CARE: Counterfactual-based Algorithmic Recourse for Explainable Pose Correction

Bhat Dittakavi
Indian Institute of Technology Hyderabad
ai19resch11001@iith.ac.in

Bharathi Callepalli
Variance.ai
callepalli@gmail.com

Aleti Vardhan
Manipal Institute of Technology
vardhanaleti2001@gmail.com

Sai Vikas Desai Vineeth N Balasubramanian
Indian Institute of Technology Hyderabad
{cs17mtech11011, vineethnb}@iith.ac.in

Abstract

With increasing popularity of home-based fitness regimen post-pandemic, there has been a growing interest in fitness monitoring solutions. Owing to this, human pose monitoring has gained significant commercial importance in the field of computer vision. Most efforts in the past focused on the task of human pose classification for various applications. In this work, we instead focus on a critical aspect of human pose monitoring that naturally follows from basic pose classification i.e., pose analysis and correction. Specifically, we study human pose correction through the lens of algorithmic recourse. Algorithmic recourse involves a model providing explanations on a how a model arrived at a decision, along with possible actions to drive the model to output a favorable decision. To this end, we develop CARE (Counterfactuals based Algorithmic Recourse for Explainable pose correction), a novel framework that uses counterfactual explanations to provide recourse for incorrect human poses, thereby helping a user correct their pose. Experiments on three diverse datasets, including two fitness datasets and one hand gestures dataset, demonstrate the effectiveness and applicability of CARE.

1. Introduction

Human pose estimation is a fundamental problem in computer vision, with a myriad applications including action recognition, sports, healthcare, human-computer interaction, and surveillance. Many efforts to estimate and classify human pose have been proposed in recent years [2, 7, 25, 31]. However, the related task of pose correction, which plays a vital role in human pose monitoring, has received limited attention in literature. Pose correction typically involves the following steps: (1) human pose estimation, followed by (2) assessing the correctness of the estimated pose w.r.t. an expected pose, and finally (3) offering actionable interventions to rectify any detected errors. A sample illustration is shown in Figure 1. Pose correction has applications in various practical domains, including personal fitness, sports training, and rehabilitation. For instance, an AI-powered fitness coach can utilize pose correction techniques to deliver real-time feedback and guidance during fitness routines like yoga and pilates, thereby assisting users in achieving the correct pose. In the context of sports training, a pose correction system can analyze live or practice footage of athletes, effectively identifying subtle mistakes in their posture or movements, and subsequently providing personalized performance assessments. Similarly, a pose correction system can serve as a valuable tool for individuals undergoing physical rehabilitation to track and analyze the progress while performing rehabilitative exercises.

Some recent works have addressed the problem of human pose monitoring. For instance, Katayama et al. [10] developed a privacy-preserving point cloud extraction method to assess a user's posture while sitting on a work desk. Kishore et al. [12] devised a voice-based feedback system to provide instructions to fix incorrect yoga poses. In addition, 3D fitness monitoring datasets such as Fit3D [4], 3D-Yoga [15], and EC3D [34] have been released. While RGB-D and motion capture data contain rich information about human movements, obtaining such data needs specialized equipment whose costs are often restrictive. To allow for much wider usage, our pose correction system relies on 2D image data, thus making it more widely deployable in practice on consumer devices such as smartphones.

Among methods geared specifically towards human pose *correction*, [34] train a Graph Convolutional Network (GCN) to provide corrective feedback, but provide results on only 3 exercise categories. [10, 12] develop methods to

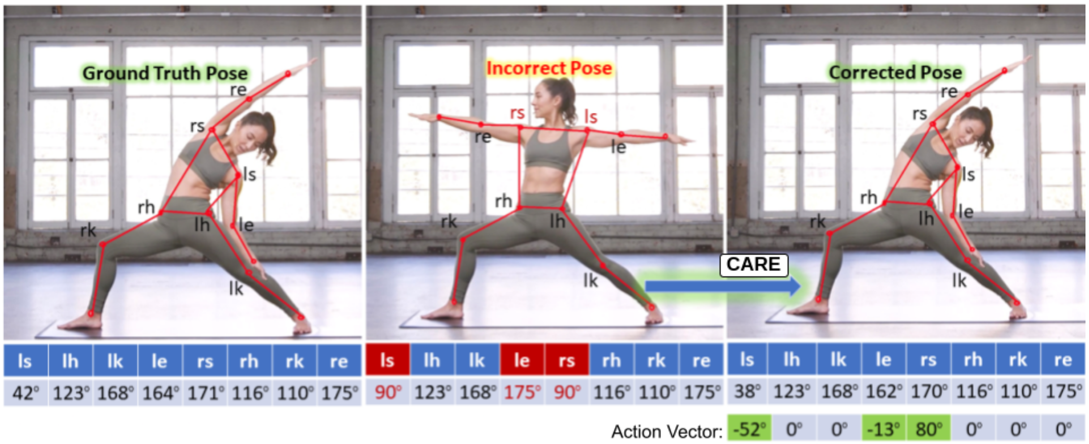

Figure 1. **Illustration of Human Pose Correction:** Figure shows an expected yoga pose (*left*); an incorrectly performed user pose, which is incorrect at the left shoulder (**ls**), right shoulder (**rs**) and left elbow (**le**) (*middle*); and the final correctly performed user pose (*right*). The values below these poses denote joint angles in degrees. CARE identifies the incorrect joint angles (**ls** = 90°, **le** = 175°, **rs** = 90° in red) and provides a sparse action vector (actions -52°, -13°, 81 ° in green) to obtain the corrected pose. Note that although the joint **re** is not in its right position in the middle figure, fixing **rs** automatically fixes **re**, which motivates the need for sparsity in the proposed action vector.

provide pose feedback, but no quantitative analysis of the correction methods is seen. [3, 4] propose statistical methods to correct exercises; [4] however relies on expensive motion capture technology, while we focus on developing a pose correction system that relies on easy-to-obtain 2D image data and evaluate our system on a large number of pose categories. The effort closest to our work is [3] in its pre-processing steps; however, it aims to localize the joint angle error, while our work herein quantifies the joint angle corrections through an action vector and provides actionable inputs to the user.

