AI Vision to Care: A QuadView of Deep Learning for Detecting Harmful Stimming in Autism

By

Nyamtulla Shaik

M.S Computer Science, University of Kansas, 2025
B. Tech Computer Science, Jawaharlal Nehru Technological University, 2020

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science In Computer Science.

Chair: Dr. Sumaiya Shomaji

Dr. Bo Luo

Dr. Dongjie Wang

Date Defended: May 6, 2025

The thesis committee for Nyamtulla Shaik certifies that this is the approved version of the following thesis:

AI Vision to Care: A QuadView of Deep Learning for Detecting Harmful Stimming in Autism

Chair: Dr. Sumaiya Shomaji

Date Accepted: May 6, 2025

Abstract

Stimming refers to repetitive actions or behaviors used to regulate sensory input or express feelings. Children with developmental disorders like autism (ASD) frequently perform stimming. This includes arm flapping, head banging, finger flicking, spinning, etc. This is exhibited by 80-90% of children with Autism, which is seen in 1 among 32 children in the US. Head banging is one of these self-stimulatory habits that can be harmful. If these behaviors are automatically identified and notified using live video monitoring, parents and other caregivers can better watch over and assist children with ASD.

Classifying these actions is important to recognize harmful stimming, so this study focuses on developing a deep learning-based approach for stimming action recognition. We implemented and evaluated four models leveraging three deep learning architectures based on Convolutional Neural Networks (CNNs), Autoencoders, and Vision Transformers. For the first time in this area, we use skeletal joints extracted from video sequences. Previous works relied solely on raw RGB videos, which are vulnerable to lighting and environmental changes. This research explores Deep Learning based skeletal action recognition and data processing techniques for a small unstructured dataset that consists of 92 home-recorded videos collected from publicly available sources like YouTube. Our robust data cleaning and pre-processing techniques helped the integration of skeletal data in stimming action recognition, which performed better than existing works with a classification accuracy of up to 87%.

In addition to using traditional deep learning models like CNNs for action recognition, this study is among the first to apply data-hungry models like Vision Transformers (ViTs) and Autoencoders for stimming action recognition on the dataset by leveraging data augmentation

techniques. This research advances the development of automated systems that can assist caregivers in more efficiently tracking stimming activities.

Acknowledgments

Completing this thesis has been a significant milestone in my academic journey, and I am grateful to everyone who supported me through it.

Firstly, I want to express my heartfelt gratitude to my advisor, Dr. Sumaiya Shomaji. Her constant encouragement, patience, and guidance played a significant role in helping me complete this work. From our first meeting, she was always there to offer thoughtful advice and keep me on track. I learned much under her mentorship and truly appreciate everything she has done for me.

I am also very thankful to Dr. Bo Luo, my PhD advisor, for being understanding and supportive while I completed this master's research. His thoughtful insights and continued support have been invaluable, and I am genuinely thankful for the academic freedom and trust he offered me.

I would also like to thank Dr. Dongjie Wang for being on my committee and for the helpful feedback and suggestions that improved the quality of this thesis.

I want to acknowledge Dr. Tamzidul Hoque for being a kind, encouraging presence and friendly support.

I want to give my teammate, Sailesh Pandey, a special shoutout for his contributions to the data collection phase. His efforts laid the groundwork for key components of this study.

To my friends, thank you for always being there. Your encouragement, thoughtful check-ins, and the moments of laughter we shared made this journey lighter and more meaningful. I appreciate your support through all the ups and downs.

Lastly, and most importantly, I want to thank my family. Your love, support, and trust in my decision to pursue higher studies gave me the strength to see this through.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Stimming

A self-stimulatory behavior known as "stimming" involves repeated stereotyped sounds, movements, or actions used by individuals suffering from autism spectrum disorder (ASD) or other developmental disorders to control sensory input, show enthusiasm, or manage stress. Individuals may exhibit various stimming behaviors, such as hand flapping, back-and-forth rocking, spinning, repeating words, or vocalizations.

Types of Stimming

Stimming behaviors can be classified into non-harmful and harmful (Bishop et al., 2006). Some of the stimming behaviors, like Hand-flapping, finger flicking, rocking, spinning objects, humming, echolalia (repeating words or phrases), and pacing, are generally harmless and serve as self-soothing or sensory regulation methods for individuals with ASD. Whereas other stimming behaviors like Head-banging, self-biting, skin-picking, hair-pulling, and hitting oneself may cause physical harm or injury to the individual and require intervention and support to address.

Likelihood of Stimming

Individuals with autism spectrum disorder are more likely to exhibit stimming behaviors compared to neurotypical individuals. However, stimming can also be observed in individuals with other developmental or sensory processing disorders like. (DeWeerdt, 2023) obsessive-compulsive disorder (OCD), Intellectual Disability (ID), Sensory Processing Disorder (SPD), Anxiety Disorders, Tourette Syndrome, Schizophrenia Spectrum Disorders.

Statistics on Autism Prevalence in the US

According to Maenner et al. (2023), the prevalence of autism spectrum disorder (ASD) among children in the United States was estimated to be approximately 1 in 54 in 2016 and increased to 1 in 32 by 2022. This represents an increase from previous estimates, indicating a rising trend in reported prevalence.

Percentage of Individuals with Stimming

Stimming behaviors are a common characteristic of autism spectrum disorder. According to the study by Leekam et al. (2007) published in the Journal of Autism and Developmental Disorders, up to 80-90% of individuals with autism engage in stimming behaviors to some extent.

Age of Diagnosis

The age at which autism is diagnosed can vary, but a study by Zuckerman et al. (2015) suggests that most children are diagnosed with ASD by age 4. Early intervention and diagnosis are critical for improving outcomes and providing support for individuals with autism and their families.

Impact of ASD on Families

ASD can have a significant impact on families, including emotional, financial, and logistical challenges. According to the study by Buescher et al. (2014), families of children with ASD incur an average of $17,000 more in annual medical and non-medical expenditures compared to families of children without ASD.

Traditionally, monitoring stimming behaviors in children with autism spectrum disorder (ASD) relies heavily on continuous human observation. Parents, caregivers, or teachers must closely watch children to recognize harmful behaviors and intervene appropriately. However,

constant supervision is not always feasible due to time, resources, and attention constraints, especially in home or school environments where multiple responsibilities compete for attention. The reliance on manual observation also introduces the risk of delayed intervention or missed incidents, potentially leading to injury or emotional distress. These limitations highlight the need for automated systems that can reliably detect harmful stimming actions in real time.

Objective

The primary objective of this research is to develop deep learning-based classification models capable of recognizing stimming actions in children with autism spectrum disorder (ASD) from video sequences. Accurately distinguishing between different stimming behaviors, particularly identifying harmful actions such as head banging, is critical for enabling timely intervention. By classifying these behaviors, the models provide a foundational step toward building systems that can assist parents and caregivers in monitoring children more effectively. This research focuses on developing and evaluating classification models, laying the groundwork for future real-time video monitoring solutions that can alert caregivers to potentially harmful activities and promote safer environments for children with ASD.

Figure 1

The High-level Workflow of the system

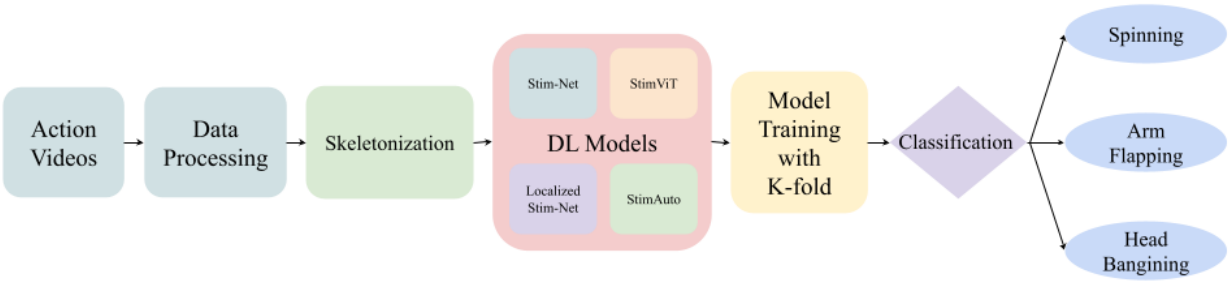


Figure 1 provides an overview of the workflow developed for stimming action recognition. The process begins with collecting and preparing raw video data, followed by a series of preprocessing steps designed to enhance data quality and extract meaningful features. In particular, skeletal information is derived from videos to capture key body movements relevant to stimming behaviors. This skeletal data is then used to train deep learning models to classify different types of stimming actions. The overall workflow is designed to enable an efficient and accurate recognition process, forming the basis for future systems that can assist in monitoring and supporting children with autism spectrum disorder (ASD).

The rest of the document contains a detailed view of each step, from data processing to analysis of results. We also review the relevant literature and compare and contrast our work with the literature and current state-of-the-art models.

Chapter 2 Literature Review

In this literature review, we explore prior research on video and some non-video-based recognition of self-stimulatory behaviors (stimming) in children with autism spectrum disorder (ASD). We examine the evolution of datasets, models, and techniques used in this domain, identifying the key technological advancements that have been made for identifying and classifying the stimming actions. We also highlight significant challenges such as dataset limitations, camera motion artifacts, and variations in video quality. The discussion focuses on how various deep learning architectures, particularly convolutional neural networks (CNNs) and temporal models like LSTMs, have been applied to improve accuracy in detecting repetitive behaviors such as hand flapping, headbanging, and spinning.

Early Work on Video-Based ASD Behavior Recognition

The earliest efforts in the automated recognition of stimming behaviors relied on traditional machine learning approaches with handcrafted feature extraction techniques. One of the most significant contributions in this domain was the Self-Stimulatory Behavior Dataset (SSBD) introduced by Rajagopalan et al. (2013). The dataset was developed to facilitate the automated hand flapping, headbanging, and spinning classification using unstructured home video recordings of children with ASD. Their study employed Space-Time Interest Points (STIP), a feature extraction technique commonly used for action recognition. The classification was performed with Support Vector Machines (SVMs). The dataset included 75 home videos that clinical specialists annotated.

Their study applied preprocessing techniques such as frame extraction, background subtraction, and motion trajectory smoothing to reduce noise and enhance motion patterns to improve feature consistency. Despite these refinements, the model achieved only 50.7%

classification accuracy using five-fold cross-validation. The research found that conventional machine learning methods, which depended on features designed by hand, were inadequate for identifying self-stimulatory behaviors because of the intricate and diverse characteristics of real-world video data.

