. Müller, "Unmasking Clever Hans predictors and assessing what machines really learn," *Nat Commun*, vol. 10, no. 1, Dec. 2019, doi: 10.1038/s41467-019-08987-4.
15. J. Gui et al., "A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends," Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.05712
16. R. Balestriero et al., "A Cookbook of Self-Supervised Learning," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.12210
17. H. Haresamudram et al., "Masked reconstruction based self-supervision for human activity recognition," in *Proceedings - International Symposium on Wearable Computers, ISWC*, Association for Computing Machinery, Sep. 2020, pp. 45–49. doi: 10.1145/3410531.3414306.
18. H. Haresamudram, I. Essa, and T. Plötz, "Assessing the State of Self-Supervised Human Activity Recognition Using Wearables," *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 6, no. 3, Sep. 2022, doi: 10.1145/3550299.
19. C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.08172
20. V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose tracking," Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.10204
21. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.05709
22. A. Parnami and M. Lee, "Learning from Few Examples: A Summary of Approaches to Few-Shot Learning," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.04291
23. C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimed Tools Appl*, vol. 76, no. 3, pp. 4405–4425, Feb. 2017, doi: 10.1007/s11042-015-3177-1.
24. M. Al-Amin, R. Qin, W. Tao, and M. C. Leu, "Sensor Data Based Models for Workforce Management in Smart Manufacturing."
25. Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," Dec. 15, 2019, *Elsevier Ltd*. doi: 10.1016/j.eswa.2019.04.057.

26. M. Al-Amin et al., "Action recognition in manufacturing assembly using multimodal sensor fusion," in *Procedia Manufacturing*, Elsevier B.V., 2019, pp. 158–167. doi: 10.1016/j.promfg.2020.01.288.
27. S. Chung, J. Lim, K. J. Noh, G. Kim, and H.-T. Jeong, "Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning.," *Sensors*, 2019, doi: 10.3390/S19071716.
28. L. Yao et al., "Compressive Representation for Device-Free Activity Recognition with Passive RFID Signal Strength," *IEEE Trans Mob Comput*, vol. 17, no. 2, pp. 293–306, Feb. 2018, doi: 10.1109/TMC.2017.2706282.
29. Z. Luo, Y. Zou, V. Tech, J. Hoffman, and L. Fei-Fei, "Label Efficient Learning of Transferable Representations across Domains and Tasks."
30. Y. Karayaneva, S. Baker, B. Tan, and Y. Jing, "Use of Low-Resolution Infrared Pixel Array for Passive Human Motion Movement and Recognition," BCS Learning & Development, 2018. doi: 10.14236/ewic/hci2018.143.
31. A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar," in *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, Association for Computing Machinery, Oct. 2019, pp. 51–56. doi: 10.1145/3349624.3356768.
32. M. Möncks, J. Roche, and V. De Silva, "Adaptive Feature Processing for Robust Human Activity Recognition on a Novel Multi-Modal Dataset."
33. H. Foroughi, B. Shakeri Aski, and H. Pourreza, "Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments."
34. J. Heikenfeld et al., "Wearable sensors: modalities, challenges, and prospects," *Lab Chip*, vol. 18, no. 2, pp. 217–248, 2018, doi: 10.1039/C7LC00914C.
35. N. T. Newaz and E. Hanada, "A Low-Resolution Infrared Array for Unobtrusive Human Activity Recognition That Preserves Privacy," *Sensors*, vol. 24, no. 3, Feb. 2024, doi: 10.3390/s24030926.
36. S. Nehra and J. L. Raheja, "Unobtrusive and Non-Invasive Human Activity Recognition using Kinect Sensor," IEEE, 2020.
37. H. Li, C. Wan, R. C. Shah, P. A. Sample, and S. N. Patel, "IDAct: Towards Unobtrusive Recognition of User Presence and Daily Activities," Institute of Electrical and Electronics Engineers, 2019, p. 245.
38. G. A. Oguntala et al., "SmartWall: Novel RFID-Enabled Ambient Human Activity Recognition Using Machine Learning for Unobtrusive Health Monitoring," *IEEE Access*, vol. 7, pp. 68022–68033, 2019, doi: 10.1109/ACCESS.2019.2917125.
39. I. Alrashdi, M. H. Siddiqi, Y. Alhwaiti, M. Alruwaili, and M. Azad, "Maximum Entropy Markov Model for Human Activity Recognition Using Depth Camera," *IEEE Access*, vol. 9, pp. 160635–160645, 2021, doi: 10.1109/ACCESS.2021.3132559.