In this work, we view the human pose correction problem from the lens of algorithmic recourse [9]. We propose a methodology for Counterfactual-based Algorithmic Recourse towards Explainable pose correction (CARE) where we accomplish pose correction by making minimal changes to the incorrect pose to achieve the nearest counterfactual pose (see Figure 1). Algorithmic recourse can be defined as a systematic set of steps to reverse an unfavorable decision by a classification model. We generate counterfactual explanations and choose the nearest counterfactual, based on which the correction steps are generated. In addition, unlike existing pose correction methods, CARE utilizes diverse counterfactual explanations to introduce flexibility in the obtained corrected poses. Our contributions are summarized as follows:

- We propose a novel approach to the human pose correction problem based on algorithmic recourse. To the best of our knowledge, this is the first such formulation of human pose correction.
- We develop CARE, an end-to-end system for explainable pose correction with a wide range of applications including fitness monitoring and healthcare; and extensively evaluate it on three diverse datasets - Yoga-20, Pilates-32, and American Sign Language (ASL).
- As part of our experimental studies, we augment existing datasets with incorrect poses for each class, thus helping provide test sets for evaluating pose correction systems. Our augmented datasets will be made publicly available.
- We introduce a novel evaluation metric – Weighted Pose Correction Error (WPCE), to judge the quality of a corrected pose obtained from such a pose correction system.

2. Related Work

In this section, we discuss earlier work from multiple perspectives that could be viewed as connected to our work, viz., pose estimation/classification, as well as pose correction systems.

Pose Estimation and Classification: Pose estimation is a well-studied computer vision task that aims to infer a set of keypoints representing the pose depicted in an image [2, 7, 25, 31]. Building upon pose estimation, pose classification enhances the understanding of the pose by assigning a semantic label or pose category to each instance. While related tasks like human action recognition focus on analyzing videos of human-object interactions [5, 29, 33], pose classification finds versatile applications across various domains. For instance, it plays a crucial role in face recognition [16], surveillance [20, 21], gesture recognition [23, 32], and human-robot interaction [11, 24]. Among the different kinds of pose classification efforts such as head, hand and body pose estimation, our work focuses on the study of full-body poses, with particular emphasis on intricate postures such as in yoga and pilates. There has been a recent increase in efforts on yoga pose classification [6, 8, 13, 14, 19, 30], highlighting the increasing attention on automatically understanding such poses. However, these aforementioned efforts only perform pose classification, and do not consider pose correction.

Pose Correction: Compared to pose classification research, the study of pose correction has been relatively limited. Some works have addressed pose correction in the context of yoga. Katayama et al. [10] introduced a privacy-preserving framework utilizing point cloud extraction to evaluate a user's sitting posture at a work desk. Additionally, Kishore et al. [12] proposed a voice-based feedback system that offers instructional cues for correcting yoga poses. To facilitate research in 3D fitness monitoring, 3D datasets such as Fit3D [4], 3D-Yoga [15], and EC3D [34] have been created. Nevertheless, the acquisition of RGB-D and motion capture data requires specialized equipment, limiting accessibility for general users. To overcome this limitation, CARE uses 2D image data, enabling easier deployment through a smartphone camera, thereby promoting wider accessibility and usability.

Existing pose correction studies have certain limitations. Some require specialized sensors [4,34], while others demonstrate efficacy only for a restricted set of poses [1,34]. Additionally, certain approaches provide rudimentary feedback [22], and some lack comprehensive quantitative analysis of their pose correction module [10, 12]. In our work, we address these limitations by: 1) devising an explainable pose correction system based on algorithmic recourse [9] to offer clear and interpretable decisions, 2) developing a system that seamlessly operates with easily obtainable 2D image data, and 3) conducting a comprehensive evaluation on diverse datasets to establish the effectiveness and versatility of CARE.

3. Proposed CARE Framework

Our overall framework is mainly comprised of a pose classifier, a counterfactual generator and an algorithmic recourse module. We describe each of these below.

3.1. Background And Preliminaries

Counterfactual Explanations: Counterfactual explanations (CFE) are used to explain a deep neural network's model prediction using an approach that seeks to find an alternative input or scenario that, if applied to the model, would have resulted in a different prediction or decision. By providing alternative scenarios [28], a CFE can help users and stakeholders understand the decision-making process of complex machine learning models and identify potential biases or limitations in the model's predictions. It is especially useful in high-stakes applications, such as medical diagnosis or finance, where it is critical to understand why a particular decision was made by a model. A CFE is typically obtained using an optimization formulations that aims to find the minimum perturbation or change to the original input data that would cause the model's output to change to the desired or alternative outcome, subject to suitable constraints.

$$\begin{aligned} \mathbf{x}' = \text{arg min}_{\mathbf{x}'} \left(dist(\mathbf{x}, \mathbf{x}') + \lambda r(\mathbf{x}, \mathbf{x}') \right) \\ \text{s.t } \mathcal{M}_{pose}(\mathbf{x}) \neq \mathcal{M}_{pose}(\mathbf{x}') \end{aligned} \tag{1}$$

Given a factual input $\mathbf{x}$ and a decision $\mathcal{M}_{pose}(\mathbf{x})$ generated by a model $\mathcal{M}_{pose}(.)$, the above optimization problem aims to find a counterfactual explanation $\mathbf{x}'$ which can alter the original decision $\mathcal{M}_{pose}(\mathbf{x})$, with minimal perturbation to x. In the context of classification tasks, the objective function that is minimized may be a combination of a factor of the classification loss and a regularization term λ to ensure that the perturbation is minimal.