Deep Learning-Based Stimming Behavior Recognition

The recent advancements in deep learning enabled researchers to move beyond handcrafted features, adopting CNNs and hybrid models to improve classification accuracy. Negin et al. (2021) developed a deep learning-based framework for recognizing stimming behaviors using real-world video recordings. Their method used YOLOv3 to detect children in video frames, followed by a Histogram of Optical Flow (HOF) to extract motion-based features. The extracted features were encoded using Bag-of-Visual-Words (BoVW) and classified using a Multi-Layer Perceptron (MLP). The dataset for this study was collected from ASD diagnostic centers, though specific details were not publicly disclosed. Their model achieved 78% accuracy, surpassing the earlier SVM-based methods. However, occlusions and lighting variability were identified as factors that reduced classification performance.

Washington et al. (2021) advanced deep learning techniques by creating a system designed to identify headbanging behaviors through a CNN-LSTM hybrid model. In this methodology, CNNs were used to extract spatial characteristics from individual frames, and LSTMs were utilized to capture temporal relationships, which enabled the system to detect temporal recurring patterns. The model was trained and evaluated on CNN-LSTM on the same SSBD dataset, employing a three-fold cross-validation approach to ensure dependable classification. It achieved an F1-score of 90.77% on detecting headbanging, demonstrating that the combination of temporal modeling and CNN-based feature extraction significantly improves

recognition accuracy. The study also highlighted the importance of pose estimation in reducing the impact of background motion artifacts in practical video data.

Tariq et al. (2018) explored mobile-based applications for ASD behavior detection by employing CNN-based feature extraction and an MLP classifier. Their model incorporated background subtraction and facial alignment to eliminate environmental distractions and improve feature quality. The dataset used in this study included 162 home videos of autistic and neurotypical children aged 2-4 years, enabling the model to learn behavioral differences across a diverse range of subjects. The system achieved an 85% classification accuracy, validating the feasibility of real-time ASD screening using mobile devices. While it was good research on classifying autism or no autism behavior, it is different from our work, as we classify stimming actions with a prior assumption of autistic behavior.

Temporal Convolutional Networks (TCNs) for Stimming Detection

Recent research has increasingly recognized the limitations of recurrent networks in long-term sequence modeling. As a result, recent research has increasingly adopted temporal convolutional networks (TCNs) for video-based stimming classification. One of the significant studies in this area was conducted by Wei et al. (2023). The study aimed to improve classification robustness by addressing dataset noise, background distractions, and feature extraction challenges.

To enhance dataset quality, the authors curated an improved version of SSBD, removing 11 noisy samples and incorporating 12 new subjects. This resulted in a final dataset containing 61 videos, which were further segmented into 168 short clips. Their model leveraged Inflated 3D ConvNets (I3D) with Multi-Stage Temporal Convolutional Networks (MS-TCNs), allowing for the simultaneous extraction of spatial and temporal features. The system achieved a Weighted

F1-score of 0.83, significantly outperforming earlier deep learning-based methods. The study also introduced ESNet, a lightweight deep learning model optimized for real-time deployment on embedded systems. The ESNet-based model achieved a Weighted F1-score of 0.71, demonstrating that computationally efficient architectures can achieve competitive accuracy while reducing processing overhead. Their results confirmed that MS-TCNs outperformed LSTM-based models, reinforcing the advantages of convolutional temporal models for recognizing repetitive behaviors in children with ASD.

Recent Advances in Transformer-Based and Multi-Branch Architectures

With the increasing complexity of video-based ASD classification, recent studies have explored Transformer-based models and dual-branch architectures to improve performance. Serna-Aguilera et al. (2024) proposed a dual-branch CNN model integrating facial expression recognition (FER) with temporal modeling to improve classification accuracy. The study was conducted on video datasets from the University of Arkansas (UARK) and the University of Texas at San Antonio (UTSA). Employing a five-fold cross-validation approach, the model attained an accuracy of 81.48% in distinguishing children with ASD and those without ASD, illustrating the effectiveness of merging facial expression analysis with spatial and temporal assessments for classifying ASD behaviors.

Skeletal-based action recognition

One key influence in selecting a skeleton-based method for stimming action recognition is the Double-Feature Double-Motion Network (DD-Net) proposed by Yang et al. (2020). This work is particularly noteworthy for balancing computational efficiency and robust action recognition, essential in systems intended for real-time analysis or large-video data processing.

A significant advantage of DD-Net lies in its strategy for handling viewpoint and motion speed variations. Specifically, the authors introduce Joint Collection Distances (JCD), a feature representation that uses the pairwise distances between skeletal joints. Unlike raw 3D coordinates, which can vary significantly with rotation and camera placement, JCD is relatively invariant to viewpoint changes, providing a more stable feature space for classifying skeletal movements. This property is especially relevant for recognizing subtle or varied stimming behaviors, where subjects may move unpredictably and do not necessarily stay aligned with a fixed camera perspective.

In addition, DD-Net incorporates both “slow” and “fast” motion representations, capturing different temporal scales of movement. This multi-scale approach enables the model to recognize actions that can differ in speed, an attribute also reflected in stimming behaviors, which may alternate between small, repetitive gestures and more pronounced movements.

Beyond feature engineering, DD-Net employs one-dimensional convolutions to process skeletal time-series data. This design choice allows for parallelization across the temporal dimension. It leads to very high inference speeds, often in the range of thousands of frames per second on standard GPU hardware. Such computational efficiency is beneficial for real-time applications or scenarios requiring rapid feedback, a key consideration for interventions or assistive technologies in autism care.

Drawing on these strengths, the principles established by DD-Net have been adapted to the stimming action recognition domain. Specifically, viewpoint-invariant geometric features, multi-scale motion cues, and efficient convolution-based temporal modeling align well with the challenges of identifying and categorizing stimming behaviors in video data.

Summary of literature studied

Table 1 summarizes major studies relevant to stimming action recognition and autism spectrum disorder (ASD) detection using video data. These works vary in their objectives, target classes, datasets, and reported performance.

Table 1

Literature review comparison

Name of Paper	Year	Objective	Accuracy / F1 Score	Dataset	Classes
Rajagopalan et al.	2013	Classify Stimming Actions	50.70%	SSBD	3
Zamzmi et al.	2019	Differentiate pain vs. stimming	84.20%	Infant pain behavior datasets	2 (Pain/Stimming)
Tariq et al.	2020	ASD detection	85%	162 home videos (2–4 years old)	2 (ASD/No ASD)
Negin et al.	2021	Classify Stimming Actions	78%	Private ASD diagnostic center dataset	3
Washington et al.	2021	Headbanging detection	F1-score: 90.77%	SSBD	2
Ali et al.	2022	Classify Stimming Actions	86.04% (Activis), 75.62% (SSBD)	Activis, SSBD	3
Wei et al.	2023	Classify Stimming Actions	F1-score: 83%	SSBD	3
Serna-Aguilera et al.	2024	ASD detection	81.48%	UARK, UTSA private datasets	2 (ASD/No ASD)

Challenges

Limited, Real-World Data

The availability of suitable video datasets for autism stimming actions is minimal. The only public dataset available is the SSBD dataset used in this research. Gathering video clips of autism stimming behaviors is not as simple as recording in a controlled lab setting. Families tend

to capture moments on their phones, often in less-than-ideal conditions. Because of that, each clip can have its quirks, like different resolutions, random angles, and cluttered backgrounds, making it harder for standard pose-estimation and learning methods to work effectively. Plus, the overall number of subjects and videos remains relatively small, meaning there is not as much variety for the model to learn from.

Moving Camera Angles, Noise, and Occlusions

In the videos of SSBD, children can move out of view, or someone might walk in front of the camera at the worst possible moment, obscuring the action. On top of that, handheld devices introduce the extra shake. All these issues create inconsistent or missing data, complicating efforts to pinpoint what is happening in the clip.

Differentiating Stimming from Non-Stimming Actions

Stimming behaviors can be subtle and sometimes look like ordinary movements. Whether it is fluttering hands that might be confused with a playful wave or rocking that seems like a dance move, it is easy for a model to get tripped up. Separating these repetitive actions from everyday gestures becomes a significant hurdle when everything appears in the same video context.

Multi-Class Stimming Classification

Many studies focus on a single stimming action, like headbanging, but kids often switch between repetitive movements. Handling four distinct classes with a limited number of subjects will add complexity to both the dataset and the models needed. However, it is crucial for a more realistic, all-inclusive approach.

Reliability of Skeletal Coordinates in Non-Professional Footage

Pose estimation tools require clear images, consistent lighting, and unobstructed perspectives. Home videos often do not provide these conditions. Misaligned joints, low-confidence key points, absent frames, and out-of-focus subjects to the extent that the model mistakes the shadow of a human for generating the skeleton. These can quickly hinder the model's comprehension of movement. Developing a method to handle these imperfect skeletal coordinates is crucial for ensuring the system operates effectively in typical settings.

Contributions

End-to-End Pipeline for Multi-Class Stimming Recognition

This project outlines a seamless process from segmenting raw videos to the relevant stimming clips, extracting skeletal features, and finally classifying multiple stimming behaviors in one unified pipeline. It moves beyond single-action detection, making it more fitting for the variations in children's repetitive movements.

Handling Real-World, Unstructured Videos

Rather than relying on clean, professionally recorded footage, this approach is designed to work with the unpredictable nature of handheld recordings. Variations in camera angles, shifting backgrounds, and occasional obstructions are all accounted for in the preprocessing pipeline. From precisely segmenting relevant video clips to extracting skeletal data in a way that remains stable despite visual inconsistencies, the system is built to function effectively even under imperfect conditions.

Focus on Skeletal-Based Deep Learning

The approach highlights the most essential aspect, the motion itself, by transforming raw video frames into sequences of joints. This skeletal focus lessens the chance that the model will be thrown off by lighting changes, background objects, or the child's clothing. Various

augmentation techniques along the temporal dimension add further resilience to movement speed and duration differences.

Application of CNN and Autoencoders in ASD Research

While CNNs and autoencoders are recognized tools in computer vision, applying them to detect subtle stimming behaviors presents unique challenges and insights. This study shows how well these architectures adapt to the overlapping, repetitive motions typical of children's stimming, potentially guiding future work in similar contexts.

Broader Clinical and Research Implications

By systematically tackling multiple stimming actions, this work moves closer to a tool that could one day help caregivers, therapists, or researchers. Whether for real-time monitoring, early intervention, or more profound behavioral studies, a reliable, skeletal-based detection framework is promising for improving how we understand and respond to stimming in everyday life.

