IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 October 2024, accepted 27 October 2024, date of publication 4 November 2024, date of current version 13 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3491655

RESEARCH ARTICLE

PE-USGC: Posture Estimation-Based Unsupervised Spatial Gaussian Clustering for Supervised Classification of Near-Duplicate Human Motion

HARI IYER AND HEEJIN JEONG, (Member, IEEE)
The Polytechnic School, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, AZ 85212, USA

Corresponding author: Heejin Jeong (heejin.jeong@asu.edu)

This work was supported by the National Institute for Occupational Safety and Health (NIOSH) under Grant T42OH008672.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of Arizona State University under Application No. STUDY00016442.

ABSTRACT Near-duplicate human motion classification presents significant challenges due to the subtle differences and high similarity between actions. This paper introduces a posture estimation-based Gaussian Mixture Model (GMM) clustering algorithm as an enhancement to traditional pixel-based Convolutional Neural Networks (CNNs). The CNN architectures evaluated include ResNet-18, SqueezeNet, DenseNet, and MobileNet, which are used to classify images based on pixel data of images extracted from videos of participants performing chopping, sawing, and slicing tasks. While these CNN models perform well in extracting deep hierarchical features from images, differentiating between near-duplicate tasks can be challenging due to a lack of context in cross-frame human motion. In contrast, the posture-based approach focuses on capturing the spatial and temporal patterns of human body landmarks during task execution, using posture landmark points to classify human motion. Notably, posture-based task classification outperformed pixel-based task classification by 7.2%, with a lesser demand for image frame rate to achieve better accuracy. As the frames per second (FPS) increased from 1 to 30, the accuracy of posture-based classification improved from 76.3% at 1 FPS to 96.97% at 30 FPS. Additionally, we evaluated our model using the UCF Sports Action dataset as a benchmark to compare with state-of-the-art human task classification methods. These comparative analyses highlight the strengths and limitations of each approach, demonstrating that integrating posture data can enhance the classification accuracy of near-duplicate human motion.

INDEX TERMS Posture landmarks, kinematics-aware motion analysis, deep learning, feature extraction, frame rate optimization, hierarchical features, image processing, spatial clustering, task classification, temporal patterns.

I. INTRODUCTION

Near-duplicate human motion classification is a crucial and challenging task in computer vision and pattern recognition [1], [2]. The ability to accurately classify near-duplicate tasks, which exhibit subtle differences and high similarity between actions [3], is vital for various applications like human-machine interaction [4], sports analytics [5], healthcare [6], and workplace safety [7]. Traditional pixel-based approaches [8], [9], primarily using Convolutional Neural Networks (CNNs) [10], [11], have shown significant promise in various image classification tasks. However, they often fall short when dealing with near-duplicate motions due to the inherent complexity and subtle variations in human activities. Human motion analysis [12], [13], [14] has been an active research area for decades, driven by its broad applicability and the growing demand for intelligent systems capable of understanding and interpreting human

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

activities [15], [16]. Accurate motion classification can improve interactive gaming and training experiences [17], physical therapy techniques [18], and workplace safety [19] by monitoring and analyzing workers' actions to prevent injuries. Early methods used techniques such as Hidden Markov Models (HMMs) [20] and Dynamic Time Warping (DTW) [21] to analyze motion sequences [22]. These methods, however, often required extensive domain knowledge to design effective features and had potential issues with generalization across different datasets and scenarios. The advancements in deep learning, particularly CNNs, have shifted the focus towards automated feature extraction and end-to-end learning frameworks that can learn directly from unprocessed data, significantly improving performance in various tasks [23], [24]. CNNs have been remarkably successful in image classification, object detection, and segmentation tasks due to their hierarchical feature extraction capabilities [25], [26], [27]. Despite the advancements in this field, near-duplicate motion classification remains particularly challenging [1], [2], [28] due to the minute differences between actions that can result in significant performance drops in traditional classification models. The primary issue lies in the nature of the data: human motions involve temporal dynamics [29] and spatial relationships [30] that are difficult to capture using static image frames alone. Near-duplicate tasks, such as different styles of chopping vegetables or varying techniques of a golf swing, often appear very similar in pixel-based representations but may involve distinct movements that are critical for accurate classification. To address these challenges, researchers have explored various strategies, including the use of optical flow [31], 3D CNNs [26], and recurrent neural networks (RNNs) [32] to incorporate temporal information into the classification process. Optical flow methods compute the motion of objects [33], [34] between consecutive frames, providing a richer representation of dynamic activities. However, these methods can be computationally intensive and sensitive to noise. 3D CNNs expand the traditional 2D convolution operations to the dimension of time [35], capturing spatiotemporal features directly from video data. While effective, 3D CNNs often need significant computing resources and extensive annotated training data [36], [37]. RNNs, particularly Long Short-Term Memory (LSTM) [38] networks, are designed to handle sequential data and have shown promise in modeling temporal dependencies in motion sequences. Yet, they can be challenging to train and may suffer from issues such as vanishing gradients and lengthy training duration [39].

To address the limitations of pixel-based image classification methods, the current study proposed the use of posture estimation [40], [41] and Gaussian Mixture Models (GMMs) [42]. Posture estimation involves detecting and tracking landmark points on the human body to model the configuration of limbs and joints [43]. This method shifts the focus from pixel-level image analysis to understanding the spatial arrangement and movement of the body, providing a more contextually relevant representation of human motion. By capturing the positions and movements of key body parts, posture-based methods can differentiate between near-duplicate tasks more effectively than pixel-based techniques, which may overlook subtle yet critical differences. GMMs offer a probabilistic approach to clustering and classification [44], [45], modeling data as a mixture of several Gaussian distributions. This approach is particularly well-suited for posture-based analysis, where the spatial distribution of landmark points [46], [47] can vary widely between different activities. GMMs can capture the variability and uncertainty in the data, providing a framework for classifying intricate motion patterns. In this study, we aimed to provide a comparison between pixel-based CNN models and posture-based Gaussian clustering for near-duplicate motion classification. The CNN models evaluated include ResNet-18 [48], SqueezeNet [49], DenseNet [50], and MobileNet [51]. These models are tested on a dataset of culinary tasks, which involve repetitive and near-duplicate actions with the regular kitchen knife. This study aimed to advance the understanding of near-duplicate motion classification by providing a detailed comparison of pixel-based and posture-based methods. By using the strengths of each approach and integrating spatio-temporal information, we sought to develop a classification framework that could have diverse applications in real-world scenarios. The findings of this research will contribute to the ongoing research in improving human motion analysis [15] and help build more intelligent and context-aware systems.

This paper offers the following contributions:

1. We introduce the Posture Estimation-based Unsupervised Spatial Gaussian Clustering (PE-USGC) for near-duplicate human motion classification, improving accuracy by incorporating spatial and temporal patterns in human motion into the analysis.
2. We performed an evaluation of multiple CNN architectures, including ResNet-18, SqueezeNet, DenseNet, and MobileNet, to classify near-duplicate tasks. This comparative analysis revealed the strengths and weaknesses of each model in motion classification compared to PE-USGC.
3. The PE-USGC method demonstrated that posture-based classification outperforms pixel-based methods, at lower frame rates, requiring less storage, memory, and computational resources while maintaining accuracy.
4. A benchmark analysis has been performed with state-of-the-art (SOTA) techniques for human task classification on the University of Central Florida (UCF) Sports Action dataset [52], [53].

The paper is organized as follows: Section II reviews the related work on pixel- and posture-based near-duplicate human motion classification methods. Section III details the methodology and data collection. Section IV explains the data analysis, including the CNNs used, such as ResNet-18, SqueezeNet, DenseNet, and MobileNet, as well as the posture-based task classification approach. Section V

presents the results of two approaches, comparing the performance of pixel- and posture-based methods. Section VI discusses the results and interprets the findings based on other studies and literature. Finally, Section VII concludes the paper and discusses potential future research directions, highlighting limitations that could be areas for improvement.

II. LITERATURE REVIEW

A. ResNet-18

ResNet-18 is a CNN that has various applications [54], [55], [56], [57] in deep learning. It is part of the Residual Network (ResNet) family [48], which introduced a method for training deep networks using residual learning frameworks. The main feature of ResNet-18 is the residual block [58] as shown in Equation (1), which helps address the vanishing gradient problem by allowing gradients to flow more easily through the network. In traditional CNNs, increasing network depth can lead to higher training error [59], known as the degradation problem. ResNet addresses this by using shortcut connections [60] that bypass one or more layers, enabling the network to learn residual functions relative to the layer inputs rather than unreferenced functions. The formulation of a residual block is:

$$y = F(x, W) + x \tag{1}$$

where x is the block's input, and W denotes the weights of the layers within the block.

ResNet-18 uses identity shortcut [48] connections in its residual blocks to allow gradients to flow through the network, preventing them from vanishing [61]. Comprising 18 layers, including convolutional layers, batch normalization [62], ReLU activation [63], and pooling layers [64], ResNet-18 is structured into stages with multiple residual blocks, capturing various levels of abstraction from simple to complex patterns. It can be applied to classify near-duplicate human motion, using object detection and image analysis. Beyond its current uses, ResNet-18 can be applied to new challenges, such as real-time motion recognition, gesture detection in virtual reality, and robotic navigation.

B. DenseNet

DenseNet [50] is a deep CNN designed to improve feature reuse [65] and gradient flow through its dense connectivity. In DenseNet, every layer receives input from all previous layers and propagates its output to all sequentially subsequent layers [66], creating a network where features are shared across layers. This connectivity [50] can be expressed as:

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]) \tag{2}$$

where x_l is the output of the l-th layer, H_l represents operations like Batch Normalization, ReLU, and convolution, and $[x_0, x_1, \ldots, x_{l-1}]$ denotes the feature map concatenation from preceding layers.

In the context of near-duplicate motion classification, DenseNet's architecture allows each layer to receive inputs from all preceding layers, enabling the network to capture and retain detailed information about subtle variations in human motion across different frames. The sharing of feature maps across layers, as described by Equation (2), ensures that fine-grained features learned in earlier layers are directly available to subsequent layers for motion pattern recognition. This capability is particularly useful in distinguishing nearly identical human motions [67], where the differences may be minimal but critical for accurate classification. By using DenseNet, the model can effectively learn and discriminate between these subtle differences, making it well-suited for applications in fields such as sports analysis [5], rehabilitation [68], and ergonomic assessments [69].