Algorithmic Recourse: With growing use of machine learning models in decision making in several critical applications (e.g. medicine, law, finance), there is a need for such decisions to be explainable. In this context, algorithmic recourse [9] goes beyond counterfactual explanations by describing concrete actions that need to be taken to reverse a possibly unfavorable decision made by a machine learning model. Building on counterfactual generation, algorithmic recourse is formulated [26] as follows:

$$\begin{aligned} \delta^* \in \underset{\delta}{\text{argmin}} \, \text{cost}(\delta; \mathbf{x}) \text{ s.t. } \mathcal{M}_{pose}(\mathbf{x}') \neq \mathcal{M}_{pose}(\mathbf{x}), \\ \mathbf{x}' = \mathbf{x} + \delta, \\ \mathbf{x}' \in \mathcal{P}, \delta \in \mathcal{F} \end{aligned} \tag{2}$$

Given a factual input $\mathbf{x}$ and a decision $\mathcal{M}_{pose}(\mathbf{x})$ generated by a model $\mathcal{M}_{pose}(.)$, the above optimization problem aims to find the smallest action δ^* to obtain a counterfactual explanation $\mathbf{x}'$ which can alter the original decision $\mathcal{M}_{pose}(\mathbf{x})$. A set of domain-specific constraints related to plausibility $\mathcal{P}$ and feasibility $\mathcal{F}$ can be optionally applied [26] to the optimization problem.

In CARE, we propose an explainable pose correction system for 2D image data based on counterfactual explanations and use algorithmic recourse to obtain actionable recommendations that can be taken to correct the pose. In particular, we are not just interested in assessing the goodness of the pose but also recommending actionable interventions to correct that pose.

3.2. Pose Classifier

As stated earlier, while RGB-D and motion capture data are much more robust to view point changes, we focus on 2D image data which is more accessible. To obtain pose keypoints data in our system, we consider a data distribution D comprising RGB images representing human poses. Our first step involves utilizing a pre-trained pose estimation model $\mathcal{M}_{keypoints}$ to extract pose keypoints for each image. It is worth noting that the pre-trained model may not be specifically trained on D, which means the obtained keypoints may contain some noise. Nevertheless, our pose

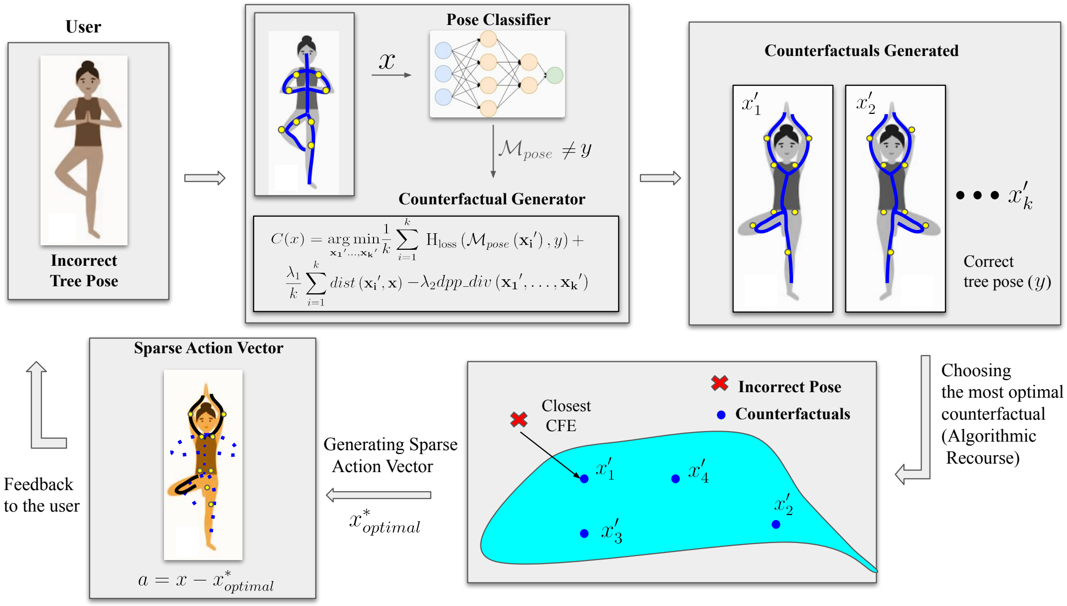

Figure 2. **CARE Framework:** Given an incorrectly formed pose, we extract pose features and pass them through the counterfactual generator. A set of CFEs are generated using algorithmic recourse-based constraints. We then find the most optimal counterfactual closest to the incorrect pose and generate a corrective action vector to help the user correct their pose.

correction system is designed to handle such noise. Given an image I from the distribution D, we extract the keypoint set K using the pre-trained model:

$$K := \mathcal{M}_{keypoints}(I)$$
$$\text{where } K = \{k_i\} \text{ for } i = 1, 2, 3, .., N$$

Traditional pose estimation methods provoide a set of keypoints for each pose. However, for pose correction, the angles formed between different joints in the human body are crucial. We hence convert the keypoint vector into a list of angles that contribute to the pose. Specifically, we compute the angle formed by keypoints k_1, k_2, and k_3 at k_2 as follows:

$$\angle k_1 k_2 k_3 = cos^{-1}\left(\frac{\overline{k_2 k_1}.\overline{k_2 k_3}}{\|\overline{k_2 k_1}\|\|\overline{k_2 k_3}\|}\right)$$

where $\overline{k_i k_j}$ represents a vector connecting the keypoints k_i and k_j. By applying the aforementioned approach, we map each image I in the dataset to its corresponding pose vector (or angle vector). This process allows us to generate a derived data distribution D_A consisting of angle vectors derived from the original data distribution D. The distribution D_A consisting of pose angle vectors is now used to train a fully connected neural network model $\mathcal{M}_{pose}$ as a pose classifier. This network is designed to classify each vector into one of C pose categories. In summary, given an image of a person performing a pose, we generate pose keypoints using a pre-trained pose estimator, extract pose angle vectors, and classify the pose into one of C categories.

In order to study a pose correction system, we require a dataset of poses that have been performed incorrectly. For a given pose class c_i, an incorrect pose is one in which at least one of the angles is erroneously executed while attempting to perform the pose c_i. While it is possible to automatically generate such a dataset by randomly perturbing the angle vectors from D_A, such random perturbations often lack realism due to the constraints of human body flexibility. To generate more realistic 'erroneous poses' for each pose class, we impose additional constraints (e.g. the new joint angle should maintain a certain angle range for a given joint) on the perturbations of pose angle vectors from each class in the training data. This approach allows us to create an incorrect pose dataset $\widetilde{D_A}$ that is closer to the real world. Each vector in this dataset corresponds to a negative class $\tilde{c_i}$ (any class other than c_i). For instance, a perturbed pose vector that deviates from the correct execution of the Bow Pose in a Yoga dataset will be assigned to the negative class "not Bow Pose".