Background

This section provides an overview of the key techniques used in this study, including Convolutional Neural Networks (CNNs), skeletal models for action recognition, autoencoders, and vision transformers. These models are critical in classifying stimming behaviors from video sequences, enabling the system to extract, represent, and learn meaningful motion features.

Convolutional Neural Networks (CNNs) for Action Recognition

Convolutional Neural Networks (CNNs) are deep learning architecture specifically designed to process structured grid-based data, such as images and videos. Unlike traditional, fully connected neural networks, CNNs use convolutional layers to automatically learn hierarchical feature representations, enabling them to detect edges, textures, and more complex

patterns without requiring handcrafted feature extraction. These properties make CNNs highly effective for human action recognition tasks, including movement classification and behavioral analysis (LeCun, Bengio, & Hinton, 2015).

A typical CNN consists of multiple layers that progressively extract complex features from raw input. Convolutional layers apply learnable filters to detect spatial features at different levels of abstraction. Pooling layers down sample feature maps to reduce computational cost while retaining the most salient information. Fully connected layers map the extracted features to class probabilities, enabling the model to distinguish between different actions.

CNNs have demonstrated state-of-the-art performance in action recognition when applied to skeletal data. Instead of processing raw video frames, CNN-based approaches can model skeletal sequences by analyzing spatial relationships between key joint positions over time (Ke et al., 2017). This research applies CNNs to skeletal joint sequences to extract spatial-temporal features from stimming behaviors, facilitating robust classification across multiple stimming actions.

Skeletal Models for Human Action Recognition

Skeletal models provide an alternative representation of human motion by encoding body dynamics through key joint positions rather than relying on raw pixel-based representations. These models have been widely adopted in action recognition due to their ability to capture motion trajectories while reducing sensitivity to background noise and appearance variations (Yan et al., 2018).

Pose estimation frameworks such as MediaPipe Pose (Lugaresi et al., 2019) generate skeletal key points that represent major body joints, including the head, shoulders, arms, and

legs. Each frame in a video is processed to extract these joint coordinates, which are then aggregated over time to form a temporal sequence of body movement.

The advantages of skeletal representations for action recognition

Viewpoint and lighting invariance: Unlike RGB-based methods, skeletal models focus on body motion rather than appearance, making them more robust to camera angles and lighting conditions.

Background noise reduction: Skeletal models eliminate distractions caused by background objects, clothing variations, or environmental clutter by analyzing joint movements instead of pixel intensity.

Efficient data representation: Skeletal data consists of a limited set of numerical coordinates, reducing computational complexity compared to full-frame video processing.

This study uses skeletal representations to track and classify repetitive stimming behaviors, such as hand flapping and headbanging. By focusing on joint trajectories, the model becomes more resilient to real-world challenges, including camera shake, occlusions, and inconsistent recording conditions.

Autoencoders for Feature Learning and Noise Reduction

Autoencoders are deep learning models designed for unsupervised feature learning and dimensionality reduction. They consist of two primary components:

Encoder: Compresses high-dimensional input data into a lower-dimensional latent space, capturing the most relevant features.

Decoder: Reconstructs the original input from the latent representation, ensuring that the encoded features retain essential information.

Autoencoders are particularly useful in scenarios where data contains noise or redundancy. Using autoencoders in action recognition helps with better feature extraction and dealing with noisy data.

Denoising skeletal sequences: The model learns to filter out inconsistencies caused by pose estimation errors or background interference by training an autoencoder to reconstruct clean versions of noisy input sequences.

Extracting latent motion features: The encoder compresses high-dimensional skeletal data into a structured form, improving the ability of classification models to distinguish between different action categories.

This research employs autoencoders to enhance the representation of stimming behaviors by filtering out pose estimation artifacts and emphasizing key motion patterns. By integrating autoencoders into the classification pipeline, the model achieves a more robust and generalizable understanding of stimming movements, leading to improved classification performance across real-world video datasets.

Vision Transformers

Dosovitskiy et al. (2021) introduced the Vision Transformer (ViT), which is derived from the transformer model, which uses a self-attention mechanism to retain long temporal relations, unlike traditional convolutional networks, which focus mainly on localized spatial dependencies. ViTs utilize self-attention mechanisms to analyze long-range dependencies across entire sequences of poses. The model divides the image into tokens by splitting the image into fixed-size non-overlapping patches, which are converted to embeddings and passed to the transformer encoder for further processing. This capability to process global context allows them to capture complex motion patterns effectively.

The transformer encoder consists of a multi-head self-attention (MHSA), which is a mechanism that computes relationships between all patches of the image by generating query, key, and value (QKV) representations; they will help determine how much attention each patch should pay to others. This is followed by feed-forward networks (FFNs) that refine the learned representations, along with the FFN layer normalization and residual connections that ensure stable gradient flow during training. Unlike traditional CNNs that extract local features through convolutional kernels, self-attention enables ViTs to capture long-range dependencies, allowing the model to understand contextual relationships between distant parts of an image, which can effectively recognize structural patterns of the image.

ViTs have shown impressive performance when trained on massive datasets. However, they also have some limitations, such as high computational costs due to self-attention calculations having quadratic complexity, making them less efficient on smaller datasets and resource-constrained environments (Wu et al., 2021)

Chapter 3: Dataset and Preprocessing

Dataset

In this research, we use the Self-Stimulatory Behavior Dataset (SSBD) introduced by Rajagopalan et al. (2013) as a base dataset. It is a collection of YouTube videos recording the stimming actions observed in children with autism spectrum disorder (ASD). These behaviors involve videos of flapping hands, spinning, or banging their heads. The SSBD dataset is the only publicly available stimming action video dataset that can recognize these behaviors automatically, making it a crucial resource for this study.

Since the videos are filmed in real-life situations, they face various challenges, such as poor lighting, clumsy backgrounds, shaky cameras, and other people or objects occasionally blocking the view. These factors make the dataset challenging but also particularly valuable for developing behavior recognition tools that work well in realistic settings.

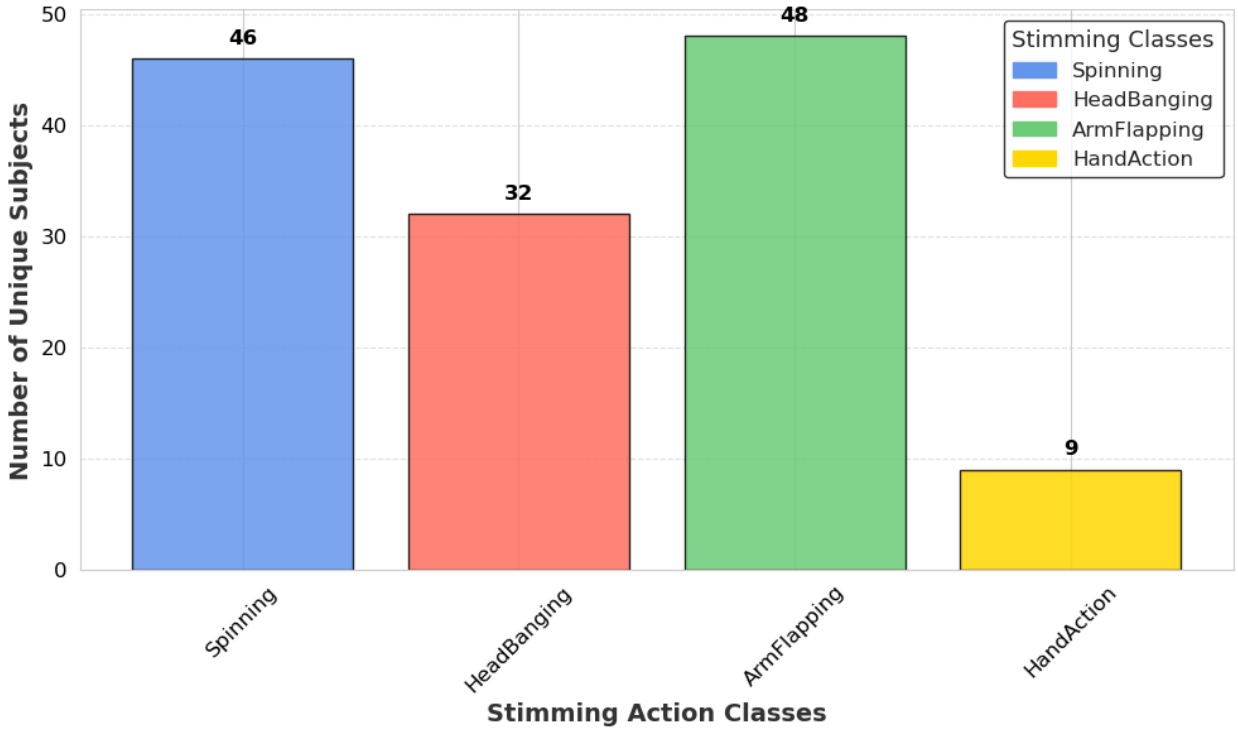
The videos are grouped into three main types of stimming behaviors: arm flapping, head banging, and spinning. Arm flapping involves quick, repeated movements, often in response to excitement or stress. Headbanging is when a child repeatedly hits their head against a surface, which can sometimes be harmful. Spinning refers to turning in circles, either by the child or involving objects like a spinning chair and is often considered a soothing activity for children with ASD.

As the dataset is not current, some YouTube videos are unavailable now; they may have been taken down or restricted access. A recent study by Wei et al. (2023), tried to curate more videos from YouTube and increase the dataset. We scanned YouTube with autism-related keywords and collected a few more videos, initially adding up to 135 videos in our extended dataset. We also introduced a new class called HandAction for finger-flicking stimming action.

As a result, we end up with 32 subjects for headbanging, 46 for spinning, 48 for arm flapping, and nine for hand action, as shown in **Figure 2.**

Figure 2

Distribution of Subjects Across Classes in the ASBD Dataset



Even though the SSBD dataset was introduced in the 2000s, no significant advancements have been made to address these challenges or develop a public dataset for stimming videos. Despite these challenges, the SSBD dataset is crucial for this research.

Video Preprocessing

The data processing pipeline starts with downloading videos from YouTube and data-wrangling steps to clean up and organize videos for further processing.

Video Downloading

The first stage of the dataset construction is collecting and structuring videos that exhibit stimming behaviors. After the downloading, the data undergoes careful structure to ensure

accurate results as the videos are extracted from publicly available platforms like YouTube. This data collection process is the foundation for analysis and model development.