C. SqueezeNet

SqueezeNet [49] is a comparatively lightweight CNN architecture designed to match AlexNet's accuracy with 50-times lesser parameters. The architecture focuses on reducing model size (less than half a megabyte) while maintaining strong performance, primarily using nine Fire modules (fire 1 to fire 9) each comprising of Squeeze and the subsequent expansion convolution layers, as shown in Equation (3). Each Fire module consists of a Squeeze layer with 1×1 convolutions to reduce input channels, followed by an Expand layer with a mix of 1×1 and 3×3 convolutions, efficiently balancing parameter count and feature learning [70], [71]. This process is represented as:

$$\text{Fire}(x) = \text{Expand}(\text{Squeeze}(x)) \tag{3}$$

For near-duplicate human motion classification, SqueezeNet's design and efficient feature extraction capabilities [49] can be useful. Near-duplicate human motions, which involve subtle differences in movement patterns, require a model that can distinguish fine-grained [72] features while being computationally efficient. The architecture's ability to compress the input data through the Squeeze layers, followed by the expansion of feature maps using 1×1 and 3×3 convolutions, helps SqueezeNet to capture both local and global motion patterns effectively. SqueezeNet's small model size allows it to be deployed on devices with limited memory and processing power, such as mobile phones and embedded systems for applications involving real-time task classification, without compromising the accuracy of motion classification. Moreover, the architecture's efficiency enables faster inference times, making it suitable for real-time applications where quick decision-making is essential.

D. MobileNet

MobileNet [51] is a set of CNNs built for mobile-based computer vision applications. The key goal of MobileNet is to create a efficient model that can run efficiently on resource-limited devices while maintaining strong accuracy. This is achieved through the use of depthwise separable convolutions [73], [74], a technique that significantly reduces the number of parameters and computational complexity. Depthwise separable convolutions break down standard

convolution into two parts (see Equations 4 and 5). This method reduces computational costs while maintaining the network's capacity to learn complex features by first applying a depthwise convolution, where an individual filter is applied to every input channel, followed by a pointwise convolution that uses 1×1 convolutions to combine the outputs. The process is mathematically expressed as:

$$Z = W_d * X \tag{4}$$

$$Y = W_p \cdot Z \tag{5}$$

where W_d and W_p represent the weights of the depthwise and pointwise convolutions, X is the input, $*$ denotes convolution, and $\cdot$ represents pointwise convolution.

MobileNet's architecture is well-suited for near-duplicate human motion classification due to its efficiency and scalability in handling complex visual patterns on devices with limited processing power. Applications like occupational safety [75], [76] might require handheld devices to detect near-duplicate motions or images in general, characterized by subtle differences. MobileNet's depthwise separable convolutions allow the network to capture intricate spatial features of human motion while keeping the model lightweight enough to run in real-time. The reduced computational cost not only enables faster inference times but also makes MobileNet adaptable to a variety of platforms, including smartphones, wearables, and embedded systems, where real-time motion analysis can be important. The modular nature of MobileNet allows for easy scaling of the network's depth and width, making it flexible for different application needs without sacrificing performance. Previous studies have demonstrated that MobileNet maintains competitive accuracy with significantly fewer parameters, making it an ideal choice for scenarios where both efficiency and accuracy are important [73], [74].

E. POSTURE ESTIMATION

Posture estimation is an application of computer vision, focusing on determining the configuration of the human body from visual data. Over the years, the field has advanced from traditional methods relying on skeleton extraction to modern approaches powered by deep learning [77]. The main goal of posture estimation is to accurately predict the positions and orientations of body parts in images or video frames. Modern techniques often use CNNs and RNNs to capture both spatial and temporal information. A common approach involves regressing the coordinates of landmark points as shown in Equation (6), which represent the locations of joints in the human body. This process can be mathematically formulated as:

$$\widehat{p_i} = f\,(I; \theta) \tag{6}$$

where $\widehat{p_i}$ is the predicted position of the i-th landmark point, I is the input image, and θ represents the model parameters [78].

The accuracy of posture estimation models has been improved by using large-scale annotated datasets and advanced neural network architectures. For instance, the Human3.6M dataset, which includes millions of images with annotated human poses, has played a crucial role in training models [79]. Deep learning frameworks such as OpenPose and AlphaPose have set new benchmarks in the field, achieving SOTA performance in both 2D and 3D posture estimation. OpenPose, introduced by Cao et al. [77], is a multi-person posture estimation framework that detects landmark points for multiple individuals in an image. It uses a two-branch CNN to predict part affinity fields and confidence maps, which are then combined to generate accurate landmark point positions. OpenPose has been widely adopted in applications ranging from sports analysis to human-computer interaction.

AlphaPose, developed by Fang et al. [78], further improves the accuracy of multi-person posture estimation by addressing the occlusion problem. It uses a single-person pose estimator and a pose-guided proposal generator to handle occlusions effectively. The use of Spatial Pose Attention (SPA) mechanisms in AlphaPose enhances the model's capability to target contextually relevant areas, improving the precision of landmark point predictions. Integrating RNNs with CNNs has also been explored to capture temporal dynamics in videos. Models like Pose-RNN use the sequential nature of video data to refine posture estimation over time, resulting in smoother and more accurate predictions [80]. The combination of spatial and temporal information allows these models to handle complex movements and varying poses more effectively. In addition to deep learning approaches, posture estimation has benefited from incorporating prior knowledge and constraints. Techniques such as pose priors and limb length constraints help regularize the model's predictions, ensuring anatomically plausible poses. These constraints are often integrated into the loss function during training, making the model more accurate and realistic estimations [81].

MediaPipe [46] improves the efficiency of real-time posture estimation by using a multi-stage pipeline approach. It uses machine learning models to monitor landmarks points over a frame, including face and body landmarks. MediaPipe's implementation of temporal sampling [46] could identify the sequential flow of frames extracted from the video, for more accurate posture estimation. This approach incorporates prior knowledge through model constraints and priors, ensuring anatomically consistent outputs. These constraints are integrated into the model training process, extracting more accurate and realistic posture estimations, which is crucial for applications in fitness, gaming, and augmented reality. Sánchez-Brizuela et al. [82] created a real-time hand segmentation that uses posture estimation using MediaPipe. The system was developed in Python and attained reasonable accuracy, running at over 90 frames per second (FPS) without relying on advanced computing or code optimization. This shows the scalability of MediaPipe, and a viable option for posture estimation over video frames.

Posture estimation has found applications across various domains, including healthcare, sports, and entertainment. In healthcare, accurate posture estimation is essential for assessing patients' physical conditions and monitoring rehabilitation progress. For example, it can be used to analyze gait patterns, detect movement disorders, and provide feedback for physical therapy exercises [83]. It also facilitates the analysis of complex movements, such as those in gymnastics, martial arts, and dance [84]. The entertainment industry has also used posture estimation for applications such as motion capture and animation. By capturing human movements accurately, posture estimation allows for the creation of realistic animations and virtual avatars, which are used in video games, films, and virtual reality experiences, enhancing the realism and interactivity of digital content [85].

Despite these advancements, several challenges remain in posture estimation. One primary challenge is handling occlusions, where parts of the body are obscured by other objects or individuals. Occlusions can significantly impact the accuracy of landmark point predictions, leading to incomplete or incorrect poses [86], [87]. Techniques like pose-guided proposal generators and attention mechanisms have been developed to address this issue, but further improvements are needed [78]. Another challenge is dealing with variations in body shapes, sizes, and clothing. The diversity of human appearances can affect the model's generalizability across different individuals and scenarios. Large and diverse training datasets, along with data augmentation techniques, are essential to addressing this challenge and improving the model's generalization capabilities [88]. The integration of CNNs, RNNs, and prior knowledge has enabled the development of techniques for predicting human poses from images and videos.

F. SPATIAL CLUSTERING

GMMs are a fundamental tool in unsupervised learning, providing a probabilistic framework for clustering and density estimation [97]. GMMs assume that data is made using a mixture of several Gaussian distributions, each corresponding to a different cluster. This probabilistic approach offers significant flexibility in modeling complex datasets, making GMMs suitable for a wide array of applications, including spatial clustering and posture estimation [98]. A key advantage of GMMs is their ability to handle overlapping clusters. Unlike hard clustering methods such as K-means, which rigidly assign each data point to a single cluster, GMMs assign a probability to each data point for belonging to multiple clusters. This soft clustering capability allows GMMs to capture the inherent uncertainty and variability in real-world datasets, particularly in cases where data points do not fit neatly into distinct categories [42].

GMMs are widely applied across various fields, including image processing, bioinformatics, and speech recognition. In image processing, for example, GMMs are used for image segmentation, aimed at partitioning an image into meaningful regions. By modeling the pixel intensity distribution with a mixture of Gaussians, GMMs can effectively separate different objects and background regions within an image, providing flexibility in handling variations in lighting, color, and texture [99]. This results in more accurate segmentation outcomes. In bioinformatics, GMMs are used to analyze gene expression data, helping to identify clusters of genes with similar expression patterns. By modeling gene expression levels as a mixture of Gaussian distributions, GMMs can uncover underlying biological states and processes, facilitating a deeper understanding of gene functions and interactions [100]. This highlights GMMs' capacity to manage high-dimensional data and detect complex patterns that are challenging for traditional clustering methods. In the domain of speech recognition, GMMs have long been used for modeling the distribution of acoustic features in speech signals, aiding in the classification of phonemes and words [101]. The flexibility of GMMs in handling the variability and uncertainty in speech data makes them particularly well-suited for this task. By modeling the probability distribution of speech features, GMMs enhance the accuracy of speech recognition systems, even in noisy and variable conditions. Recent advancements have seen the integration of GMMs with deep learning frameworks, leading to the development of hybrid models such as Variational Autoencoders (VAEs) and Gaussian Mixture Variational Autoencoders (GMVAEs). These models combine the strengths of GMMs with the expressive power of deep neural networks, using GMMs to capture the latent structure of data while using deep learning techniques to manage complex, high-dimensional inputs [102]. This integration has opened new possibilities for unsupervised learning and generative modeling, enabling more powerful and flexible representations of complex data.