3.3. Counterfactual Generation and Algorithmic Recourse

Given an input feature $\mathbf{x}$ and its corresponding output y from a machine learning model $\mathcal{M}_{pose}$, a counterfactual explanation, $\mathbf{x}'$ is a perturbation of $\mathbf{x}$ to generate a different or desired output y by the same model or algorithm $\mathcal{M}_{pose}$.

$$x' = \arg\min_{x'} \left(\mathrm{H_{loss}}(\mathcal{M}_{pose}(\mathbf{x}'), y) + |\mathbf{x} - \mathbf{x}'|\right)$$

where $\mathrm{H_{loss}}$ is the hinge loss. We use the counterfactual is closest to the input instance for feasibility.

However, algorithmic recourse not only considers feasibility, but also actionability for the user to achieve the desired outcome. Actionability is accomplished through the generation of counterfactuals by perturbing only the mutable features of the input instance. For all counterfactuals generated, the overall loss function is defined as:

$$C(\mathbf{x}) = \underset{\mathbf{x_1}',\ldots,\mathbf{x_k}'}{\arg\min} \frac{1}{k}\sum_{i=1}^{k} \mathrm{H}_{\text{loss}}\left(\mathcal{M}_{pose}\left(\mathbf{x_i}'\right), y\right) + \frac{\lambda_1}{k}\sum_{i=1}^{k} \mathrm{dist}\left(\mathbf{x_i}', \mathbf{x}\right) - \lambda_2 dpp\_div\left(\mathbf{x_1}', \ldots, \mathbf{x_k}'\right) \tag{3}$$

where the first term is the hinge loss that pushes $\mathcal{M}_{pose}(\mathbf{x}')$ towards y, the second term maintains the proximity between $\mathbf{x}$ with k being the number of counterfactuals and x_i' the counterfactual, and the third term maximizes the diversity of counterfactuals and is implemented following [18] as $\det(S)$, represented as *dpp_div* in (Eqn 3), where S is a kernel matrix with $S_{ij} = \dfrac{1}{1+dist(\mathbf{x_i}',\mathbf{x_j}')}$. λ_1 is the proximity weight and λ_2 is the diversity weight. We subsequently follow the algorithmic recourse formulation (Eqn 2) to obtain an actionable pose correction by considering the nearest counterfactual. An optimal pose correction ensures change in a minimal (sparse) set of features (joint angles in our pose correction framework) to accomplish the desired pose class. In the above mentioned optimization formulation (Eqn 3), the first term ensures the output class is the desired class, y, different from the current predicted class, y'. The second term which is the proximity term ensures minimal changes in the joint angles to achieve the desired class, y optimally. In case of pose correction, the third term (Diversity loss), helps achieve the right variant of the class pose even though the output pose belongs to the desired class y. This is also captured in our experimental results in Figure 5.

3.4. Overall Integrated Pipeline

We now describe our overall integrated pipeline, as also illustrated in Figure 2. Assuming we have a pre-trained pose estimator/classifier, when a new user pose image enters the system and is classified as incorrect, we provide the incorrect pose angle vector data to the counterfactual generator. Immutable features, if any, are also provided to the counterfactual generator. Our formulation in Eqn 3 helps generate only the actionable counterfactuals by leveraging the diversity factor in the loss function. Out of all the generated counterfactuals, we pick the closest one to the incorrect pose. To encourage sparsity in a generated counterfactual, we follow [18] in conducting a post-hoc operation where we restore the value of continuous features back to their values in x greedily until the predicted class changes. This ensures that the subject can reach the desired pose with the least effort. With all these components, the optimal counterfactual satisfies the recourse properties of proximity, sparsity and actionability. We then generate the action vector by taking the difference between the incorrect pose and the optimal counterfactual. This action vector is provided to the user to correct the pose optimally. If the user fails to correct his/her incorrect pose, another set of counterfactuals are generated and the loop continues.

4. Experiments and Results

Datasets: We validate the extensibility of our proposed framework by showing the results on 3 datasets - Yoga-20, Pilates-32 and American Sign Language Dataset. We select the 20 most diverse classes from the Yoga-82 dataset [27], which includes rotated versions of certain poses with identical joint angular values. This dataset contains approximately 29,000 images from 82 pose classes. We focus on single-view poses and choose poses with 2D angles to ensure robustness. The training set for Yoga-20 consists of 2,665 images. The Pilates-32 dataset comprises publicly available images of individuals performing 32 Pilates exercises targeting core muscles. It contains 2,225 training images. We use the American Sign Language (ASL) dataset to evaluate our proposed framework across multiple domains, initially intended for hand gesture recognition. This dataset consists of 28 classes representing each letter of the English alphabet, along with "Space" and "Delete" buttons on a keyboard. The training dataset comprises 48,566 images. More details of these datasets, including sample images, are provided in the Appendix.