The SSBD dataset comes with a spreadsheet that contains important details such as video URLs, availability status, and timestamps that indicate the segments of each video where stimming behaviors are observed. Using this metadata, the videos are systematically collected and prepared for the next stages of processing.

Automated Video Downloading: The URLs from metadata file are passed to video downloader script which ensures proper downloading and saving the video. This approach can only get accessible videos from YouTube.

Storage and Organization: Subsequently, acquired videos are systematically stored in an organized directory structure. The structure also links each video file to the relevant metadata entry for traceability along the data pipeline. Automating the data collection process and having error handling to tackle issues like faulty URLs, restricted access or damaged video files ensures the collected video data is correct and complete.

After downloading, each video undergoes a careful manual observation to check for the quality of the video and the action performed. Some videos are discarded at this stage, which are considered unusable as subjects are too small, have constant occlusions, or have poor resolution. This sets a good foundation for further data processing and model development.

Extraction of Action-Specific Video Clips

After raw videos are obtained, the next stage of the processing pipeline is video segmentation, which in our data processing is nothing but separating out video segments that show stimming behavior. This is important as complete videos tend to have irrelevant content, and this step narrows down the dataset to relevant actions. The actual segmentation process starts

with the metadata file, which provides start and end timestamps that define the time range within each video where stimming actions could be found. We strip the corresponding video files using these timestamps and extract solely the pertinent action segments. Each segment is then saved as a separate video file using a standard naming convention to maintain a subject ID so it can be easily found in the future. This segmentation reduces the amount of redundant or unnecessary data, so the resulting dataset is optimized for analysis and model training. As a result, we end up with 144 videos, as some videos have multiple action segments.

The resulting action segments vary in duration as the duration of each action varies by each subject, averaging around 15 seconds each. Having a constant number of frames in each input video clip is necessary to feed it to the model easily, so we have decided to further slice each action clip into smaller clips with a fixed number of frames. A size 72 is chosen, which ensures complete action is maintained in the video.

Manual Inspection and Cleaning of Videos

Each video was evaluated based on five qualitative criteria: the consistency between the observed action and its assigned label, the clarity and continuity of the action performance, the resolution quality of the video, the presence of multiple individuals within the frame, and the degree of camera stability, including any subject occlusions or shakiness. Videos that failed to

meet acceptable standards across these dimensions were excluded from the dataset. As a result of this filtering process, the dataset was reduced to include only 92 subjects.

Figure 3

The video quality ratings distribution

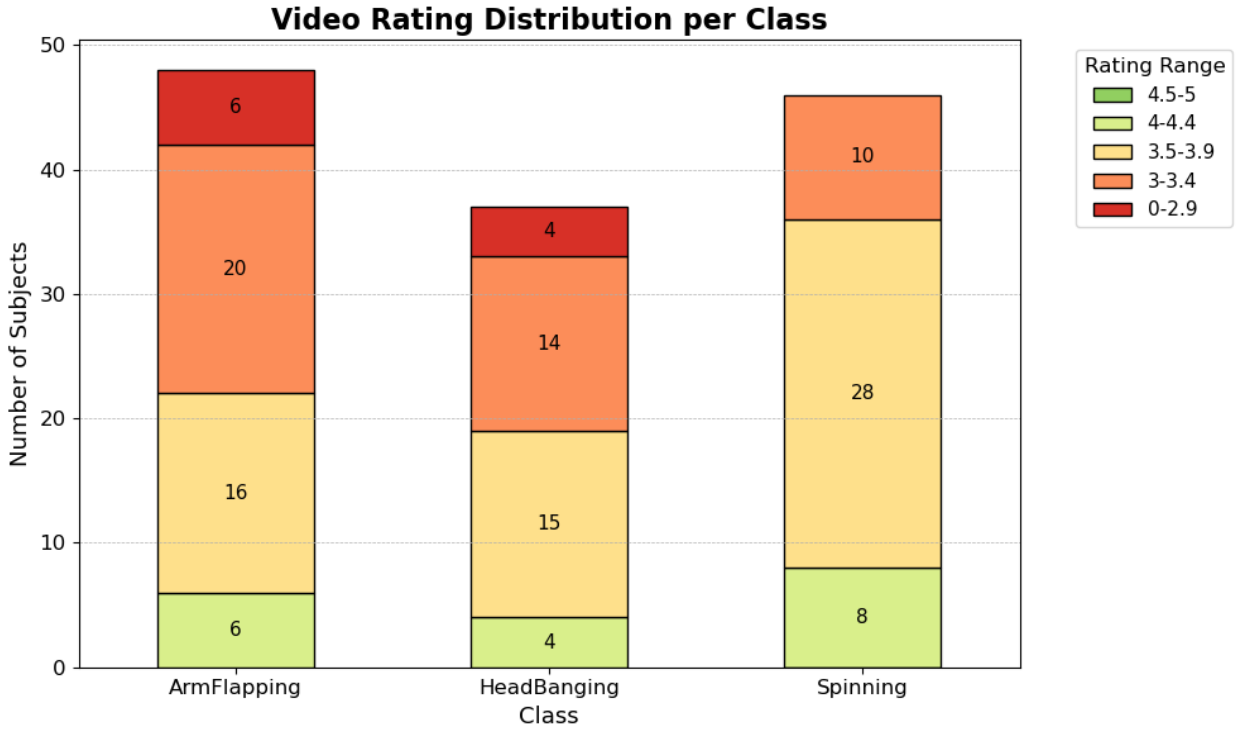


Figure 3 shows the ratings of videos in all the classes; the portion in red shows the videos with poor ratings. The top 30 videos are chosen for the final dataset to remove noisy videos and balance the dataset.

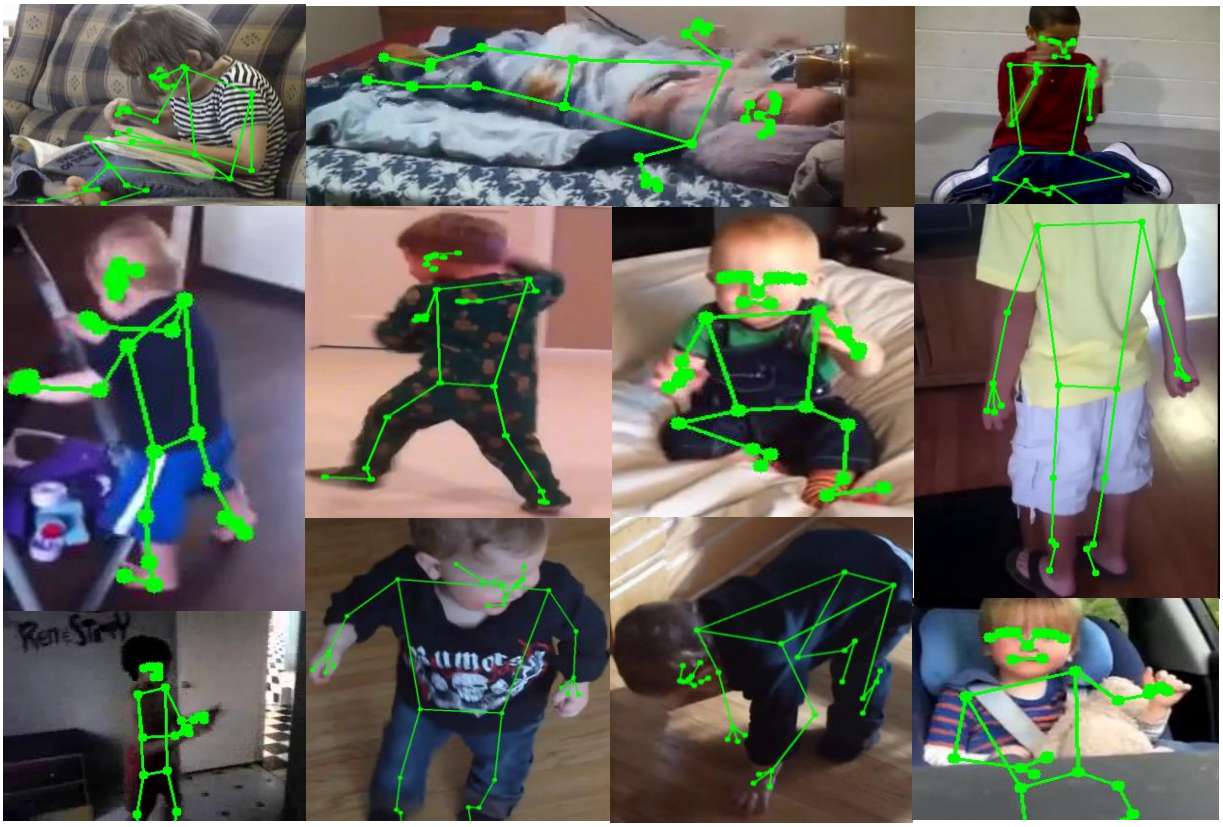
Skeletal Data Extraction and Processing

Now it is time to convert raw videos into the skeletal format. For this task, we have used MediaPipe, an open-source Machine learning framework developed by Google for real-time computer vision tasks, including human pose estimation. It is a low-latency, lightweight, cross-platform solution for desktop and mobile applications (Lugaresi et al., 2019). MediaPipe Holistic is a module within the framework that can recognize a human being and gives

information on the bounding box and joint coordinates. It is like a skeleton with 33 anatomical key points representing human body joints, including upper and lower body landmarks. The examples of skeletons produced are shown in **Figure 4.** MediaPipe Holistic can provide 3D spatial coordinates (x, y, z) for each key point, where the z-dimension is the relative depth of the joints, so you can better understand body dynamics.