In the context of spatial clustering, GMMs are particularly effective for analyzing the distribution of key landmark points in posture estimation. Posture estimation involves determining the configuration of the human body from visual data, typically by identifying landmark points or landmarks such as joints and body parts [77]. GMMs can model the spatial distribution of these landmark points as a mixture of Gaussian distributions, capturing the variability and uncertainty in human posture. By clustering landmark points based on their spatial coordinates, GMMs can identify distinct posture patterns and classify different types of movements [103]. This probabilistic approach allows for the modeling of overlapping and complex postures, providing a more accurate and nuanced representation of human motion.

For instance, in a motion capture system, GMMs can be used to cluster key landmark points of various body parts, such as the head, shoulders, elbows, and knees. By modeling the spatial distribution of these landmark points, GMMs can identify different postures and movements, such as standing, sitting, walking, or running. The soft clustering approach of GMMs is particularly advantageous for representing transitional postures and movements, where landmark points

TABLE 1. Comparative review of posture estimation methods in human action analysis.

Study	Method	Application	Novelty	Strengths	Limitations	Accuracy
Ahmad & Lee, 2006 [89]	Cartesian optical flow and shape-based features	Human action recognition	Multidimensional hidden Markov model for action recognition	Classification accuracy across different camera angles	Difficulty distinguishing similar actions (e.g., walking, running)	87.5%
Li et al., 2008 [90]	Action graph and GMM	Human action classification	Salient posture-based action graph	Scalability and support for incremental learning	Lack of online learning capability	Over 85%
Yan et al., 2011 [91]	Spatiotemporal movement feature modeling	Human action recognition	Integration of spatiotemporal points of interest	Robustness to video segmentation variations	Limited effectiveness in unstable camera environments	86.6%
Chen et al., 2021 [92]	Pose decomposition based on bones	Anatomy-aware pose estimation	Joint shift loss to train networks using bone properties	Use of information across video frames	Task-specific prediction limitations	Superior performance on Human3.6M and MPI-INF-3DHP datasets
Zheng et al., 2021 [93]	Transformer-based PoseFormer	3D human pose estimation	Temporal correlation modeling using transformers	Strong temporal modeling capability and high accuracy	Need for pre-training for smaller datasets	88.6%
Yang et al., 2022 [94]	U-shaped spatiotemporal transformer	3D human pose estimation	Multi-level and multi-scale spatiotemporal representation	Incorporation of human skeletal topology	-	Outperforms SOTA methods
Li et al., 2023 [95]	Multi-scale coordinate attention framework	Human pose estimation	Coordinate attention mechanism for model enhancement	Reduction of redundancy and improved detection speed	Difficulty in crowded scenes	4.8% improvement over baseline
Ji et al., 2024 [96]	Rigid and non-rigid motion modeling	Human pose estimation	Separation of rigid and non-rigid motions for better estimation	Use of pre-encoded knowledge for 3D pose generation	Limited generalization	14% improvement over SOTA
PE-USGC (current work)	Posture estimation-based unsupervised spatial Gaussian clustering	Near-duplicate human motion classification	Integration of posture kinematics and spatiotemporal joint movements	High accuracy in distinguishing subtle motion variations	Unaccounted body movement styles due to bodily differences	96.97%

may not fit neatly into a single cluster. GMMs can also be applied to analyze temporal sequences of landmark point positions, capturing the dynamics of human motion over time. By modeling the temporal evolution of landmark points as a mixture of Gaussian distributions, GMMs can identify patterns of movement and classify different types of activities. This capability is especially useful in applications such as sports analysis, physical therapy, and human-computer interaction, where understanding the dynamics of human motion is crucial [104]. In addition to their flexibility, GMMs offer several practical advantages. The Expectation-Maximization (EM) algorithm used for parameter estimation is relatively straightforward to implement and can handle missing data and incomplete observations [105]. This makes GMMs suitable for real-world applications where data may be noisy or incomplete.

GMMs also provide a probabilistic framework that can easily incorporate prior knowledge and constraints, enhancing their ability to model complex data. However, GMMs also face challenges. One major challenge is selecting the number of Gaussian components, which can significantly influence clustering results. Model selection criteria such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) are commonly used to determine the optimal number of components, balancing model complexity and goodness of fit [106], [107], [108]. Another challenge is the computational complexity associated with fitting GMMs to large datasets, particularly in high-dimensional spaces. Advances in efficient algorithms and parallel computing techniques have been developed to address this issue, enabling the application of GMMs to big data problems [109]. GMMs are a powerful and versatile tool for clustering and density estimation. Their probabilistic framework and flexibility in modeling complex data distributions make them suitable for diverse applications, ranging from image processing and bioinformatics to speech recognition and posture estimation. The ability of GMMs to handle overlapping clusters and model the spatial distribution of landmark points in posture estimation highlights their potential for analyzing and understanding human motion. As research in this field continues to advance, further methodological and computational developments promise to enhance the

capabilities and applicability of GMMs in tackling complex data analysis challenges.

Table 1 presents a comparative review of various posture estimation methods applied in human action analysis. Each study is characterized based on its methodology, application area, novel contributions, strengths, limitations, and reported accuracy. Notable methods include multidimensional hidden Markov models [89], action graphs [90], and spatiotemporal feature modeling [91], [94] for action recognition [110], as well as recent advancements like Transformer architectures and unsupervised spatial Gaussian clustering [93]. The highest accuracy (96.97%) is achieved by the current work, PE-USGC, which integrates posture kinematics and spatiotemporal joint movements to distinguish subtle motion variations effectively.

III. METHODOLOGY

A. EXPERIMENT DESIGN

This study examined three culinary tasks, as shown in Figure 1, that involved repetitive near-duplicate physical movements: chopping (see Figure 1(a)), sawing (see Figure 1(b)), and slicing (see Figure 1(c)). Chopping [111] requires using a knife to cut food into pieces with a straight downward motion. This action needs force and is used for food items like vegetables and fruits, where the size and shape of the resulting pieces are not necessarily uniform depending on the shape of the food. Sawing [111] is a cutting method that uses a back-and-forth motion, just like a hacksaw is operated. This technique cuts through tougher food items, where a more controlled cutting motion is needed. Slicing [112] is the process of cutting food into fine pieces using a smoothly vertical motion. Slicing can be considered a granular version of chopping. These tasks were carried out using typical food items and kitchen tools commonly found in households or restaurants. Participants were given initial instructions that included basic expectations and safety guidelines to prevent injuries while using sharp tools, based on the outcomes from a previous study [113]. Their actions were recorded using three standard video cameras, capturing the front, left, and right views of each participant.

B. PARTICIPANTS

Thirty-four participants in total performed the three culinary tasks. The lab experiment involved 24 laypersons (12 males and 12 females) aged between 22 and 62 years (M = 27.2; Population SD = 8.46), while the field experiment included 10 professional workers (7 males and 3 females) aged between 23 and 73 years (M = 39.9; SD = 19.39). Most of the skilled workers involved had more than 6 years of experience in culinary work. Only one participant was left-handed. This research adhered to the American Psychological Association Code of Ethics and received approval from the Institutional Review Board at Arizona State University (STUDY00016442). Informed consent was obtained from all participants.

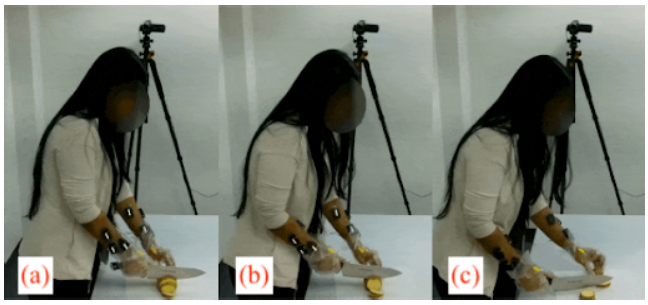


FIGURE 1. Representative near-duplicate culinary tasks performed by participant: (a) Chopping, (b) Sawing, and (c) Slicing.

IV. DATA ANALYSIS

A. PIXEL-BASED IMAGE CLASSIFICATION

The primary objective of this study was to evaluate the performance of various deep learning models in classifying near-duplicate human motions, particularly within the context of culinary tasks such as chopping, sawing, and slicing. The analysis focused on assessing the effectiveness of the four CNN architectures (see Figure 2), using a well-structured experimental framework that allowed for evaluation across key performance metrics, including accuracy, precision, recall, and F1-score.

Each participant's data was systematically organized into distinct categories corresponding to the tasks at hand. The images underwent preprocessing steps for effective training of the CNN models. The preprocessing pipeline included resizing the images to a standard dimension of 224×224 pixels, converting the images into tensors, and normalizing them based on the mean and standard deviation values derived from the ImageNet dataset [48]. These preprocessing steps were crucial in standardizing the input data, ensuring that the models could learn meaningful patterns without being affected by inconsistencies in image size or color distribution.