Evaluation Metrics: We employ the standard top-1 accuracy metric to assess our pose classifier's effectiveness. For evaluating our pose correction system, we consider the Percentage of Corrected Poses (PCP) metric, computed as: $PCP = \frac{100}{T}\sum_{i=1..T}[\text{ Error } \leq \beta\,]$. Here, T refers to the size of the test set and $[.]$ denotes the Iverson bracket notation. β denotes threshold used to compute the percentage of correct poses. Examples of β values can be seen in the column headings in Table 1. We utilize two other measures of *error*: (i) **Mean Absolute Difference (MAD)**, where we first obtain the mean absolute difference between the corrected pose vector and the ground truth pose vector, i.e. MAD Error $= \frac{|p^i_{corrected} - p_{gt}|}{N}$ where N denotes the length of the pose vector; and (ii) **Weighted Pose Correction Error (Weighted PCE)**, which we introduce in this work. Given an incorrect pose vector p_{inc}, the corrected pose vector $p_{corrected}$ (obtained from our pose correction system) and the ground truth pose vector p_{gt}, we calculate the weighted PCE as follows. Given a pose vector of N joint angles $[a_1, a_2, .., a_N]$, we compare the incorrect pose p_{inc} and the ground truth pose p_{gt} to divide these angles into two disjoint sets: A_C, a set of angles which are already correct in the incorrect pose, and A_I, a set of angles which are incorrect in the incorrect pose. Then,

Dataset	Metric	Method	Thresholds									
			1	2	3	4	5	6	7	8	9	10
Yoga-20	MAD	Medoid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.03+-0.0	0.05+-0.0	0.09+-0.0	0.12+-0.0	0.15+-0.0
		Centroid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.02+-0.0	0.04+-0.0	0.07+-0.0	0.1+-0.0	0.13+-0.0	0.18+-0.0
		Decision Tree	0.05+-0.0	0.06+-0.0	0.08+-0.0	0.15+-0.0	0.25+-0.0	0.37+-0.0	0.48+-0.01	0.59+-0.01	0.67+-0.01	0.75+-0.02
		ANN Baseline	0.0+-0.0	0.05+-0.01	0.28+-0.0	0.6+-0.03	0.8+-0.01	0.89+-0.01	0.93+-0.01	0.95+-0.01	0.96+-0.0	0.98+-0.0
		CARE	**0.17+-0.01**	**0.46+-0.01**	**0.77+-0.03**	**0.9+-0.01**	**0.95+-0.01**	**0.97+-0.01**	**0.98+-0.0**	**0.99+-0.0**	**0.99+-0.0**	**0.99+-0.0**
	Wegihted PCE	Mediod	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.04+-0.0	0.08+-0.01	0.11+-0.01	0.16+-0.0	0.2+-0.01	0.25+-0.01
		Centroid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.03+-0.0	0.06+-0.0	0.1+-0.0	0.13+-0.0	0.19+-0.01	0.25+-0.0	0.31+-0.01
		Decision Tree	0.05+-0.0	0.07+-0.0	0.14+-0.0	0.27+-0.01	0.41+-0.01	0.56+-0.01	0.67+-0.01	0.76+-0.01	0.82+-0.01	0.87+-0.01
		ANN Baseline	0.01+-0.0	0.24+-0.02	0.68+-0.01	0.87+-0.0	0.93+-0.01	0.95+-0.0	0.97+-0.0	0.98+-0.0	**0.99+-0.0**	**1.0+-0.0**
		CARE	**0.64+-0.01**	**0.87+-0.02**	**0.95+-0.01**	**0.97+-0.01**	**0.98+-0.0**	**0.99+-0.0**	**0.99+-0.0**	**0.99+-0.0**	**0.99+-0.0**	**1.0+-0.0**
ASL	**Metric**		Thresholds									
			0.5	**1**	1.5	**2**	2.5	**3**	3.5	**4**	4.5	**5**
	MAD	Mediod	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.03+-0.0	0.05+-0.0	0.09+-0.0	0.12+-0.0	0.15+-0.0
		Centroid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.02+-0.0	0.04+-0.0	0.07+-0.0	0.1+-0.0	0.13+-0.0	0.18+-0.0
		Decision Tree	0.05+-0.0	0.06+-0.0	0.08+-0.0	0.15+-0.0	0.25+-0.0	0.37+-0.0	0.48+-0.01	0.59+-0.01	0.67+-0.01	0.75+-0.02
		ANN Baseline	0.0+-0.0	0.05+-0.01	0.28+-0.0	0.6+-0.03	0.8+-0.01	0.89+-0.01	0.93+-0.01	0.95+-0.01	0.96+-0.0	0.98+-0.0
		CARE	**0.17+-0.01**	**0.46+-0.01**	**0.77+-0.03**	**0.9+-0.01**	**0.95+-0.01**	**0.97+-0.01**	**0.98+-0.0**	**0.99+-0.0**	**0.99+-0.0**	**0.99+-0.0**
	Weighted PCE	Mediod	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.02+-0.01	0.05+-0.0	0.1+-0.01	0.18+-0.0	0.27+-0.01	0.36+-0.0
		Centroid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.04+-0.0	0.1+-0.0	0.17+-0.0	0.24+-0.0	0.33+-0.0	0.41+-0.0
		Decision Tree	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.08+-0.01	0.23+-0.0	0.41+-0.01	0.55+-0.01	0.64+-0.01	0.74+-0.0	0.8+-0.0
		ANN Baseline	0.0+-0.0	0.03+-0.01	0.58+-0.02	0.85+-0.01	0.93+-0.0	0.97+-0.0	0.98+-0.0	0.99+-0.0	0.99+-0.0	**1.0+-0.0**
		CARE	**0.97+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**
Pilates	**Metric**		Thresholds									
			1	2	3	4	5	6	7	8	9	10
	MAD	Medoid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0
		Centroid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.01+-0.0	0.03+-0.0	0.04+-0.0	0.06+-0.0
		Decision Tree	**0.21+-0.01**	0.21+-0.01	0.22+-0.01	0.25+-0.01	0.27+-0.01	0.31+-0.01	0.36+-0.01	0.45+-0.01	0.54+-0.01	0.61+-0.01
		ANN Baseline	0.04+-0.01	0.43+-0.03	0.8+-0.01	**0.97+-0.01**	**0.99+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**
		CARE	**0.21+-0.02**	**0.55+-0.01**	**0.79+-0.0**	0.89+-0.01	0.93+-0.01	0.95+-0.01	0.96+-0.01	0.97+-0.01	0.98+-0.0	0.98+-0.0
	Weighted PCE	Medoid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.02+-0.0	0.03+-0.01
		Centroid	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.0+-0.0	0.01+-0.0	0.03+-0.0	0.05+-0.0	0.07+-0.0	0.09+-0.0	0.12+-0.01
		Decision Tree	0.21+-0.01	0.22+-0.01	0.24+-0.01	0.27+-0.01	0.33+-0.0	0.41+-0.01	0.53+-0.01	0.63+-0.01	0.71+-0.0	0.78+-0.01
		ANN Baseline	0.32+-0.0	0.94+-0.0	**0.99+-0.0**	**0.99+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**	**1.0+-0.0**
		CARE	**0.67+-0.02**	**0.87+-0.0**	0.92+-0.02	0.95+-0.02	0.96+-0.01	0.97+-0.01	0.98+-0.0	0.98+-0.0	0.99+-0.0	0.99+-0.0