40. H. Wu, W. Pan, X. Xiong, and S. Xu, "Human Activity Recognition Based on the Combined SVM&HMM," in *Proceeding of the IEEE International Conference on Information and Automation*, IEEE, 2014, p. 1317.
41. Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," May 27, 2015, *Nature Publishing Group*. doi: 10.1038/nature14539.
42. B. Açış and S. Güney, "Classification of human movements by using Kinect sensor," *Biomed Signal Process Control*, vol. 81, Mar. 2023, doi: 10.1016/j.bspc.2022.104417.
43. R. G. Ramos, J. D. Domingo, E. Zalama, and J. Gómez-García-bermejo, "Daily human activity recognition using non-intrusive sensors," *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165270.
44. P. Choudhary, P. Kumari, N. Goel, and M. Saini, "An Audio-Seismic Fusion Framework for Human Activity Recognition in an Outdoor Environment," *IEEE Sens J*, vol. 22, no. 23, pp. 22817–22827, Dec. 2022, doi: 10.1109/JSEN.2022.3208271.
45. J. Quero, M. Burns, M. Razzaq, C. Nugent, and M. Espinilla, "Detection of Falls from Non-Invasive Thermal Vision Sensors Using Convolutional Neural Networks," MDPI AG, Oct. 2018, p. 1236. doi: 10.3390/proceedings2191236.
46. H. Iyer and H. Jeong, "PE-USGC: Posture Estimation-Based Unsupervised Spatial Gaussian Clustering for Supervised Classification of Near-Duplicate Human Motion," *IEEE Access*, vol. 12, pp. 163093–163108, 2024, doi: 10.1109/ACCESS.2024.3491655.
47. E. S. Rahayu, E. M. Yuniarno, I. K. E. Purnama, and M. H. Purnomo, "Human activity classification using deep learning based on 3D motion feature," *Machine Learning with Applications*, vol. 12, p. 100461, Jun. 2023, doi: 10.1016/j.mlwa.2023.100461.
48. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks." [Online]. Available: https://github.com/liuzhuang13/DenseNet.
49. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions."
50. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385
51. K. Wickstrøm, M. Kampffmeyer, K. Ø. Mikalsen, and R. Jenssen, "Mixing up contrastive learning: Self-supervised representation learning for time series," *Pattern Recognit Lett*, vol. 155, pp. 54–61, Mar. 2022, doi: 10.1016/j.patrec.2022.02.007.

52. S. Liu et al., "Audio Self-supervised Learning: A Survey," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.01205
53. M. C. Schiappa, Y. S. Rawat, and M. Shah, "Self-Supervised Learning for Videos: A Survey," Jun. 2022, doi: 10.1145/3577925.
54. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. in Adaptive Computation and Machine Learning series. MIT Press, 2016. [Online]. Available: https://books.google.com/books?id=omivDQAAQBAJ
55. Z. Xie et al., "SimMIM: a Simple Framework for Masked Image Modeling." [Online]. Available: https://github.com/microsoft/SimMIM
56. W. Ye, G. Zheng, X. Cao, Y. Ma, and A. Zhang, "Spurious Correlations in Machine Learning: A Survey," Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.12715
57. A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A Survey on Contrastive Self-supervised Learning," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2011.00362
58. Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What Makes for Good Views for Contrastive Learning?," May 2020, [Online]. Available: http://arxiv.org/abs/2005.10243
59. K. Shah, D. Spathis, C. I. Tang, and C. Mascolo, "Evaluating Contrastive Learning on Wearable Timeseries for Downstream Clinical Outcomes," Nov. 2021, [Online]. Available: http://arxiv.org/abs/2111.07089
60. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.05709
61. C. Kwak and A. Clayton-Matthews, "Multinomial Logistic Regression," *Nurs Res*, vol. 51, pp. 404–410, 2002, [Online]. Available: https://api.semanticscholar.org/CorpusID:21650170
62. D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.
63. S. Ji and Y. Xie, "Logistic Regression: From Binary to Multi-Class."
64. S. Ruder, "An overview of gradient descent optimization algorithms," Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.04747
65. D. Hong, J. Wang, and R. Gardner, "Chapter 1 - Fundamentals," in *Real Analysis with an Introduction to Wavelets and Applications*, D. Hong, J. Wang, and R. Gardner, Eds., Burlington: Academic Press, 2005, pp. 1–32. doi: https://doi.org/10.1016/B978-012354861-0/50001-4.
66. H. Xuan, A. Stylianou, X. Liu, and R. Pless, "Hard Negative Examples are Hard, but Useful," 2020, pp. 126–142. doi: 10.1007/978-3-030-58568-6_8.