The computations were conducted using Compute Unified Device Architecture (CUDA) backend, which enabled efficient processing of the data on compatible hardware. This approach was chosen to use the parallel processing capabilities of modern GPUs, accelerating the training process and allowing for more iterations within a reasonable timeframe. The models were trained using the Adam optimizer [114], [115] (see Equations 7, 8, 9, and 10), a common choice in deep learning due to its learning rate adaptiveness when doing model training. The Adam optimizer updates the model parameters θ according to the following equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_\theta J(\theta) \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla_\theta J(\theta))^2 \tag{8}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{9}$$

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{10}$$

where m_t and v_t are the first and second moment estimates, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected moment estimates, α is the learning rate, β_1 and β_2 are decay rates for the moving

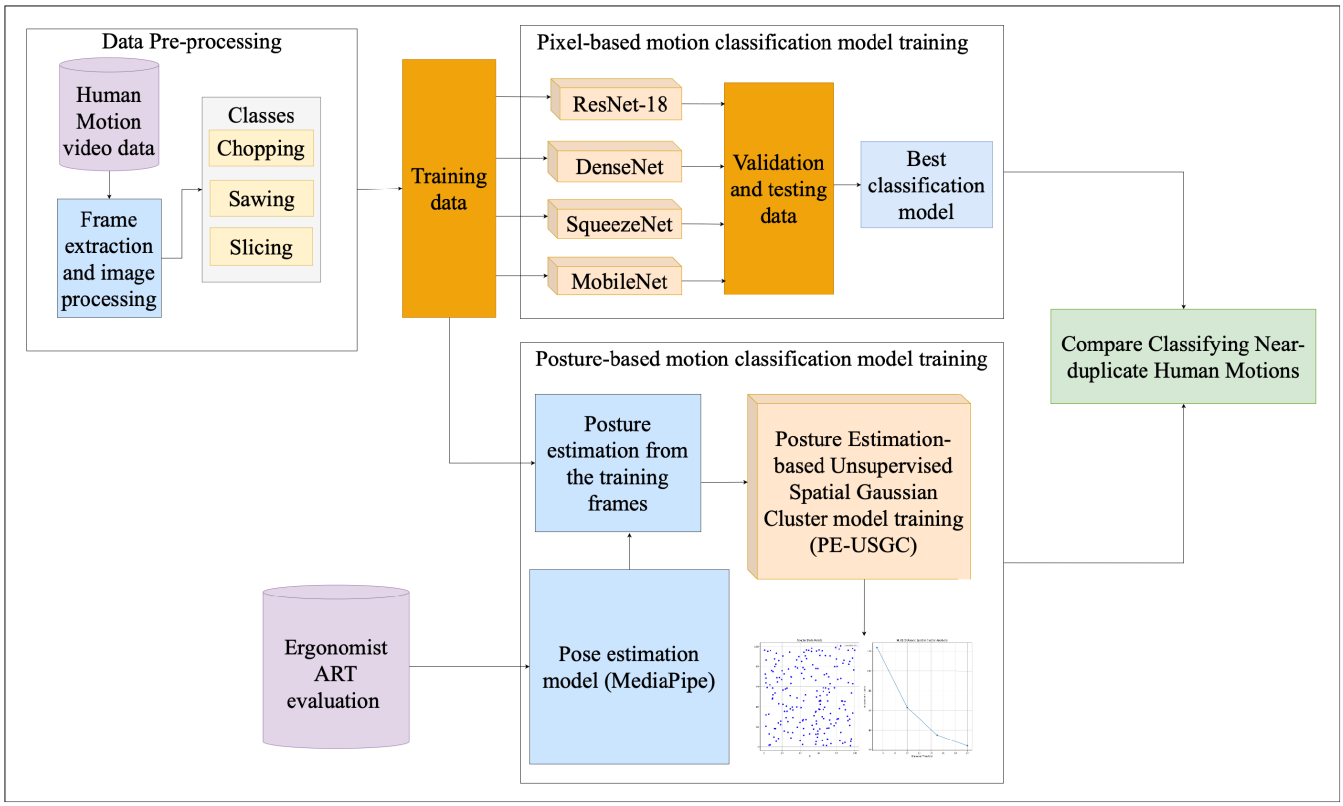


FIGURE 2. Pipeline architecture of the proposed analysis for pixel-based and posture-based near-duplicate human motion classification.

averages, ϵ is a small constant to prevent division by zero, $\nabla_\theta J(\theta)$ is the gradient of the loss function $J(\theta)$ with respect to the parameters, θ_t is the current parameter value, and θ_{t+1} is the updated parameter value.

An initial learning rate of 0.01 was selected, with the learning rate subsequently adjusted through a StepLR scheduler. The learning rate at epoch t was adjusted, as shown in Equation (11).

$$\alpha_t = \alpha_0 \times \gamma^{\lfloor \frac{t}{T} \rfloor} \tag{11}$$

where α_0 is the initial learning rate, γ is the decay factor, and T is the step size.

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{12}$$

where y represents the true label, $\hat{y}$ is the predicted probability, and N is the number of classes.

The training process involved iteratively minimizing the cross-entropy loss function (see Equation 12) [116]. In every epoch, the model parameters were adjusted through backpropagation. This process involved computing the loss function gradients with respect to the parameters of the model and then modifying the parameters in a way that reduced the loss. The progress of the training was monitored using the running loss, which provided a real-time indication of how well the model was learning from the data. Additionally, the model's accuracy was tracked by comparing the predicted labels with the true labels for each batch of images.

Upon completion of the training phase, the models were evaluated on the same dataset using a set of well-established performance metrics. The evaluation metrics included accuracy (see Equation 13), precision (see Equation 14), recall (see Equation 15), and F1-score (see Equation 16), which together provided a assessment of the model's classification capabilities. Accuracy, defined as the ratio of correctly predicted instances to the total count of instances, offered a general measure of the model's effectiveness:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{13}$$

In this equation, TP (True Positives) represents the number of instances where the model correctly identified the positive class, while TN (True Negatives) represents the instances where the model correctly identified the negative class. FP (False Positives) refers to cases where the model incorrectly predicted the positive class, and FN (False Negatives) represents instances where the model incorrectly predicted the negative class.

Precision, which calculated the proportion of true positive predictions relative to all positive predictions, provided details about the model's ability to correctly identify instances of each class without being misled by false

positives:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

Recall, or the true positive rate, measured the model's ability to capture all relevant instances of each class, reflecting its sensitivity:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

Finally, the F1-score, the harmonic mean of recall and precision, provided a measure that accounted for both, false positives and false negatives:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

The results of the evaluation were aggregated for direct comparison between the different models and configurations tested. This structured approach enabled the identification of the most effective model architecture for the task at hand, highlighting the strengths and potential limitations of each model in the context of near-duplicate human motion classification.

B. POSTURE-BASED TASK CLASSIFICATION

Posture-based task classification involved categorizing different activities based on the spatial and temporal patterns of human body landmarks, as shown in Algorithm 1. The landmark points were estimated using MediaPipe and saved. This analysis used data collected where participants performed three culinary tasks, chopping, sawing, and slicing. The process included data loading, feature extraction, clustering using GMM, dimensionality reduction with Principal Component Analysis (PCA), and classification with a Random Forest classifier [117]. To begin with, the dataset was loaded and combined from all participants. Each participant had performed three tasks: chopping, sawing, and slicing. For each task, the landmark data is collected, which includes the x, y, and z coordinates of key points on the body tracked over time. The primary dataset is represented as shown in Equation (17).

$$D = \{(X_i, y_i)\}_{i=1}^{N} \tag{17}$$

where X_i represents the landmark points and y_i the corresponding task label.

From the raw landmark data, we calculated additional features such as velocity and acceleration, as shown in Equations (18) and (19), respectively. Velocity is the rate at which the position changes with respect to time and is calculated using:

$$v_i(t) = \frac{x_i(t+1) - x_i(t)}{\Delta t} \tag{18}$$

where $x_i(t)$ represents the position of landmark i at time t, and Δt is the time difference between frames.

Algorithm 1 Data Processing and Model Training

Input: Dataset directory path $\mathcal{D}$, step size s
Output: Trained Random Forest classifier $\mathcal{C}$, GMM $\mathcal{G}$
Define functions to calculate velocity and acceleration:
    velocities $\leftarrow \frac{\text{landmarks}[1:] - \text{landmarks}[:-1]}{0.033 \times s}$
    accelerations $\leftarrow \frac{\text{velocities}[1:] - \text{velocities}[:-1]}{0.033 \times s}$
Load and combine data from all participants:
    for $p \in \{1, \ldots, 34\}$:
        for (task, label) $\in$ {(Chopping, 0), (Sawing, 1), (Slicing, 2)}:
            Load data df from $\mathcal{D}$
            Subsample data df[:: s]
            Calculate features: landmarks, velocities, accelerations
            Append features and labels
        end for
    end for
return combined_features, labels
Split data into training and testing sets
Fit GMM:
$\mathcal{G} \leftarrow$ GaussianMixture($n\_components = 3$)
Train $\mathcal{G}$ on X_{train}
Predict clusters for X_{train} and X_{test}
Visualize clusters using PCA
Train Random Forest Classifier:
$\mathcal{C} \leftarrow$ RandomForestClassifier($n\_estimators = 100$)
Train $\mathcal{C}$ on X_{train}, y_{train}
Predict y_{pred} on X_{test}
Evaluate the model:
Compute accuracy, precision, recall, F1-score, and confusion matrix
Calculate accuracy for each task:
for label $\in \{0, 1, 2\}$:
    Compute task-specific accuracy
return Model performance metrics

Acceleration, which is the rate of change of velocity, is computed as:

$$a_i(t) = \frac{v_i(t+1) - v_i(t)}{\Delta t} \tag{19}$$

where $v_i(t)$ represents the velocity of landmark i at time t, and Δt is the time difference between frames.

These calculated velocities and accelerations are then combined with the original landmarks to form a feature vector:

$$\boldsymbol{f}_i = [x_i, v_i, a_i] \tag{20}$$

The next step involved splitting the dataset into training and testing sets, as shown in Equation (21). This split ensured that the model can be evaluated on unseen data to assess its generalization performance. The dataset D is partitioned into D_{train} and D_{test} such that:

$$|D_{\text{train}}| + |D_{\text{test}}| = |D| \tag{21}$$

TABLE 2. Performance metrics of PE-USGC at different frame rates averaged over 10 runs of randomized train-test data.