Table 1. **Main Results:** We show experimental results of CARE on 3 datasets - Yoga-20, Pilates-32 and ASL dataset. We report the percentage of corrected poses (PCP, scaled to range 0-1) based on two evaluation metrics - 1) Mean Absolute Difference and 2) Weighted Pose Correction Error.

$$WPCE = \alpha \sum_{a_i \in A_I} \Delta a_i + (1-\alpha) \sum_{a_c \in A_C} \Delta a_c$$

where α is a weight close to 0, Δa_i and Δa_c refer to the absolute difference in the angle values between the corrected pose and ground truth pose for incorrect and correct angles respectively. More specifically, a correct angle in a pose matches exactly with the corresponding angle in the ground truth pose. Here, we calculate the sum of absolute differences between angles from the corrected pose and the ground truth pose for both sets A_I and A_C and weight them with α and $1-\alpha$ respectively. Considering $\alpha \approx 0$, this error penalizes any instance where the pose correction system changes an already correct pose angle.

Baselines: While existing works in this domain utilize 3D data or specialized sensors for pose correction, due to the lack of available baselines to compare our work on the same setting, we introduce baseline methods to compare with our proposed framework. We compare our approach (CARE) with the following baselines: (i) *Centroid of Class Data:* For an incorrect pose, we obtain the mean pose vector (from the training set) of its intended class as the corrected pose; (ii) *Mediod of Class Data:* For an incorrect pose, we obtain the median pose vector (from the training set) of its intended class as the corrected pose; (iii) *Decision Tree Regression:* We use a traditional learning model based on decision trees, which takes an incorrect pose vector as input and generates the corrected pose vector as output; and (iv) *NN Regression:* A 4-layer neural network that takes an incorrect pose vector as input and generates the corrected pose vector as output.

Dataset	**Architecture**	**No. of Classes**	**Top-1 Accuracy**
Yoga-20	[256, 128, 64, 32]	20	93.7
ASL	[256, 128, 64]	28	97.7
Pilates	[512, 256, 128]	32	94.8

Table 2. **Pose Classification Performance:** Table showing Top-1 accuracy for each dataset.

Hyperparameters: For all datasets, we use the Mediapipe [17] pre-trained pose estimation model. To train pose classifiers, we train a shallow, fully connected neural network for each dataset (details in Table 2). We use Adam optimizer with a learning rate of 0.001. We define 8 joint angles per vector for Yoga-20 and Pilates-32 datasets to obtain the pose vectors. For ASL, we define a pose vector of 19 joint angles. For obtaining counterfactuals, we follow [18] with default values of 0.5 and 1.0 for proximity and diversity weights respectively. We compute weighted pose correction error by setting α to $\frac{1}{N}$ where N is the number of joints.

Results: We begin with a discussion of the pose classifier's performance, since it is an essential component of our

Dataset	α	Thresholds				
		1	2	3	4	5
Yoga-20	0.1	0.552	0.806	0.874	0.922	0.954
	0.2	0.412	0.766	0.862	0.922	0.954
	0.3	0.326	0.72	0.862	0.926	0.954
	0.4	0.272	0.622	0.84	0.922	0.96
	0.5	0.238	0.54	0.786	0.91	0.964
ASL	α	**Thresholds**				
		0.25	0.5	0.75	1	1.25
	0.1	0.864	0.984	0.996	0.996	0.998
	0.2	0.79	0.982	0.996	0.996	0.998
	0.3	0.67	0.978	0.996	0.998	0.998
	0.4	0.518	0.97	0.996	0.998	0.998
	0.5	0.456	0.888	0.994	0.998	0.998
Pilates	α	**Thresholds**				
		1	2	3	4	5
	0.1	0.64	0.85	0.904	0.94	0.962
	0.2	0.61	0.858	0.918	0.952	0.972
	0.3	0.612	0.86	0.928	0.96	0.972
	0.4	0.55	0.868	0.936	0.966	0.976
	0.5	0.504	0.874	0.946	0.974	0.982

Table 3. **Ablation Study on α:** PCP metric for various thresholds of Weighted Pose Correction Error. We experiment with 4 different α values for each dataset.

pose correction pipeline. Table 2 presents these details; the pose classifier achieves high accuracy, surpassing 90% on all three datasets, which is considered a suitably high range in this field. We subsequently show the evaluation our pose correction system in Table 1, which shows the PCP measure using MAD and Weighted PCE. The mean and standard deviation of the evaluation metrics are reported on three runs. Our pose correction framework consistently outperforms the baselines, even at lower thresholds by a considerable margin indicating that the corrected poses highly incline with the ground truth poses. Additionally, it is observed that the Weighted PCE scores are generally higher than the MAD scores for CARE, especially on the Yoga dataset. This could be due to our framework's focus on correcting incorrect joint angles while preserving correctly aligned angles. Overall, our experiments demonstrate promising results for our framework across all three datasets.

5. Discussions and Ablation Studies

Varying α in Weighted PCE: As seen in the earlier section, the weighted pose correction error metric uses a hyperparameter α which decides the extent of penalization for modifying correct and incorrect angles. Table 3 shows the results of our studies with varying α values. It can be seen that the value of α has a negligible effect on Weighted PCE at higher thresholds. However, at lower thresholds such as 1 degree, Percentage of Correct Poses (PCP) decreases as α increases. This clearly demonstrates that a lower α helps in penalizing the adjustment of already correct angles, because an ideal pose correction system does not modify the already correct joint angles, but rather focuses on correcting the incorrectly formed angles.