Figure 4

Skeletonized video frame examples

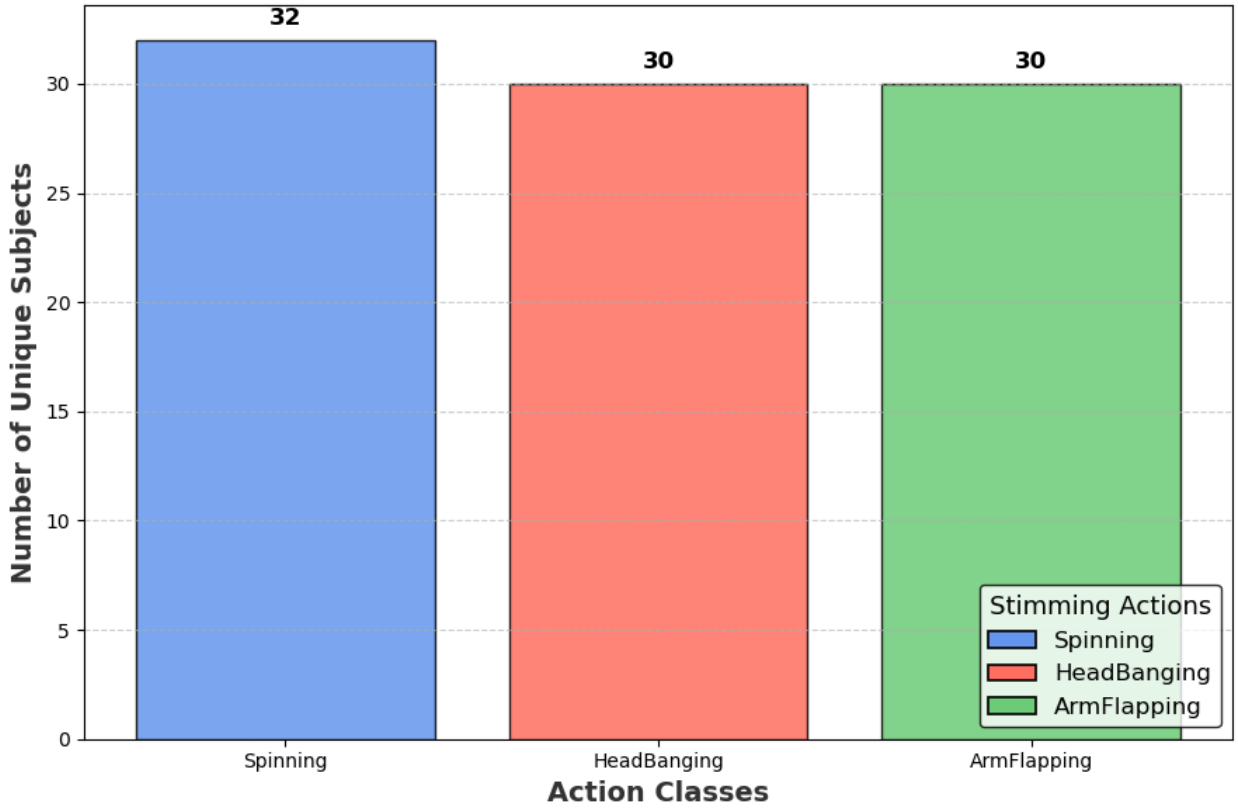


Each frame in a video clip is run through the MediaPipe Holistic model, which produces 33 key points representing the important human joints in 3D space (x, y, z). The joint coordinates from each frame are concatenated over the clip to create a temporal sequence of the body movement. The sequences from each video clip are saved as NumPy arrays to make storage

efficient and compatible with machine learning workflows. This organized format reduces computational requirements compared to raw video data. It is more resilient to changes in appearance, lighting, and background, so the model can focus on the body movement related to stimming behaviors.

Figure 5

Distribution of subjects across all the classes after cleaning



After cleaning the noisy videos and clips, the dataset has 32 subjects for Spinning, 30 for Headbanging, and 30 for Arm Flapping. The distribution of classes and the number of videos is shown in **Figure 5.**

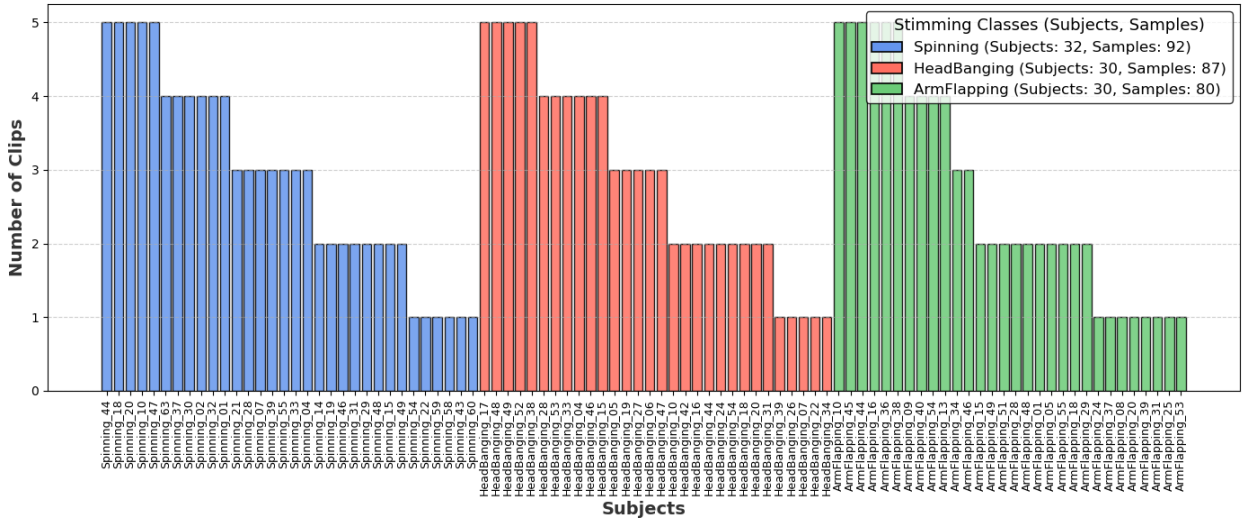
Temporal Normalization and Augmentation:

A temporal normalization, where the nonuniform frames are interpolated to match the required frames, is applied to all input sequences to ensure consistency across the dataset.

Furthermore, the excess frames are averaged. This is to avoid inconsistencies due to different video lengths and to sync all clips to the exact temporal resolution. The normalization resizes each segment to 72 frames using interpolation to preserve temporal structure and motion information. A median filter is also applied to smooth the joint trajectories and remove noise from pose estimation errors. These pre-processing steps make the dataset temporally consistent and noise-robust to train deep learning models.

Figure 6

Samples in Each Subject Before Augmentation



Each subject's video varies in duration, resulting in differences in the number of video clips the subject has. This may lead to data imbalance and bias in mode learning; we can see this from **Figure 6**: few subjects have only one clip, whereas some have as many as five clips. A deep learning model may learn more or overfit on subjects that have more clips. Some subjects have very few clips, which would not contribute enough to learning, as a model cannot learn patterns with few examples. This imbalance would cause poor generalization of the model.

To address these issues, we have decided to augment the video clips and balance the number of clips for each subject. An augmentation strategy with variable scaling, rotation, and

translation properties is employed on the videos. Various levels of augmentation are performed on subjects to produce different sizes of datasets. A data set with one clip per subject has a size of 92, another dataset with three clips per subject has a size of 276 videos as shown in **Figure 7**, another dataset is augmented to 5 videos per subject, so it has 460 videos and a dataset augmentation with 10 videos and 15 videos per subject has size of 920 and 1380 shown in **Figure 8**.

Figure 7

Dataset After Augmentation of 1 Video for Subject

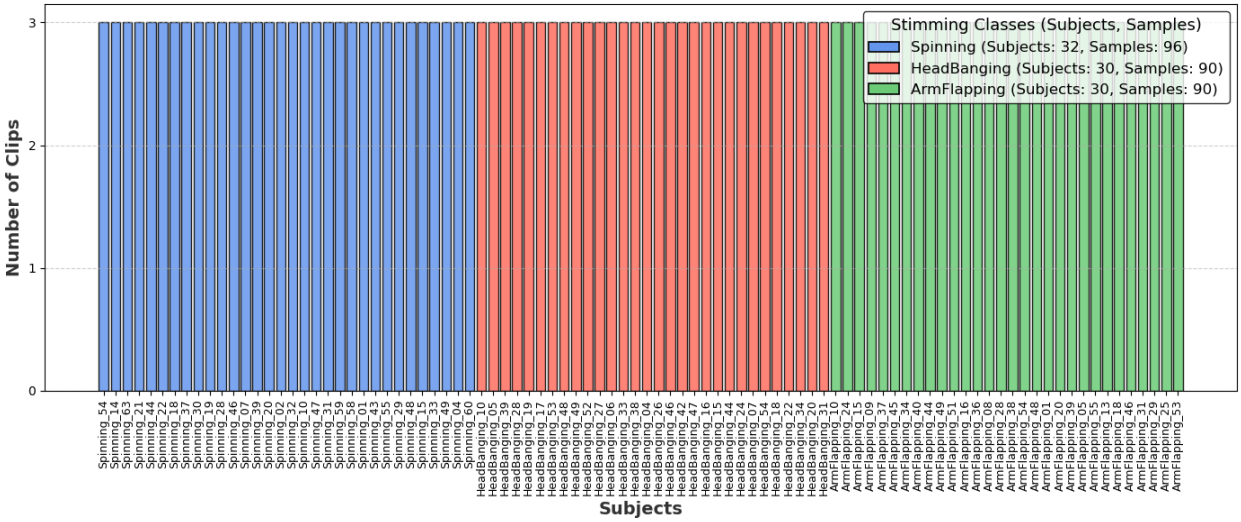


Figure 8

Dataset After Augmentation of 10 Videos for Subject

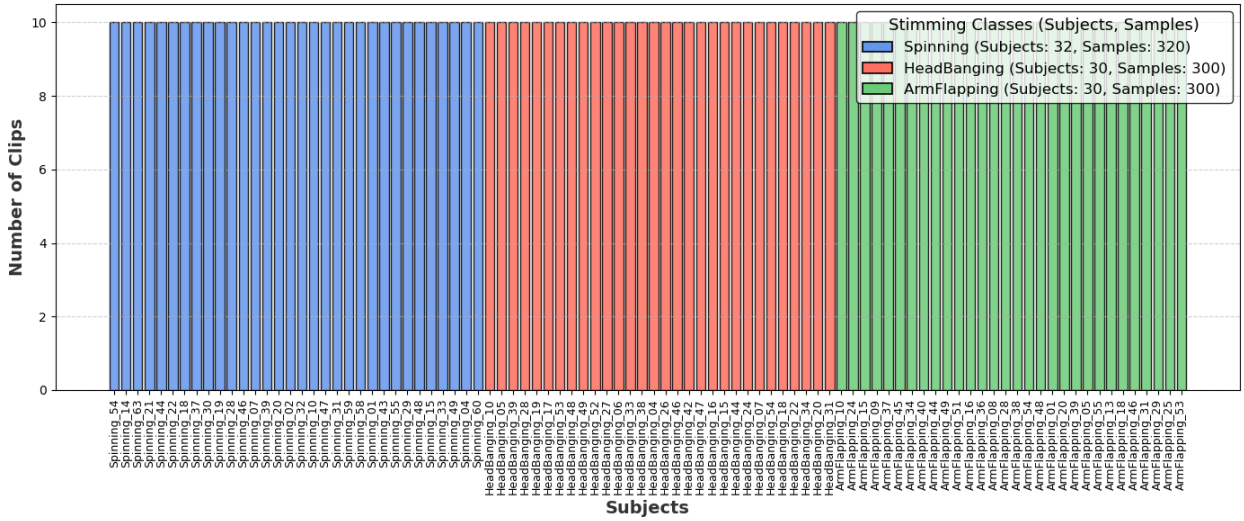


Table 2

Dataset size after various numbers of augmentations

Number of Videos per Subject	Size of the Dataset
1	92
3	276
Original [1-5]	289
5	460
10	920
15	1380

Table 2 contains information about levels of augmentations applied and how the dataset size is balanced and increased.