FPS	Chopping [Mean (SD)]				Sawing [Mean (SD)]			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
1	66.47% (±4.24%)	0.77 (±0.03)	0.67 (±0.04)	0.71 (±0.03)	72.94% (±3.70%)	0.82 (±0.04)	0.73 (±0.04)	0.77 (±0.03)
5	88.86% (±1.01%)	0.92 (±0.02)	0.89 (±0.01)	0.90 (±0.01)	91.17% (±0.67%)	0.95 (±0.01)	0.91 (±0.01)	0.93 (±0.01)
10	93.62% (±0.62%)	0.93 (±0.01)	0.94 (±0.01)	0.93 (±0.01)	93.52% (±0.47%)	0.96 (±0.00)	0.94 (±0.00)	0.95 (±0.00)
30	96.39% (±0.50%)	0.96 (±0.01)	0.96 (±0.01)	0.96 (±0.00)	96.16% (±0.26%)	0.98 (±0.00)	0.96 (±0.00)	0.97 (±0.00)
FPS	**Slicing [Mean (SD)]**				**Average across tasks [Mean (SD)]**			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
1	89.49% (±1.73%)	0.76 (±0.03)	0.89 (±0.02)	0.82 (±0.02)	76.30% (±3.22%)	0.78 (±0.03)	0.76 (±0.03)	0.77 (±0.03)
5	95.69% (±0.69%)	0.92 (±0.01)	0.95 (±0.01)	0.93 (±0.00)	91.91% (±0.79%)	0.93 (±0.01)	0.92 (±0.01)	0.92 (±0.01)
10	96.85% (±0.37%)	0.95 (±0.01)	0.97 (±0.00)	0.96 (±0.00)	94.66% (±0.49%)	0.95 (±0.00)	0.95 (±0.00)	0.95 (±0.00)
30	98.36% (±0.24%)	0.97 (±0.00)	0.98 (±0.00)	0.98 (±0.00)	96.97% (±0.33%)	0.97 (±0.00)	0.97 (±0.00)	0.97 (±0.00)

TABLE 3. Performance metrics of CNN architecture at 30 fps dataset.

CNN Architecture	Accuracy by task			Average across tasks			
	Chopping	Sawing	Slicing	Accuracy	Precision	Recall	F1-score
ResNet-18	93.33%	94.06%	95.89%	94.43%	0.94	0.95	0.94
DenseNet	90.91%	83.00%	98.12%	90.68%	0.92	0.90	0.91
SqueezeNet	73.97%	83.64%	89.61%	82.41%	0.78	0.82	0.80
MobileNet	87.01%	90.00%	97.50%	91.50%	0.94	0.92	0.93

Clustering was performed using GMM. GMMs assume that the data is derived from a mixture of several Gaussian distributions, each representing a cluster. This probabilistic approach provides flexibility in modeling the data and can handle overlapping clusters effectively [118]. The GMM was trained on the feature vectors (see Equation 20) from the training set, and the parameters of the model were calculated using the Expectation-Maximization (EM) algorithm [42]. The EM algorithm iteratively maximized the expected complete data log-likelihood, updating the model parameters θ as shown in Equation (22).

$$\theta_{\text{new}} = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \log \left(\pi_k \mathcal{N}\left(x_i \mid \mu_k, \Sigma_k\right)\right) \quad (22)$$

where θ_{new} represents the updated parameters that maximize the expected complete data log-likelihood. Here, π_k, μ_k, and Σ_k are the mixture weights, means, and covariance matrices of the k^{th} Gaussian component, respectively. The expression $\mathcal{N}\left(x_i \mid \mu_k, \Sigma_k\right)$ represents the probability density function of the k^{th} Gaussian component evaluated at the data point x_i. The responsibilities γ_{ik}, defined as

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i \mid \mu_j, \Sigma_j)},$$

denote the posterior probability that x_i belongs to the k^{th} component. The outer sum $\sum_{i=1}^{N}$ runs over all N data points, while the inner sum $\sum_{k=1}^{K}$ runs over all K Gaussian components in the mixture model. The $\arg\max_{\theta}$ term indicates that we are finding the value of θ that maximizes the expected complete data log-likelihood.

V. RESULTS

The performance metrics of the PE-USGC model and the four CNN architectures at different frame rates are presented in the Tables 2 and 3, respectively. The results highlight significant differences in accuracy, precision, recall, and F1-score across the models and frame rates. The training and validation loss and accuracies of the CNN models are shown in Figure 3. For the CNN architectures, ResNet-18 demonstrated the highest average accuracy of 94.43%, followed by DenseNet at 90.68%, MobileNet at 91.50%, and SqueezeNet at 82.41%. Average precision, recall, and F1-score metrics were also highest for ResNet-18, with values of 0.94, 0.95, and 0.94, respectively. The PE-USGC model showed improvement in performance with increasing frame rates. At 1 fps, the model achieved an average accuracy of 76.30%, with precision, recall, and F1-score values of 0.78, 0.76, and 0.77, respectively. As the frame rate increased, the model's accuracy and other metrics improved significantly. At 5 fps, the average accuracy reached 91.91%, with precision, recall, and F1-score values of 0.93, 0.92, and 0.92, respectively. At 10 fps, the model's performance further improved, with an average accuracy of 94.66%, precision and recall both at 0.95, and an F1-score of 0.95. At the highest frame rate of 30 fps, the PE-USGC model achieved its best performance, with an average accuracy of 96.97%, precision, recall, and F1-score all at 0.97. In comparison to the CNN architectures evaluated at 30 fps (Table 3), the PE-USGC model outperformed CNN models across tasks.

VI. DISCUSSION

The results demonstrate that the PE-USGC model is highly effective for classifying human motion, particularly at higher frame rates. Table 2 shows the performance results PE-USGC averaged over 10 runs and Table 3 shows the results for four CNN architectures evaluated. The model's performance shows a significant improvement as the frame rate increases, with the highest frame rate of 30 fps yielding the best outcomes. This trend suggests that higher frame rates provide more detailed temporal information, which enhances the model's ability to accurately classify the tasks. When comparing the PE-USGC model to traditional CNN architectures, it was found that that the PE-USGC model achieved comparable or even better performance at

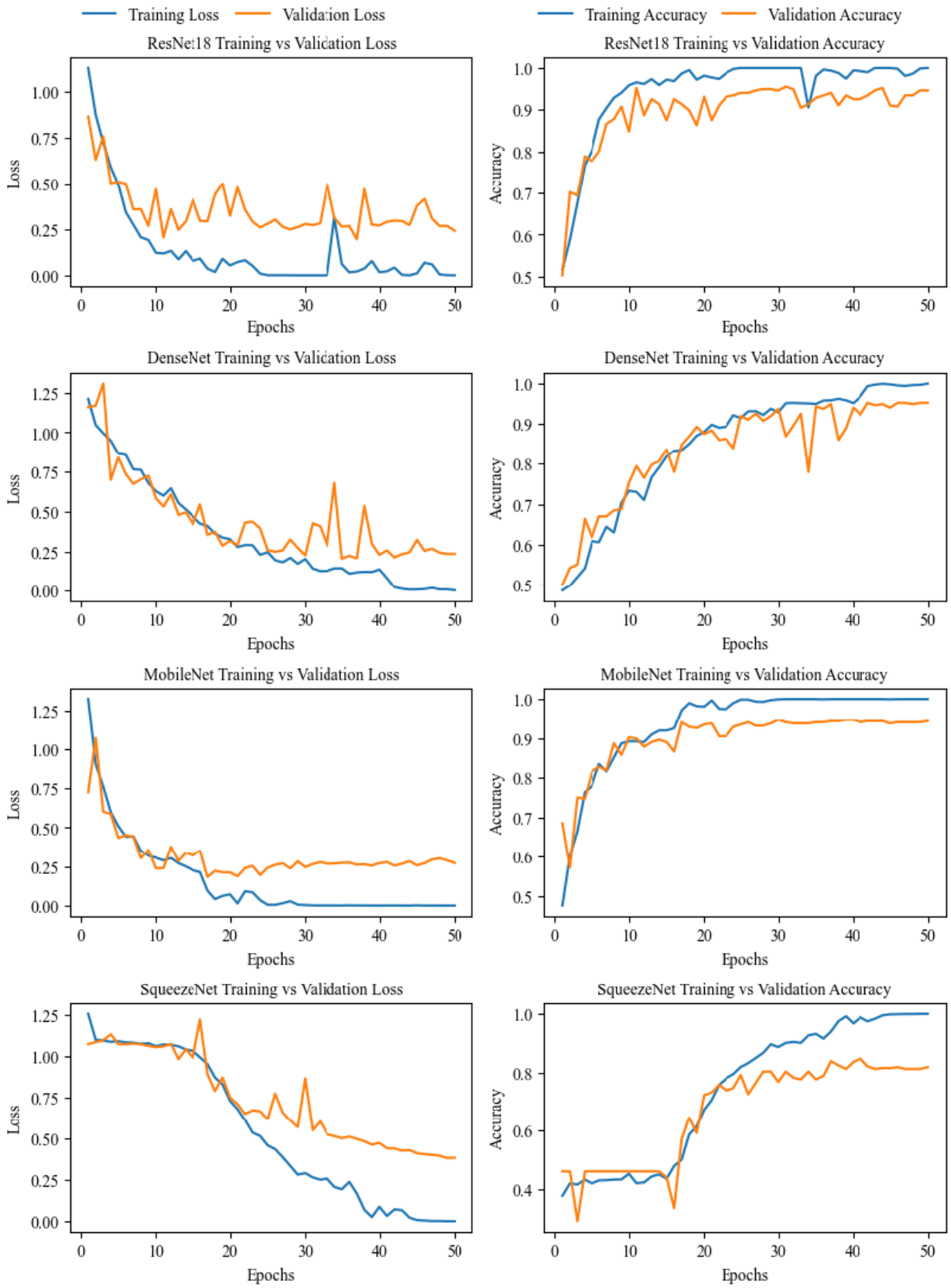


FIGURE 3. Training and validation loss and accuracy curves for the four CNN architectures evaluated.