Figure 3. **Pose correction using CARE on Half-Moon Yoga (HM) and Low-Lunge Yoga poses:** In both rows, the correctly formed (first image), incorrectly formed (second image) and CFE optimized CARE (third image) for HM (top) and LL (bottom) Yoga poses. HM pose is achieved by bringing down the right lower limb from 84 degrees to form 119 degrees at the right hip joint (rh) and opening the right knee (rk) to 142 degrees. This is the optimum Half-Moon pose using the nearest CFE. With the existing joint angle values, the right hip joint and the right knee joint are the two key joints that need to be changed for the target pose change. To achieve LL, CARE recommends pushing the left lower limb to the ground by making smaller angle at the left hip (lh), i.e.. at the angle formed at lh, between, ls,lh and lk.

Qualitative Results: Figure 3 illustrates pose correction using CARE for Half-Moon and Low Lunge Yoga poses on the Yoga dataset respectively. The action vector is sparse with changes only in right hip (rh) and right knee (rk) values for Half-Moon pose and at lh for Low-Lunge pose. More qualitative results, including ones on the ASL dataset, are provided in the Appendix.

Varying Range of Perturbations in Generation of Augmented Dataset: To see the impact of varying perturbations of incorrect poses in the augmented dataset, we performed a study with different perturbation ranges on all the three datasets. Figure 4 shows the results of our study. We do not include the Centroid and the Medoid baselines in these results, as the corrected poses for these baselines are heuristically determined and don't depend on the perturbations of the incorrect poses. We assess the regression-based baselines and compare them against the proposed CARE method. Expectedly, we see MAD dropping as we increase the range of perturbations for all experiments. However, we see CARE outperforming all baselines across these perturbation ranges unanimously. The baseline regression models perform well on a fixed set of perturbations that they are trained on. CARE provides a flexible framework that allows this improved performance.

Multi-variant/Diverse Counterfactuals of Incorrect Poses: Our counterfactual generation step (Eqn 3) allows the generation of a diverse set of counterfactuals for a given

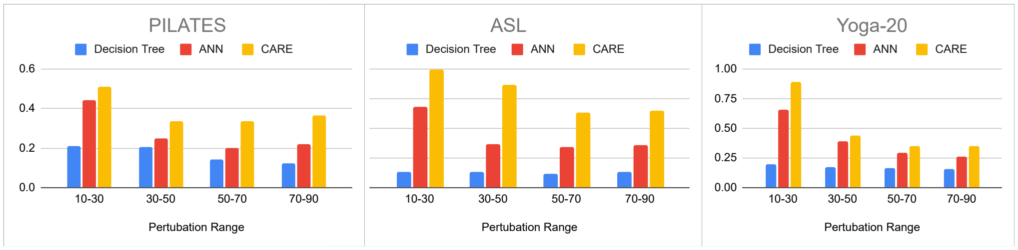

Figure 4. Study of varying range of perturbations while generating the augmented dataset. CARE outperforms the baselines across all these experiments.

Figure 5. **Multi-Variant Explanation:** (Top) Two variants of the Low Lunge pose (Bottom) Tree pose with one incorrect pose and the joint angle figure indicating two possible desired variants involving the left and right joints. joints - e:elbow (blue), s:shoulder (red), h:hip (green) and k:knee (yellow).

Variant	ls	lh	lk	le	rs	rh	rk	re
Input Variant	171	73	111	138	150	115	88	108
Output Variant	171	73	83	138	150	115	88	108

Table 4. **Multi-Variant Explanation:** Counterfactual for the desired variant of the Low-Lunge pose with changed values of joint angles at left knee (lk)

pose. Consider the "Low-Lunge pose" in the Yoga-20 dataset shown in Figure 5. This pose has two variants: Low-Lunge Pose Variant-1 (left) and Low-Lunge Pose Variant-2 (right). CARE leverages the diversity component to output diverse counterfactuals from which we arrive at the optimal (nearest) one from the desired variant. To study this further, we considered a setting where a desired pose is achieved, but we need a different desired variant. CARE CFEs help in making optimal corrections to the current pose variant to achieve the desired pose variant as shown in the bottom two images of Figure 5. Table 4 shows the actionable counterfactual recommended for Tree Pose Variant-2, involving the left knee.

6. Conclusions

In this work, we present CARE: Counterfactual based Algorithmic Recourse for Explainable pose correction, a novel approach that addresses the task of pose correction. While existing works in fitness monitoring have primarily focused on pose classification, we shift our attention to the critical problem of correcting poses. Our CARE system leverages the concept of algorithmic recourse, offering corrective actions when the machine learning model produces unfavorable responses. Comprising an off-the-shelf pose estimator, a pose classifier, and a counterfactual generator, CARE demonstrates a comprehensive solution for pose correction. By extracting pose keypoints from 2D image datasets using the pose estimator, we derive pose angle vectors through post-processing techniques. These vectors serve as the training input for the pose classifier. To rectify incorrect poses, we employ counterfactuals, selecting the closest instance to the incorrect pose as the corrected pose. The element-wise difference between the corrected and incorrect poses yields a sparse vector that represents the necessary corrective action. To evaluate the efficacy of CARE, we conduct experiments on Yoga, Pilates and ASL gesture recognition datasets. The results clearly demonstrate that CARE outperforms baselines across all three datasets. In addition, we introduce a new metric, Weighted Pose Correction Error (Weighted PCE), to assess the quality of corrected poses. This metric provides a comprehensive evaluation of the corrective actions performed by CARE. Avenues for future research include: (i) Developing an automated method for generating a large number of incorrect poses based on human flexibility constraints, and (ii) Improving the robustness of the pose classifier to generate more accurate and effective counterfactuals.

Acknowledgements. This work was partly supported by the Microsoft Academic Partnership Grant and the Department of Science and Technology, India through the DST ICPS Data Science Cluster program. We thank Divyagna Bavikadi and Soumi Chakraborty for their contributions to the Pilates dataset compilation. We thank the anonymous reviewers for their valuable feedback that improved the presentation of this paper.