Dataset Organization for Models:

We trained the models using k-fold cross-validation, which can give us the correct evaluation of the model on all data samples, and to make the train and test sets mutually exclusive, during the k-fold, we divide the train-test sets based on subject ID and class label during k-fold and assign all clips of that subject to respective class. So, we did not divide the data into train and test sets during the data processing stage. Instead, the data is saved as a pickle file; each file in the data pickle is a NumPy array with joint coordinates and class labels. The naming convention Armflapping_01_clip_01 is followed, with its subject ID, class name, subject number, and clip number.

In each fold, we divide the subjects into train and test links and assign each subject's clips to its respective set; this will ensure that the train and test sets are mutually exclusive and that every clip appears only in one of the k folds.

Chapter 4: Architectures and Methodologies

StimNet: Temporal Pose Differentiation Model

Model Architecture

The temporal pose differentiation-based stimming action recognition model is a specialized deep learning framework tailored to detect actions in skeletal motion data extracted from video sequences. This model is inspired by the work of Yang et al. (2020), who proposed a view-adaptive neural network for general skeletal-based action recognition, which has been adapted in our study for the specific task of recognizing stimming behaviors. This model uses one-dimensional convolutional neural networks (1D-CNNs) to process temporal pose variations to identify subtle differences in joint movements over time. The network is a hybrid model combining two fundamental modules: the Feature Extraction Module (FEM) and the Classification Module (MLP) as shown in **Figure 9**.

Convolutional Layers: The Feature Extraction Module (FEM) is designed to encode both the skeletal pose sequences and their temporal derivatives into meaningful feature representations. The module is structured to handle three distinct input streams: (i) the skeletal pose sequence in the form of joint correlation distances (JCD), (ii) slow-motion pose differences, which is the difference between the consecutive frames, and similarly (iii) fast motion pose which is difference between alternate frames in a clip. These inputs allow the network to capture movement patterns at multiple temporal scales, both inter and intra-frame. Each stream is processed independently through a series of 1D convolutional blocks. Each convolution block consists of normalization layers, activation, and dropout layers. A kernel of size three is applied in the convolution layers, which is suitable for capturing local temporal dependencies within the skeletal sequence.

Activation Functions: We use the LeakyReLU activation function after each convolution layer to introduce non-linearity to the model. LeakyReLU is chosen over standard ReLU to mitigate the issue of dead neurons by allowing small gradients even when the input is negative. This activation function aids in improving gradient flow during training, leading to faster convergence and better performance.

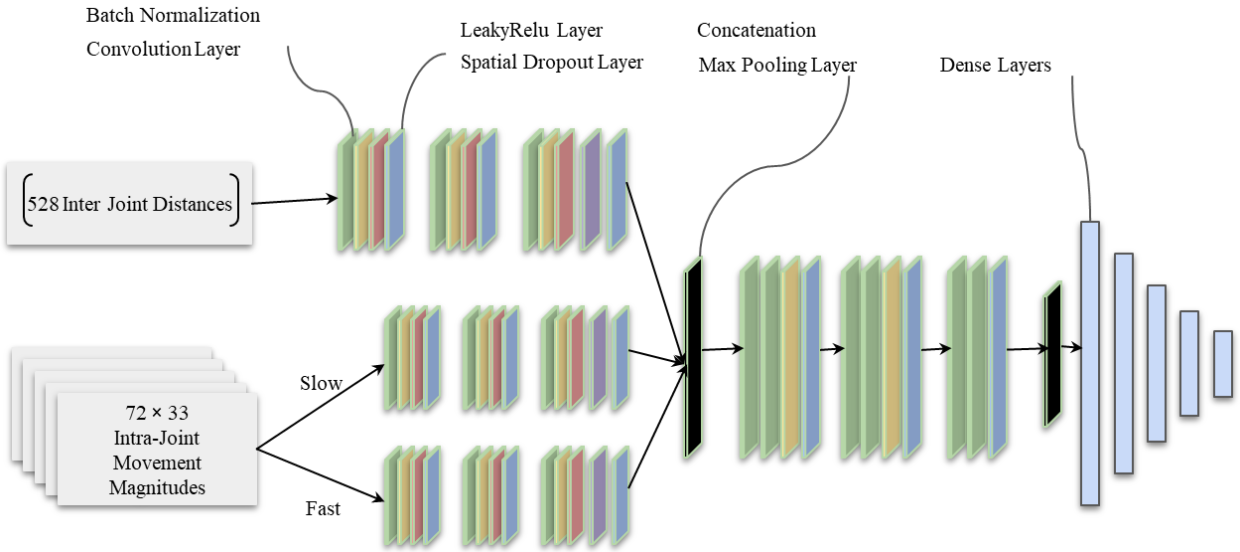
Dropout Layers: As our dataset is small, overfitting is a big threat. To avoid overfitting and increase the generalization capacity of the model, spatial dropout layers are inserted after each convolutional block. Spatial dropouts randomly delete whole feature maps during training and let the network rely on a series of independent feature representations instead of a particular activation. This dropout regularization technique is beneficial when working with skeletal motion data, as it ensures the robustness of learned features and the model does not overfit.

Pooling Operation Layer: Max-pooling layers are incorporated after the feature extraction module to progressively reduce the dimensionality of the extracted features while retaining the most important information. Max-pooling is effective in down-sampling feature maps and highlighting the dominant motion patterns within the skeletal sequence.

Fully Connected Network: The Classification Module (MLP) processes the extracted feature representation through a series of fully connected layers. It begins after a global max-pooling operation to condense the multi-stream feature maps from convolution filters into a fixed-length feature vector. This vector is then passed through four fully connected layers with 254, 128, 64, and 32 neurons, respectively. To stabilize the learning process, batch normalization and LeakyReLU activation are applied after each dense layer. A dropout layer with a rate of 0.5 of regulation is used after each dense layer to prevent overfitting. Finally, a SoftMax layer with four output neurons generates the probability distribution over the four stimming action classes.

Figure 9

Temporal Pose Differentiation Model



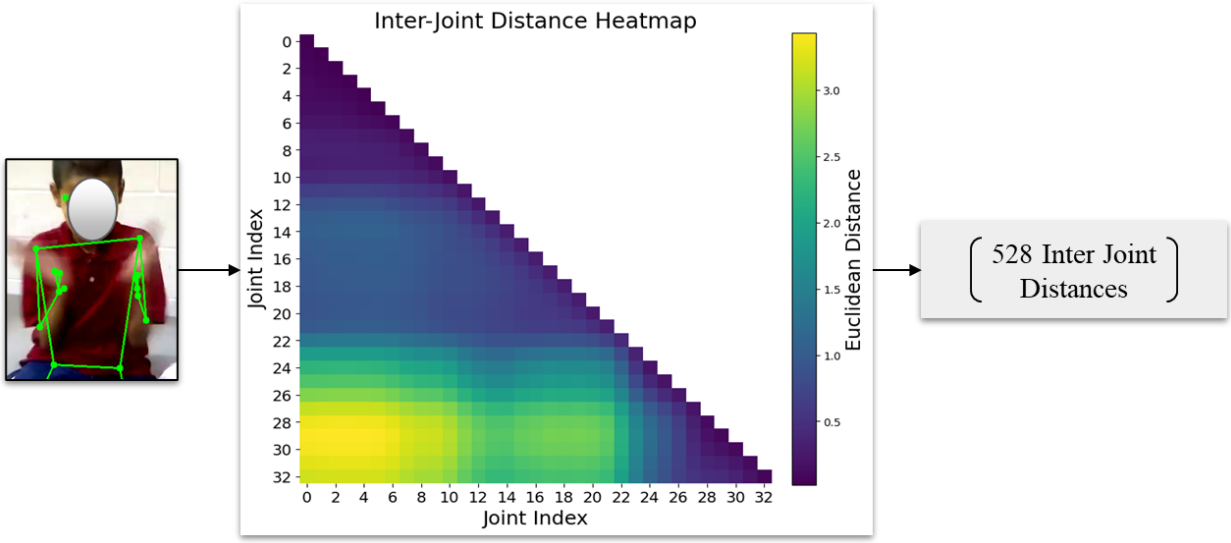
Implementation of the Temporal action recognition model

The implementation of the temporal pose differentiation-based 1D-CNN model is closely aligned with the characteristics of the stimming action recognition task. The unprocessed skeletal information obtained from each video segment is depicted as a series of 33 key points for each frame, with every key point comprising 3D coordinates (x, y, z) for all 72 frames in a segment. This dataset constitutes the Pose Input (P), the inter-joint distance among 33 joints in each frame. The skeleton with 33 3D coordinates is converted into an array of 528 inter-joint distances. The process of calculating inter-joint distances is shown in **Figure 10.**

Every video sequence is adjusted to a standard length of 72 frames through the interpolation techniques that Yang et al. (2020) have proposed. This adjustment guarantees uniform input sizes, enabling the model to handle sequences of different frame lengths.