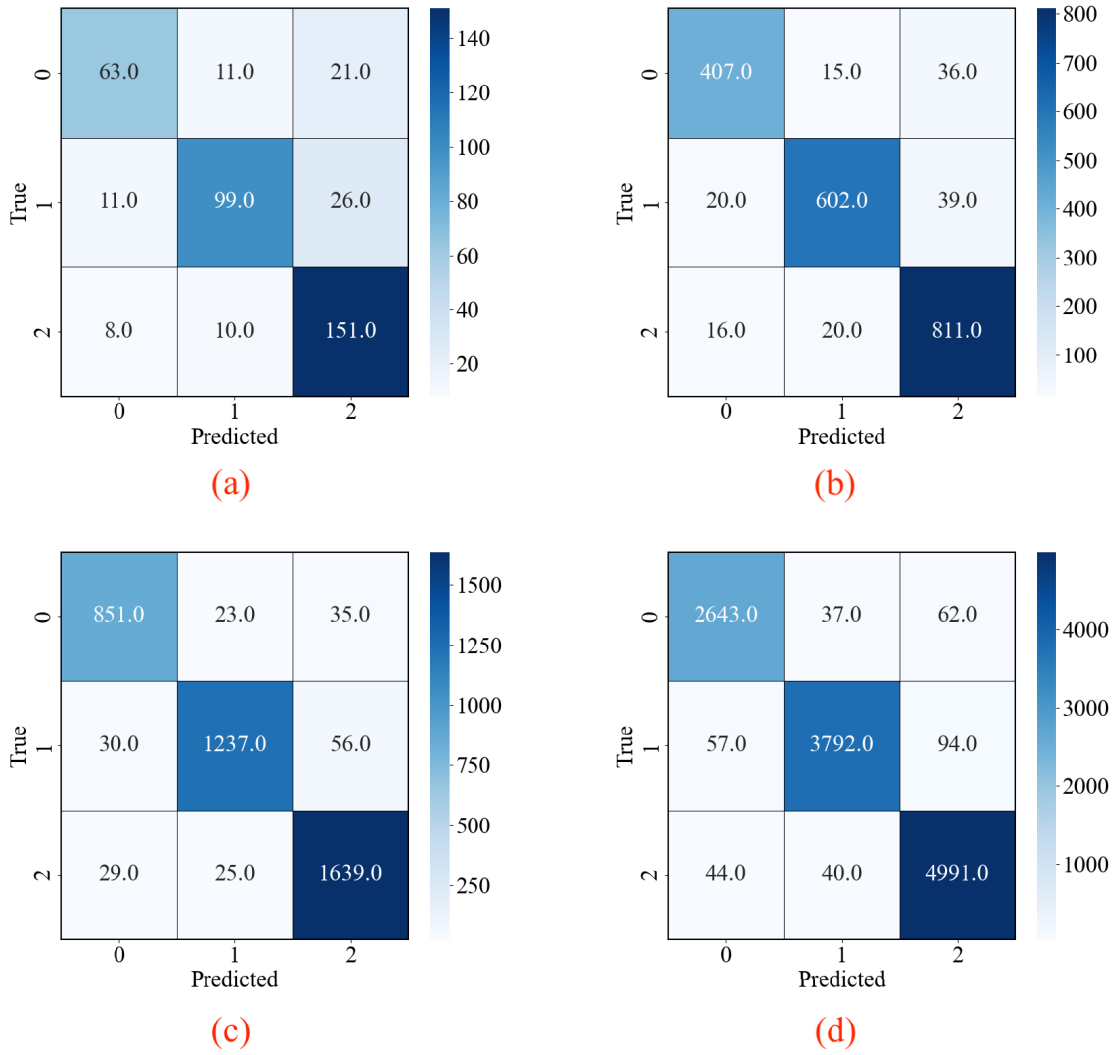


FIGURE 4. Confusion matrices for PE-USGC over 10 runs for classes 0, 1, and 2, representing chopping, sawing, and slicing, respectively. (a) 1 fps, (b) 5 fps, (c) 10 fps, and (d) 30 fps.

lower frame rates. For example, at 10 fps, the PE-USGC model's average accuracy of 94.66% is comparable to that of ResNet-18. This is an important finding, as it indicates that the PE-USGC model can achieve high performance without the substantial computational costs associated with processing at 30 fps.

At 30 fps, the PE-USGC model's performance exceeds that of all tested CNN architectures, including ResNet-18. This finding suggests that the PE-USGC model not only matches but can also surpass the performance of CNN architectures under optimal conditions. The improved performance of the PE-USGC model could be attributed to its ability to integrate posture estimation with unsupervised spatial clustering, which likely provides a more detailed understanding of task-specific movements [83]. Analyzing individual task performance, the PE-USGC model achieved an accuracy of 96.39% for the Chopping task at 30 fps, which, while slightly higher than ResNet-18's 93.33%. At 10 fps, the PE-USGC model's accuracy for Chopping was 93.62%. Even at 1 fps (see Figure 4(a)), the model maintained a reasonable accuracy of 66.47%, which is notable given the significant reduction in temporal resolution [77].

For the Sawing task, the PE-USGC model achieved an accuracy of 96.16% at 30 fps, slightly better ResNet-18's 94.06%. As the confusion matrix shows in Figure 4(c) for 10 fps, the model achieved an accuracy of 93.52%, surpassing DenseNet, MobileNet, and SqueezeNet, which highlights the model's robustness even at reduced frame rates. The model's performance at 1 fps for the Sawing task was 72.94%, which, while lower, still demonstrates reasonable classification capability at this low frame rate. The confusion matrices for 5 fps and 30 fps are shown in Figures 4(b) and 4(d), respectively.

The Slicing task yielded the highest accuracy across all tasks, with the PE-USGC model achieving 98.36% at 30 fps, outperforming all CNN architectures. Even at 10 fps,

TABLE 4. Comparative analysis of PE-USGC and other approaches using the UCF sports action dataset.

Study	Method	Accuracy (%)
El-Assal et al., 2023 [119]	Unsupervised spike-timing dependent plasticity-based learning with convolutional spiking neural networks	50.67
Sanakoyeu et al., 2018 [120]	Unsupervised similarity learning through exemplar-based approach	79.00*
PE-USGC (current work)	Unsupervised Gaussian clustering and Random Forest classification	82.16
Le et al., 2011 [121]	Unsupervised hierarchical learning of invariant spatiotemporal features using Independent Subspace Analysis for robust action recognition	86.5
Sun et al., 2014 [122]	Unsupervised hierarchical slow feature learning	86.6

*7 out of the 10 UCF Sports Action dataset classes were used.

the model achieved an accuracy of 96.85%. At 1 fps, the model had an accuracy of 89.49%, which is notable given the significantly lower spatio-temporal resolution [53]. The observed improvement in precision, recall, and F1-score metrics with increasing frame rates further highlights the effectiveness of the PE-USGC model. These metrics show that the model becomes more accurate and reliable in its classifications as it receives more temporal data. The highest precision, recall, and F1-score values at 30 fps confirm that the model's predictions are both precise and consistent, reducing the likelihood of false positives and false negatives.

The PE-USGC model's ability to achieve high performance at lower frame rates has significant implications for real-world applications. Lower frame rates reduce the amount of data that must be processed, leading to faster and more efficient computations. This is particularly beneficial in resource-constrained environments where computational power and storage are limited. The PE-USGC model's ability to maintain high accuracy and reliability at these lower frame rates makes it a valuable tool for applications requiring efficient and effective classification with limited resources. Overall, the results highlight the PE-USGC model as a powerful and efficient alternative to traditional CNN architectures for classifying culinary tasks. The model's performance at higher frame rates and its competitive performance at lower frame rates highlights its adaptability, making it an alternative for various applications, particularly those where computational resources are limited.

As shown in Table 4, PE-USGC achieves 82.16% accuracy on the UCF Sports Action dataset, which contains 150 video sequences of humans performing 10 sports actions, such as running, diving, and swinging, in diverse backgrounds. Although Le et al. [121] and Sun et al. [122] achieved higher accuracies of 86.5% and 86.6% using hierarchical feature learning and slow feature analysis, these methods rely on pixel-level representations. This could make them more sensitive to variations such as lighting, background clutter, and viewpoint changes, limiting their ability to capture temporal dynamics and motion continuity effectively. Reliance on pixel-level features can make them less effective at distinguishing near-duplicate human motions-actions with subtle differences that are visually similar as shown in Tables 2 and 3.

VII. CONCLUSION AND FUTURE WORK

In this study, we conducted a detailed analysis of near-duplicate human motion classification using both pixel-based CNNs and posture-based GMM classification. Our results show that combining posture estimation with spatial Gaussian clustering enhances classification accuracy, offering a more context-aware understanding of human activities compared to traditional pixel-based methods. While pixel-based CNNs like ResNet-18, SqueezeNet, DenseNet, and MobileNet are effective, posture-based methods particularly excel in handling subtle variations in human motion. The key findings from our analysis include improved accuracy, tolerance to frame rate variability, and computational efficiency. Posture-based task classification outperformed pixel-based methods by focusing on spatial and temporal patterns of human body landmarks, capturing details in near-duplicate tasks that pixel-based approaches often miss. GMMs showed better performance at higher frame rates, suggesting that posture-based methods can effectively use increased frame rates for better classification results. Posture-based technique required fewer image frames, which is advantageous for real-time applications in resource-limited environments.

However, our study had some limitations. First, the research was conducted on a dataset built for culinary tasks, and other types of near-duplicate human motions from other industries needs to be explored. Second, the current posture estimation method has not accounted for occlusions, where parts of the body are obscured, making accurate posture estimation challenging. Third, the study did not account for individual differences in body size, shape, or movement style, which can potentially influence the classification accuracy.

Future research should address these limitations by expanding dataset diversity, improving occlusion handling, and optimizing computational efficiency. Including a wider variety of datasets across different domains, such as sports, healthcare, and workplace activities, could prove the generalizability of posture-based classification approaches. Developing advanced techniques to handle occlusions, such as using depth sensors or multi-view camera setups, will improve accuracy in real-world scenarios. Optimizing the computational efficiency of posture estimation and GMM clustering, through methods like model pruning and hardware accelerators, will be critical for deploying these models in real-time applications. In conclusion, integrating posture estimation with spatial Gaussian clustering represents a significant advancement in the classification of near-duplicate human motions. By addressing the identified limitations and

pursuing further research, we aim to enhance the accuracy and applicability of these methods across various real-world scenarios. The code to PE-USGC model can be found on GitHub [123].

ACKNOWLEDGMENT

The authors would like to thank Dr. Jamie Gorman and Dr. Rob Gray from Arizona State University for their input in formulating the research topic. They would also like to express their gratitude to the recruited participants who performed the experiment. They acknowledge research computing at Arizona State University for providing high-performance computing resources [124] that have contributed to the research results reported within this article.