References

[1] Ardra Anilkumar, Athulya K.T., Sarath Sajan, and Sreeja K.A. Pose estimated yoga monitoring system. *SSRN*, page https://ssrn.com/abstract=3882498, 07 2021. 3

[2] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Lixuan Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking. *ArXiv*, abs/2006.10204, 2020. 1, 2

[3] Bhat Dittakavi, Divyagna Bavikadi, Sai Vikas Desai, Soumi Chakraborty, Nishant Reddy, Vineeth N Balasubramanian, Bharathi Callepalli, and Ayon Sharma. Pose tutor: An explainable system for pose correction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3540–3549, June 2022. 2

[4] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. Aifit: Automatic 3d human-interpretable feedback models for fitness training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9919–9928, June 2021. 1, 2, 3

[5] Deeptha Girish, Vineeta Singh, and Anca Ralescu. Understanding action recognition in still images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1523–1529, 2020. 2

[6] Munkhjargal Gochoo, Tan-Hsu Tan, Shih-Chia Huang, Tsedevdorj Batjargal, Jun-Wei Hsieh, Fady S. Alnajjar, and Yung-Fu Chen. Novel iot-based privacy-preserving yoga posture recognition system using low-resolution infrared sensors and deep learning. *IEEE Internet of Things Journal*, 6(4):7192–7200, 2019. 2

[7] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018. 1, 2

[8] Shrajal Jain, Aditya Rustagi, Sumeet Saurav, Ravi Saini, and Sanjay Singh. Three-dimensional cnn-inspired deep learning architecture for yoga pose recognition in the real-world environment. *Neural Comput. Appl.*, 33:6427–6441, 2021. 2

[9] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: From counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 353–362, New York, NY, USA, 2021. Association for Computing Machinery. 2, 3

[10] Hikaru Katayama, Hamada Rizk, and Hirozumi Yamaguchi. You work we care: Sitting posture assessment based on point cloud data. 02 2022. 1, 3

[11] Cem Keskin, Furkan Kıraç, Yunus Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. volume 7577, pages 852–863, 10 2012. 2

[12] D. Kishore, S. Bindu, and Nandi. Manjunath. Smart ¡i¿Yoga¡/i¿ instructor for guiding and correcting ¡i¿Yoga¡/i¿ postures in real time. volume 15, pages 254–261, 2022. 1, 3

[13] Shruti Kothari. Yoga pose classification using deep learning. 2020. 2

[14] Deepak Kumar and Anurag Sinha. Yoga pose detection and classification using deep learning. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 11 2020. 2

[15] Jianwei Li, Haiqing Hu, Jinyang Li, and Xiaomei Zhao. 3d-yoga: A 3d yoga dataset for visual-based hierarchical sports action analysis. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 434–450, December 2022. 1, 3

[16] S. Li, Xin Ning, Lina Yu, Liping Zhang, Xiaoli Dong, Yuan Shi, and Weidong He. Multi-angle head pose classification when wearing the mask for face recognition under the covid-19 coronavirus epidemic. *2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, pages 1–5, 2020. 2

[17] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019. 6

[18] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 607–617, New York, NY, USA, 2020. Association for Computing Machinery. 5, 6

[19] Chirumamilla Nagalakshmi and Snehasis Mukherjee. Classification of yoga asanas from a single image by learning the 3d view of human poses. *Digital Techniques for Heritage Presentation and Preservation*, pages 37–49, 2021. 2

[20] Javier Orozco, Shaogang Gong, and Tao Xiang. Head pose classification in crowded scenes. In *BMVC*, volume 1, page 3. Citeseer, 2009. 2

[21] Anoop Rajagopal, Ramanathan Subramanian, Elisa Ricci, Radu Vieriu, Oswald Lanz, Kalpathi Ramakrishnan, and Nicu Sebe. Exploring transfer learning approaches for head pose classification from multi-view surveillance images. *International Journal of Computer Vision*, 109, 08 2014. 2

[22] Fazil Rishan, Binali De Silva, Sasmini Alawathugoda, Shakeel Nijabdeen, Lakmal Rupasinghe, and Chethana Liyanapathirana. Infinity yoga tutor: Yoga posture detection and correction system. In *2020 5th International Conference on Information Technology Research (ICITR)*, pages 1–6, 2020. 3

[23] Rubin Bose S. and Sathiesh Kumar. Hand gesture recognition using faster r-cnn inception v2 model. pages 1–6, 07 2019. 2

[24] Bjoern Stenger, Arasanathan Thayananthan, Philip HS Torr, and Roberto Cipolla. Hand pose estimation using hierarchical detection. In *International Workshop on Computer Vision in Human-Computer Interaction*, pages 105–116. Springer, 2004. 2

[25] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), June 2019. 1, 2

[26] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 10–19, New York, NY, USA, 2019. Association for Computing Machinery. 3

[27] Manisha Verma, Sudhakar Kumawat, Yuta Nakashima, and Shanmuganathan Raman. Yoga-82: A new dataset for fine-grained classification of human poses. *CoRR*, abs/2004.10362, 2020. 5

[28] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017. 3

[29] Pichao Wang, Shuang Wang, Zhimin Gao, Yonghong Hou, and Wanqing Li. Structured images for rgb-d action recognition. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1005–1014, 2017. 2

[30] Santosh Yadav, Amitojdeep Singh, Abhishek Gupta, and Jagdish Raheja. Real-time yoga recognition using deep learning. *Neural Computing and Applications*, 31:https://link.springer.com/article/10.1007/s00521–019, 12 2019. 2

[31] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11802–11812, October 2021. 1, 2

[32] Xiaoguang Yu. Hand gesture recognition based on faster-rcnn deep learning. *Journal of Computers*, 14:101–110, 01 2019. 2

[33] Xiangchun Yu, Zhe Zhang, Lei Wu, Wei Pang, Hechang Chen, Zhezhou Yu, and Bin Li. Deep ensemble learning for human action recognition in still images. *Complexity*, 2020:1–23, 01 2020. 2

[34] Ziyi Zhao, Sena Kiciroglu, Hugues Vinzant, Yuan Cheng, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. 3d pose based feedback for physical exercises. pages 17. 1316–1332. Springer, 2022. 1, 3