Figure 10

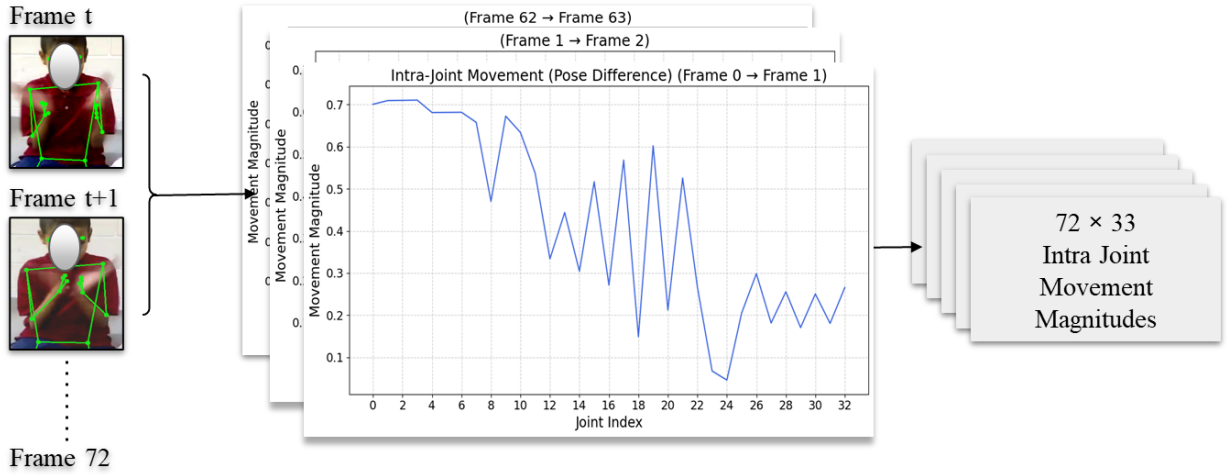
Inter-joint Distance Calculation



Temporal pose differentiation is applied to the skeletal sequences to capture movement dynamics. This process involves computing the frame-wise differences in joint positions to derive two additional feature streams (Yang et al., 2020): slow-motion differences and fast-motion differences. The process is shown in **Figure 11**. The gradual pose transitions are captured by computing the slow-motion difference stream across all frames, whereas a down-sampled sequence produces the fast-motion difference stream to emphasize quick joint movements.

Figure 11

Intra-Joint distance calculations



These distinct pose characteristics act as supplementary inputs, aiding the model in identifying both subtle and sudden changes in movement. The pose input and motion difference streams are transformed into tensors suitable for the 1D-CNN model: the pose sequence is formatted as (frame length, number of joints, joint dimensions), while the motion features are structured as (frame length, feature dimension).

The preprocessed data is fed into the multi-stream feature extraction module, with each stream processed independently through convolutional layers before being fused. The fused feature representation is then passed to the classification module for final action prediction. Adam optimizer is used in training with an initial learning rate of 0.001, and categorical cross-entropy is employed as the loss function with a batch size of 32.

Upon training completion, the model outputs a probability distribution of over four stimming action categories through a SoftMax layer. Subsequently, the most probable class is selected as the predicted label. This multi-stream, motion-sensitive approach allows the model to

detect subtle as well as rapid repetitive motions, making it highly suitable for detecting stimming behaviors from video data.

Local StimNet: Localized Temporal-CNN Based Model

Model Architecture

The localized temporal CNN-based stimming action recognition model is developed using the same architecture as the previous temporal CNN model by localizing the feature extraction at action-specific regions in the skeleton. This model's uniqueness is that it does not treat the skeleton as a single entity. It processes multiple body regions independently. By isolating different body segments, the model captures region-specific movement patterns, leading to improved recognition of fine-grained motor actions often exhibited in stimming behaviors.

Feature Extraction Module (FEM): As discussed in the previous model, this module extracts motion dynamics through a series of 1D convolutional layers followed by normalization, activation, and dropout layers. Skeletal data is processed frame by frame, with temporal pose differentiation applied to emphasize motion transitions. Here, this module is applied to individual regions of the skeleton.

Multi-Stream Parallel Networks: The architecture includes four independent sub-networks (FEM), each responsible for a feature extraction of a different body region:

Head: This network convolves the head's skeletal points to identify head-specific movements such as tilting and nodding.

Arms: This network focuses on the skeletal joints of the arms to analyze upper limb actions, including arm-flapping.

Torso: This network extracts the movements from the remaining joints, which form the torso of the body.

The respective joint coordinates from the specific body regions are passed to the four FEM model heads. As discussed in our first model, the result is concatenated and further convoluted.

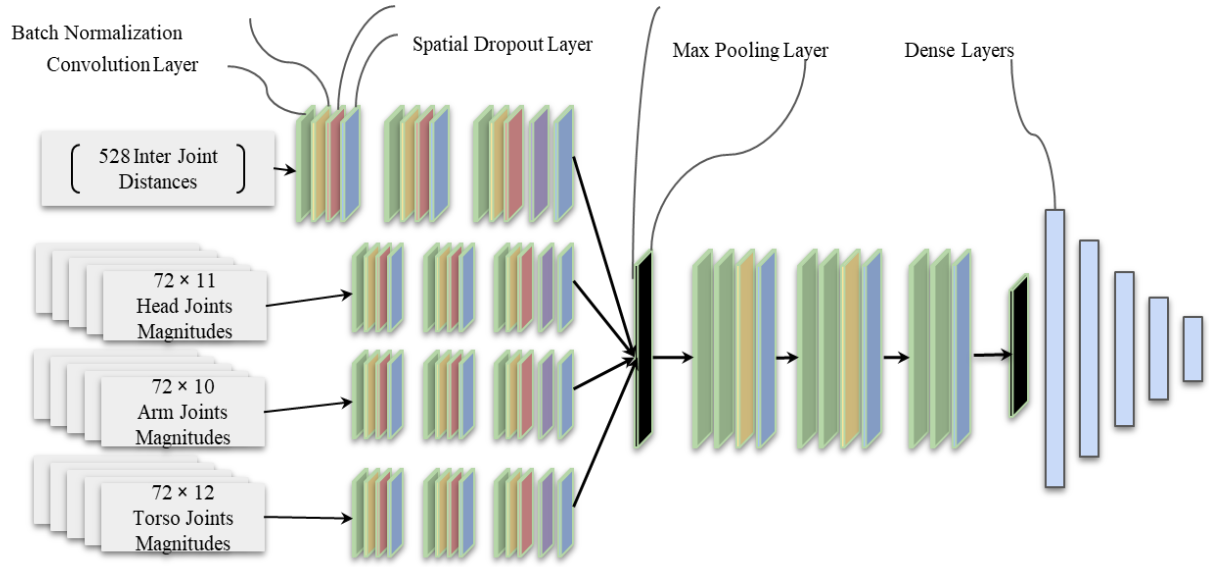
Classification Module (MLP): The output features from the four body regions are concatenated into a unified representation. Fully connected layers are then processed, and this fused feature map is used to classify the stimming action.

Fully Connected Layers

As shown in **Figure 12**, after parallel feature extraction, the outputs from the four body-region sub-networks are concatenated to form a unified feature vector. This combined representation is passed into the Classification Module (MLP), which consists of fully connected layers with 512, 254, 128, 64, and 32 neurons with the same settings as the previous.

Figure 12

Local StimNet Model Architecture



Implementation of Local StimNet

The implementation of the localized CNN-based model begins with the extraction of skeletal data from video sequences. Each frame is represented as a set of 33 key points with 3D coordinates (x, y, z), forming the Pose Input (P). The skeletal structure is partitioned into four body regions: torso, head, and arms. Each region's joint positions are isolated and prepared as separate input streams.

All video sequences are standardized to a fixed length of 72 frames to ensure uniform input dimensions. Each body region is formatted into tensors representing frame length, number of joints, and joint dimensions for pose input and frame length feature dimension for motion features.

The preprocessed data streams are fed into the four parallel sub-networks, each responsible for processing the assigned body region. The feature maps extracted from these sub-networks are concatenated and passed into the fully connected layers of the Classification

Module (MLP). The Adam optimizer is used for training, with an initial learning rate of 0.001. Categorical cross-entropy is the loss function, and a batch size of 32 is employed.

Once training is completed, the model generates a probability distribution for the four categories of stimming actions. The class with the highest probability is then chosen as the predicted label. This multi-stream, body-region-specific approach enhances the model's sensitivity to localized motion variations, making it especially effective in distinguishing between harmful and non-harmful stimming behaviors in children with autism.

StimAuto: Autoencoder-Based Multi-Stream Stimming Action Recognition Model

Model Architecture

The autoencoder-based action recognition model is developed to better understand and classify stimming behaviors. In contrast to previous CNN-based methods that largely depended on convolutional filters to capture motion characteristics, this model adopts an alternative approach by employing an autoencoder to develop compressed, noise-minimized representations of pose sequences.

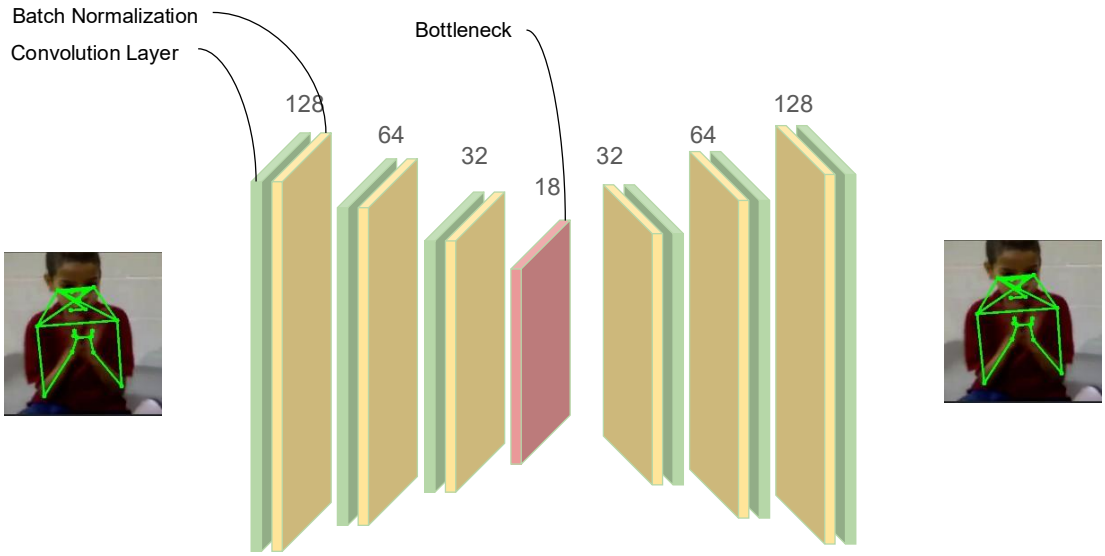
This model was built in two stages. First, we built an autoencoder, which consists of an encoder and decoder, and trained it with skeletal data. Then, we used the encoder part coupled with a classification module for classifying the actions.