REFERENCES

[1] L. Morra and F. Lamberti, "Benchmarking unsupervised near-duplicate image detection," *Expert Syst. Appl.*, vol. 135, pp. 313–326, Nov. 2019.
[2] K. K. Thyagharajan and G. Kalaiarasi, "A review on near-duplicate detection of images using computer vision techniques," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 897–916, May 2021.
[3] R. D. Oliveira, M. Cherubini, and N. Oliver, "Looking at near-duplicate videos from a human-centric perspective," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 6, no. 3, pp. 1–22, Aug. 2010.
[4] I. S. MacKenzie, *Human-Computer Interaction: An Empirical Research Perspective*. San Francisco, CA, USA: Morgan Kaufmann, 2013.
[5] S. Kanimozhi, B. R. Priya, K. Sandhiya, R. Sowmya, and T. Mala, "Human movement analysis through conceptual human-object interaction in sports video," Vellore Inst. Technol. (VIT), Tamil Nadu, India, Tech. Rep. 4525389, 2023.
[6] S. M. Lee, D. Lee, and M. J. Schniederjans, "Supply chain innovation and organizational performance in the healthcare industry," *Int. J. Operations Prod. Manage.*, vol. 31, no. 11, pp. 1193–1214, Oct. 2011.
[7] M. S. Christian, J. C. Bradley, J. C. Wallace, and M. J. Burke, "Workplace safety: A meta-analysis of the roles of person and situation factors," *J. Appl. Psychol.*, vol. 94, no. 5, pp. 1103–1127, 2009.
[8] H. Al-Sahaf, A. Song, K. Neshatian, and M. Zhang, "Two-tier genetic programming: Towards raw pixel-based image classification," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12291–12301, Nov. 2012.
[9] R. Goldblatt, W. You, G. Hanson, and A. Khandelwal, "Detecting the boundaries of urban areas in india: A dataset for pixel-based image classification in Google Earth engine," *Remote Sens.*, vol. 8, no. 8, p. 634, Aug. 2016.
[10] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.
[11] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.
[12] A. I. Cuesta-Vargas, A. Galán-Mercant, and J. M. Williams, "The use of inertial sensors system for human motion analysis," *Phys. Therapy Rev.*, vol. 15, no. 6, pp. 462–473, Dec. 2010.
[13] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Comput. Vis. Image Understand.*, vol. 73, no. 3, pp. 428–440, Mar. 1999.
[14] R. Poppe, "Vision-based human motion analysis: An overview," *Comput. Vis. Image Understand.*, vol. 108, nos. 1–2, pp. 4–18, Oct. 2007.
[15] H. Iyer, N. Macwan, S. Guo, and H. Jeong, "Computer-vision-enabled worker video analysis for motion amount quantification," 2024, *arXiv:2405.13999*.
[16] H. Iyer, "Analyzing worker videos for quantifying motion amounts through computer vision," in *Proc. Human Factors Ergonom. Soc. Annu. Meeting*, Los Angeles, CA, USA. SAGE, 2024, Art. no. 10711813241262027.
[17] Z. Li, L. Wang, and X. Wu, "Artificial intelligence based virtual gaming experience for sports training and simulation of human motion trajectory capture," *Entertainment Comput.*, vol. 52, Jan. 2025, Art. no. 100828.
[18] J. Richter, C. Wiede, B. Shinde, and G. Hirtz, "Motion error classification for assisted physical therapy—A novel approach using incremental dynamic time warping and normalised hierarchical skeleton joint data," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, vol. 2, 2017, pp. 281–288.
[19] H. Iyer, J. Isingizwe, R. Eiris, and H. Jeong, "Ladder safety assessment using head-mounted 360-degree camera-based posture estimation overlayed real-time in augmented reality," in *Proc. IEEE Conf. Virtual Reality 3D User Inter. Abstr. Workshops (VRW)*, 2024, pp. 1–4.
[20] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. ASSPM-3, no. 1, pp. 4–16, Jan. 1986.
[21] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.
[22] N. Macwan, A. J. Hude, H. Iyer, H. Jeong, and S. Guo, "High-fidelity worker motion simulation with generative AI," in *Proc. Human Factors Ergonom. Soc. Annu. Meeting*. Los Angeles, CA, USA: SAGE, Aug. 2024, Paper 10711813241262026.
[23] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Arch. Comput. Methods Eng.*, vol. 27, no. 4, pp. 1071–1092, Sep. 2020.
[24] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, M. Mofijur, A. B. M. Shawkat Ali, and A. H. Gandomi, "Deep learning modelling techniques: Current progress, applications, advantages, and challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13521–13617, Nov. 2023.
[25] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," in *Proc. Intell. Comput., Image Process. Appl.*, 2020, pp. 1–16.
[26] B. Kayalibay, G. Jensen, and P. van der Smagt, "CNN-based segmentation of medical imaging data," 2017, *arXiv:1701.03056*.
[27] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.
[28] W.-L. Zhao, S. Tan, and C.-W. Ngo, "Large-scale near-duplicate Web video search: Challenge and opportunity," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2009, pp. 1624–1627.
[29] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.
[30] E. S. L. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adaptation," *ACM Trans. Graph.*, vol. 29, no. 4, p. 1, Jul. 2010.
[31] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," in *Proc. 40th German Conf., Pattern Recognit. (GCPR)*. Stuttgart, Germany: Springer, 2019, pp. 281–297.
[32] S. Grossberg, "Recurrent neural networks," *Scholarpedia*, vol. 8, no. 2, p. 1888, 2013.
[33] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.
[34] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 989–997.
[35] K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma, "T-C3D: Temporal convolutional 3D network for real-time action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
[36] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–8.
[37] H. Jin, Z. Li, R. Tong, and L. Lin, "A deep 3D residual CNN for false-positive reduction in pulmonary nodule detection," *Med. Phys.*, vol. 45, no. 5, pp. 2097–2107, May 2018.
[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[39] H. Fei and F. Tan, "Bidirectional grid long short-term memory (BiGridLSTM): A method to address context-sensitivity and vanishing gradient," *Algorithms*, vol. 11, no. 11, p. 172, Oct. 2018.
[40] C.-F. Juang, C.-M. Chang, J.-R. Wu, and D. Lee, "Computer vision-based human body segmentation and posture estimation," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 1, pp. 119–133, Jan. 2009.
[41] R. Mehrizi, X. Peng, X. Xu, S. Zhang, D. Metaxas, and K. Li, "A computer vision based method for 3D posture estimation of symmetrical lifting," *J. Biomechanics*, vol. 69, pp. 40–46, Mar. 2018.
[42] D. A. Reynolds, "Gaussian mixture models," *Encycl. Biometrics*, vol. 741, nos. 659–663, Jun. 2009.

[43] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Human pose estimation using body parts dependent joint regressors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3041–3048.

[44] I. Prabakaran, Z. Wu, C. Lee, B. Tong, S. Steeman, G. Koo, P. J. Zhang, and M. A. Guvakova, "Gaussian mixture models for probabilistic classification of breast cancer," *Cancer Res.*, vol. 79, no. 13, pp. 3492–3502, Jul. 2019.

[45] B. Panic, J. Klemenc, and M. Nagode, "Gaussian mixture model based classification revisited: Application to the bearing fault classification," *Strojniški vestnik J. Mech. Eng.*, vol. 66, no. 4, pp. 215–226, Apr. 2020.

[46] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Guang Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," 2019, *arXiv:1906.08172*.

[47] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2879–2886.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[49] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.

[50] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[51] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[52] K. Soomro and A. R. Zamir, "Action recognition in realistic sports videos," in *Computer Vision Sports*. Springer, 2015, pp. 181–208.

[53] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[54] M. Guo and Y. Du, "Classification of thyroid ultrasound standard plane images using ResNet-18 networks," in *Proc. IEEE 13th Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Oct. 2019, pp. 324–328.

[55] M. Gao, P. Song, F. Wang, J. Liu, A. Mandelis, and D. Qi, "A novel deep convolutional neural network based on ResNet-18 and transfer learning for detection of wood knot defects," *J. Sensors*, vol. 2021, no. 1, Jan. 2021, Art. no. 4428964.

[56] S. Ayyachamy, V. Alex, M. Khened, and G. Krishnamurthi, "Medical image retrieval using Resnet-18," in *Proc. SPIE*, vol. 10954, 2019, pp. 233–241.

[57] X. Ou, P. Yan, Y. Zhang, B. Tu, G. Zhang, J. Wu, and W. Li, "Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes," *IEEE Access*, vol. 7, pp. 108152–108160, 2019.

[58] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.

[59] S. Sun, W. Chen, L. Wang, X. Liu, and T. Liu, "On the depth of deep neural networks: A theoretical view," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 1–7.

[60] O. K. Oyedotun, A. El Rahman Shabayek, D. Aouada, and B. Ottersten, "Training very deep networks via residual learning with stochastic input shortcut connections," in *Proc. 24th Int. Conf. Neural Inf. Process. (ICONIP)*. Guangzhou, China: Springer, 2017, pp. 23–33.

[61] G. Li, J. Zhang, Y. Wang, C. Liu, M. Tan, Y. Lin, W. Zhang, J. Feng, and T. Zhang, "Residual distillation: Towards portable deep neural networks without shortcuts," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8935–8946.

[62] D. Gaur, J. Folz, and A. Dengel, "Training deep neural networks without batch normalization," 2020, *arXiv:2008.07970*.

[63] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*.

[64] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," 2020, *arXiv:2009.07485*.

[65] A. Hess, "Exploring feature reuse in DenseNet architectures," 2018, *arXiv:1806.01935*.

[66] K. Zhang, Y. Guo, X. Wang, J. Yuan, and Q. Ding, "Multiple feature reweight DenseNet for image classification," *IEEE Access*, vol. 7, pp. 9872–9880, 2019.

[67] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *Proc. 16th Eur. Conf. Comput. Vision (ECCV)*. Glasgow, U.K.: Springer, Aug. 2020, pp. 474–489.

[68] W. S. W. Samsudin and K. Sundaraj, "Image processing on facial paralysis for facial rehabilitation system: A review," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng.*, Nov. 2012, pp. 259–263.

[69] A. Abobakr, D. Nahavandi, M. Hossny, J. Iskander, M. Attia, S. Nahavandi, and M. Smets, "RGB-D ergonomic assessment system of adopted working postures," *Appl. Ergonom.*, vol. 80, pp. 75–88, Oct. 2019.

[70] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.

[71] M. S. Sundari and V. C. Jadala, "SqueezeNet fusion: Enhancing rhythmic heart disease classification through integrated pattern mining and deep learning," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 1–9, 2024.

[72] D. Chang, Y. Zheng, Z. Ma, R. Du, and K. Liang, "Fine-grained visual classification via simultaneously learning of multi-regional multi-grained features," 2021, *arXiv:2102.00367*.

[73] D. Haase and M. Amthor, "Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved MobileNets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14588–14597.

[74] C. Tu, J. Lee, Y. Chan, and C. Chen, "Pruning depthwise separable convolutions for mobilenet compression," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[75] A. Pramanik, S. Sarkar, and J. Maiti, "Oil spill detection using image processing technique: An occupational safety perspective of a steel plant," in *Proc. Emerg. Technol. Data Min. Inf. Secur. (IEMIS)*. Singapore: Springer, 2019, pp. 247–257.

[76] H. Kim, B. Elhamim, H. Jeong, C. Kim, and H. Kim, "On-site safety management using image processing and fuzzy inference," in *Proc. Comput. Civil Build. Eng.*, 2014, pp. 1013–1020.

[77] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, Jan. 2021.

[78] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2353–2362.

[79] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[80] H. Duan, K.-Y. Lin, S. Jin, W. Liu, C. Qian, and W. Ouyang, "TRB: A novel triplet representation for understanding 2D human body," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9478–9487.

[81] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *ACM Comput. Surv.*, vol. 56, no. 1, pp. 1–37, 2023.

[82] G. Sánchez-Brizuela, A. Cisnal, E. de la Fuente-López, J.-C. Fraile, and J. Pérez-Turiel, "Lightweight real-time hand segmentation leveraging MediaPipe landmark detection," *Virtual Reality*, vol. 27, no. 4, pp. 3125–3132, Dec. 2023.

[83] R. A. Güler and I. Kokkinos, "HoloPose: Holistic 3D human reconstruction in-the-wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10876–10886.

[84] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proc. Comput. Vis. Workshops (ECCV)*. Zurich, Switzerland: Springer, 2015, pp. 474–490.

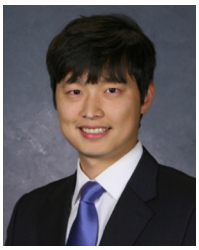
[85] W. Yin, H. Yin, D. Kragic, and M. Björkman, "Graph-based normalizing flow for human motion generation and reconstruction," in *Proc. 30th IEEE Int. Conf. Robot Human Interact. Commun. (RO-MAN)*, Aug. 2021, pp. 641–648.

[86] A. Raza, A. M. Qadri, I. Akhtar, N. A. Samee, and M. Alabdulhafith, "LogRF: An approach to human pose estimation using skeleton landmarks for physiotherapy fitness exercise correction," *IEEE Access*, vol. 11, pp. 107930–107939, 2023.

[87] G. Cai, C. Zhang, J. Xie, J. Pan, C. Li, and Y. Wu, "End-to-end 3D human pose estimation network with multi-layer feature fusion," *IEEE Access*, vol. 12, pp. 89124–89134, 2024.

[88] K. Gong, J. Zhang, and J. Feng, "PoseAug: A differentiable pose augmentation framework for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8575–8584.
[89] M. Ahmad and S.-W. Lee, "Human action recognition using multi-view image sequences features," in *Proc. 7th Int. Conf. Autom. Face Gesture Recognit. (FGR)*, 2006, pp. 523–528.
[90] W. Li, Z. Zhang, and Z. Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1499–1510, Nov. 2008.
[91] C. Yan, "Unsupervised posture modeling based on spatial–temporal movement features," in *Proc. Int. Conf. Electron. Commer., Web Appl., Commun.* Guangzhou, China: Springer, 2011, pp. 426–431.
[92] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, and J. Luo, "Anatomy-aware 3D human pose estimation with bone-based pose decomposition," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 32, no. 1, pp. 198–209, Jan. 2022.
[93] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3D human pose estimation with spatial and temporal transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11636–11645.
[94] H. Yang, L. Guo, Y. Zhang, and X. Wu, "U-shaped spatial–temporal transformer network for 3D human pose estimation," *Mach. Vis. Appl.*, vol. 33, no. 6, p. 82, Nov. 2022.
[95] X. Li, Y. Guo, W. Pan, H. Liu, and B. Xu, "Human pose estimation based on lightweight multi-scale coordinate attention," *Appl. Sci.*, vol. 13, no. 6, p. 3614, Mar. 2023.
[96] H. Ji, H. Deng, Y. Dai, and H. Li, "Unsupervised 3D pose estimation with non-rigid structure-from-motion modeling," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 3302–3311.
[97] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, 2006.
[98] G. J. McLachlan and D. Peel, *Finite Mixture Models*, vol. 299. Hoboken, NJ, USA: Wiley, 2000.
[99] T. Elguebaly and N. Bouguila, "Bayesian learning of finite generalized Gaussian mixture models on images," *Signal Process.*, vol. 91, no. 4, pp. 801–820, Apr. 2011.
[100] R. E. Kass and A. E. Raftery, "Bayes factors," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, pp. 773–795, 1995.
[101] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[102] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[103] W. Cai, "A manifold learning framework for both clustering and classification," *Knowledge-Based Syst.*, vol. 89, pp. 641–653, Nov. 2015.
[104] F. Zhang, S. Han, H. Gao, and T. Wang, "A Gaussian mixture based hidden Markov model for motion recognition with 3D vision device," *Comput. Electr. Eng.*, vol. 83, May 2020, Art. no. 106603.
[105] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B, Stat. Methodology*, vol. 39, no. 1, pp. 1–22, Sep. 1977.
[106] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, Mar. 1978.
[107] C. Gu, H. Xie, X. Lu, and C. Zhang, "CGMVAE: Coupling GMM prior and GMM estimator for unsupervised clustering and disentanglement," *IEEE Access*, vol. 9, pp. 65140–65149, 2021.
[108] Y. Wang, Y. Pang, O. Chen, H. N. Iyer, P. Dutta, P. K. Menon, and Y. Liu, "Uncertainty quantification and reduction in aircraft trajectory prediction using Bayesian-entropy information fusion," *Rel. Eng. Syst. Saf.*, vol. 212, Aug. 2021, Art. no. 107650.
[109] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1344–1348, Aug. 2005.
[110] M. Karim, S. Khalid, A. Aleryani, N. Tairan, Z. Ali, and F. Ali, "HADE: Exploiting human action recognition through fine-tuned deep learning methods," *IEEE Access*, vol. 12, pp. 42769–42790, 2024.
[111] J. Kishtwaria, P. Mathur, and A. Rana, "Ergonomic evaluation of kitchen work with reference to space designing," *J. Human Ecology*, vol. 21, no. 1, pp. 43–46, Jan. 2007.
[112] O. Janusz, "Evaluation of modern day kitchen knives: An ergonomic and biomechanical approach to design," Master's thesis, Dept. Ind. Manuf. Syst. Eng., Iowa State Univ., Ames, IA, USA, 2016.
[113] H. Iyer, J. Reynolds, C. S. Nam, and H. Jeong, "Pathfinder networks: Evaluating injury and safety using restaurant Workers' mental models," in *Proc. Human Factors Ergonom. Soc. Annu. Meeting*. Los Angeles, CA, USA. SAGE, Aug. 2024, Paper 10711813241262034.
[114] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[115] J. Needhi, "Data-driven approach to automated lyric generation," *Int. J. Eng. Comput. Sci.*, vol. 13, no. 7, pp. 26285–26290, Jul. 2024.
[116] H. Yang, K. Pasupa, A. C. S. Leung, J. T. Kwok, J. H. Chan, and I. King, "Neural information processing," in *Proc. 27th Int. Conf. (ICONIP)*, vol. 12533. Springer, 2020, p. 365.
[117] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[118] H. Iyer, M. Gandhi, and S. Nair, "Sentiment analysis for visuals using natural language processing," *Int. J. Comput. Appl.*, vol. 128, no. 6, pp. 31–35, Oct. 2015.
[119] M. El-Assal, P. Tirilly, and I. M. Bilasco, "Spiking two-stream methods with unsupervised STDP-based learning for action recognition," 2023, *arXiv:2306.13783*.
[120] A. Sanakoyeu, M. A. Bautista, and B. Ommer, "Deep unsupervised learning of visual similarities," *Pattern Recognit.*, vol. 78, pp. 331–343, Jun. 2018.
[121] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3361–3368.
[122] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, and S. Yan, "DL-SFA: deeply-learned slow feature analysis for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2625–2632.
[123] H. Iyer and H. Jeong. (2024). *PE-USGC: Posture Estimation-based Unsupervised Spatial Gaussian Clustering for Supervised Classification of Near-duplicate Human Motion*. [Online]. Available: https://github.com/iyer1729/PE-USGC
[124] D. M. Jennewein et al., "The Sol supercomputer at Arizona state university," in *Proc. Pract. Express Adv. Res. Comput.* New York, NY, USA: Association for Computing Machinery, Jul. 2023, pp. 296–301.

HARI IYER received the M.S. degree in software engineering from Arizona State University, in 2018, where he is currently pursuing the Ph.D. degree with the Human-in-Mind Engineering Research (HiMER) Lab. He was with Optimal Synthesis Inc., as a Research Engineer, with a focus on flight navigation and simulation software delivered to the NASA Ames Research Center and the Missile Defense Agency (U.S. Department of Defense). He was a Visiting Researcher with Indian Institute of Technology Bombay and an Engineering Intern with App Orchid Inc. He is a Graduate Research Associate with HiMER, Arizona State University. He was a recipient of the 2024–2025 Harold and Lucille Dunn Engineering Scholarship. He also was named the 2024 Student of the Year by the Institute of Industrial and Systems Engineers' Applied Ergonomics Society.



HEEJIN JEONG (Member, IEEE) received the Ph.D. degree in industrial and operations engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018. He was an Assistant Professor with the Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, IL, USA, from 2019 to 2022. He is currently an Assistant Professor with The Polytechnic School, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, AZ, USA, where he is also a Biomedical Engineering Graduate Faculty with the School of Biological and Health Systems Engineering. His research interests include extended reality systems for occupational safety enhancement and healthcare rehabilitation training, and human–robot collaboration in Industry 5.0 manufacturing systems.