Encoder-Decoder Structure: The autoencoder network is built with an encoder and decoder. The encoder compresses the pose sequence into a smaller feature space through 2D convolution layers, using ReLU activation for non-linearity, as shown in **Figure 13**. Once compressed, the decoder attempts to rebuild the original pose from this reduced representation. The difference between the original and reconstructed sequences, measured using mean squared

error, is backpropagated to guide the training process. Only the encoder is kept for extracting features during action recognition when training is complete.

Figure 13

Encoder-Decoder Architecture



Feature Extraction Module (FM): The encoder, which is trained to compress the skeleton to a lower dimension, is used as a feature extraction module. The weights of this model are frozen to preserve the ability to encode the features well.

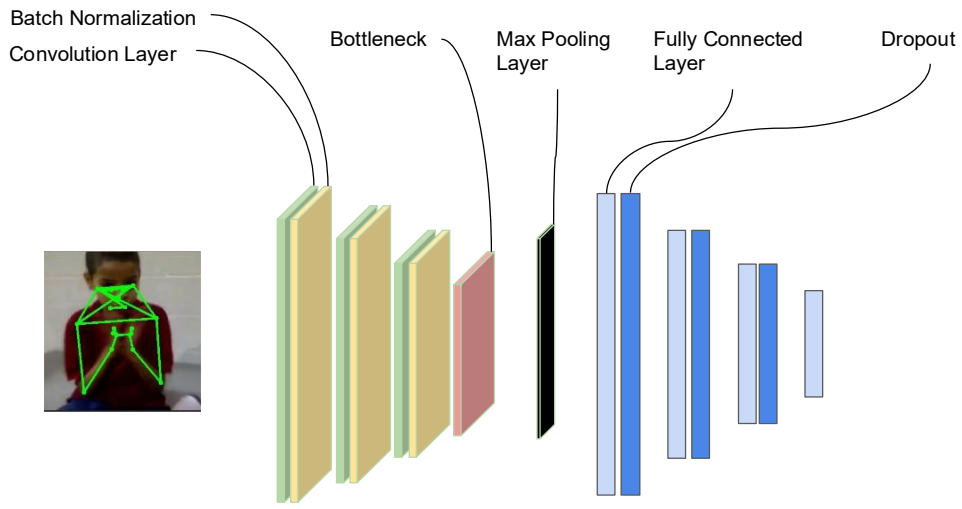
Classification Module: After the encoder processes the skeleton pose sequences, the outputs are combined into a single feature representation by applying global max pooling. This unified feature set is passed into a series of fully connected layers, which ultimately classify the stimming action.

The resultant feature vector of global max-pooling is then passed through fully connected layers with 128, 64, and 32 neurons. Each layer is followed by batch normalization and

LeakyReLU activation to ensure smooth training. Dropout regularization (0.5) is also applied to reduce the risk of overfitting. Finally, a SoftMax layer assigns probabilities across the four stimming action classes. The architecture of the autoencoder model is shown in **Figure 14.**

Figure 14

StimAuto Model Architecture



Implementation of Autoencoders

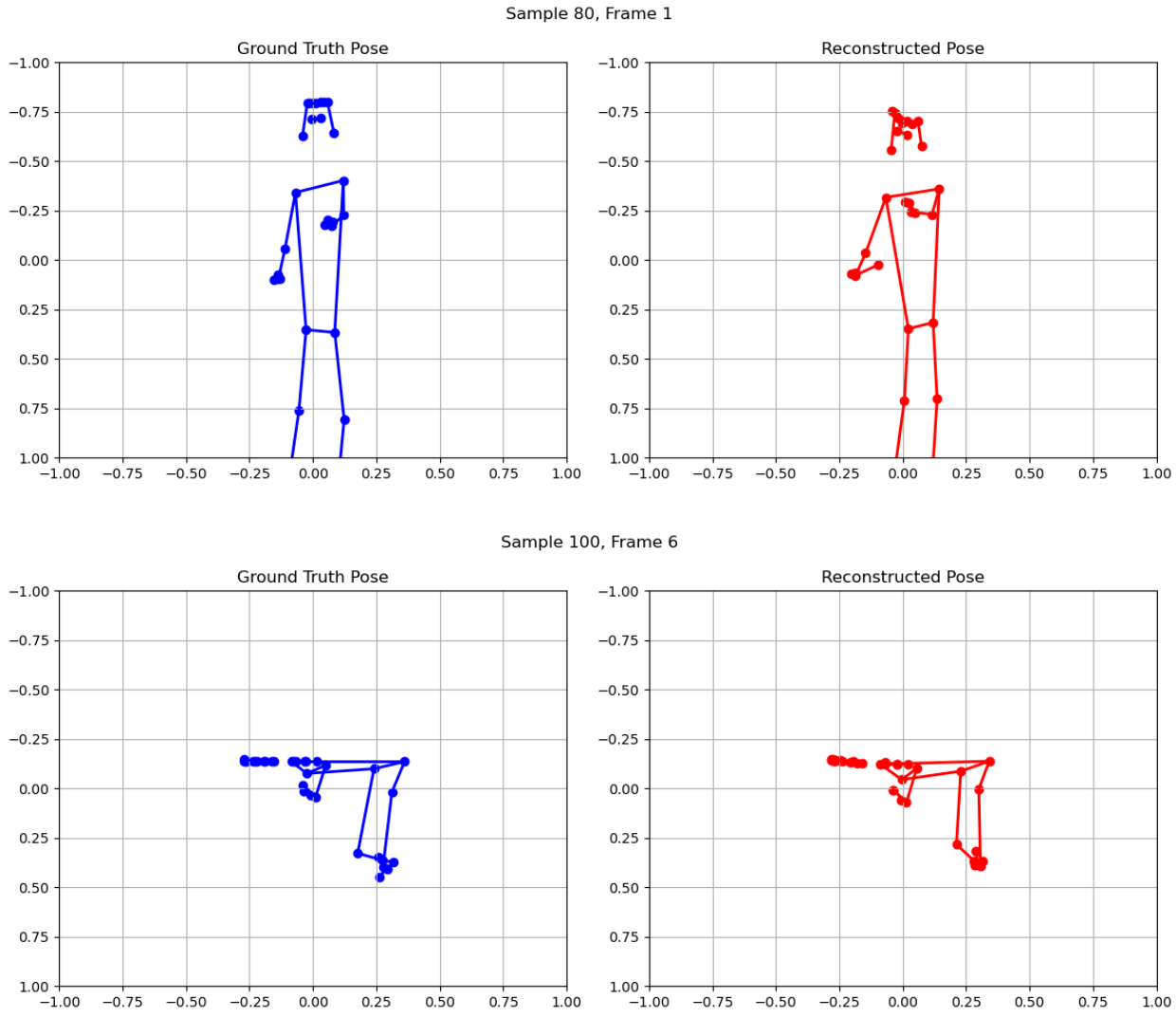
The implementation process begins with extracting skeletal data from video sequences, with each frame represented by 33 key points containing x, y, and z coordinates. During training, the encoder learns a compressed version of the pose sequence, while the decoder tries to reconstruct the original data. The autoencoder is trained using mean squared error as the loss function, optimized with Adam, and a learning rate of 0.001. Each region's autoencoder is trained separately.

After training, the encoder part of each autoencoder is extracted and used for feature extraction in the multi-stream network. During classification, the pose data is passed through the

encoders, and their outputs are combined into a single feature vector. This combined feature is fed into the classification module.

Figure 15

Ground truth vs Reconstructed pose by Autoencoder



Similar to the previous models, training utilizes the Adam optimizer and employs categorical cross-entropy as the loss function. To avoid overfitting, early stopping is implemented by tracking the performance on the validation set.

Once the model is trained, it predicts the action class by selecting the one with the highest probability from the four stimming categories. This method combines feature learning using autoencoders with multi-stream regional analysis, allowing the model to eliminate noise and concentrate on key motion patterns, which enhances its ability to identify and distinguish stimming actions.

StimVit: Vision Transformer-Based Stimming Action Recognition Model

Model Architecture

The Vision Transformer (ViT)-based model for stimming action recognition is a deep learning framework designed to classify stimming behaviors in autistic children using skeletal motion data. Unlike traditional convolutional networks, which focus mainly on localized spatial dependencies, ViTs utilize self-attention mechanisms to analyze long-range dependencies across entire sequences of poses. This capability to process global context allows them to effectively capture complex motion patterns. The ViT model used in this study is adapted from the original Vision Transformer proposed by Dosovitskiy et al. (2020), with specific modifications and fine-tuning to optimize it for the task of motion-based action recognition.

Components of ViT architecture: The Pose Sequence Embedding Module, the Transformer Encoder, and the Classification Module. The first component prepares the skeletal data by transforming it into a structured format suitable for the transformer. The second component processes these embeddings through self-attention layers to extract meaningful motion representations, and the final component classifies the stimming action based on these learned features.

Pose Sequence Embedding: The skeletal motion data is initially represented as a series of 33 key points in each frame, where each point consists of three-dimensional (x, y, z)

coordinates. Since transformers operate on sequential inputs rather than raw spatial data, the pose information is divided into smaller patches, with each patch containing a subset of key points. A linear transformation is then applied to project these patches into a high-dimensional feature space, which makes them compatible with the transformer model.

To maintain the temporal arrangement of movements, each patch is supplemented with trainable positional embeddings. Transformers typically do not handle data in a sequential manner, which makes these embeddings crucial for recording the sequence of movements throughout the frames.

Transformer Encoder: The model's foundation is the Transformer Encoder, which comprises several layers of self-attention and feed-forward networks. The self-attention mechanism enables the model to examine the relationships among different frames, allowing it to assign varying levels of importance to different time steps based on their relevance to the overall movement pattern. This approach allows the network to capture both subtle and significant motion variations, making it highly effective in recognizing stimming behaviors.

Each encoder layer follows a structured process. It begins with layer normalization, which helps stabilize training by standardizing the input distributions. The next step involves multi-head self-attention, where the model calculates relationships among all pose patches in the sequence. This approach allows the network to focus selectively on important frames while disregarding less relevant information. The output from the attention mechanism is then directed through a feed-forward network, which adds non-linearity to enhance the model's ability to learn complex motion patterns. Additionally, residual connections are incorporated at every stage to facilitate gradient flow, ensuring that the model learns efficiently and converges effectively during training.

While CNN-based architectures tend to focus on local motion changes, the ViT model's attention mechanism allows it to capture a more holistic view of stimming actions. This broader perspective helps distinguish between different stimming behaviors that may share similar localized movements but differ in their overall temporal structure.

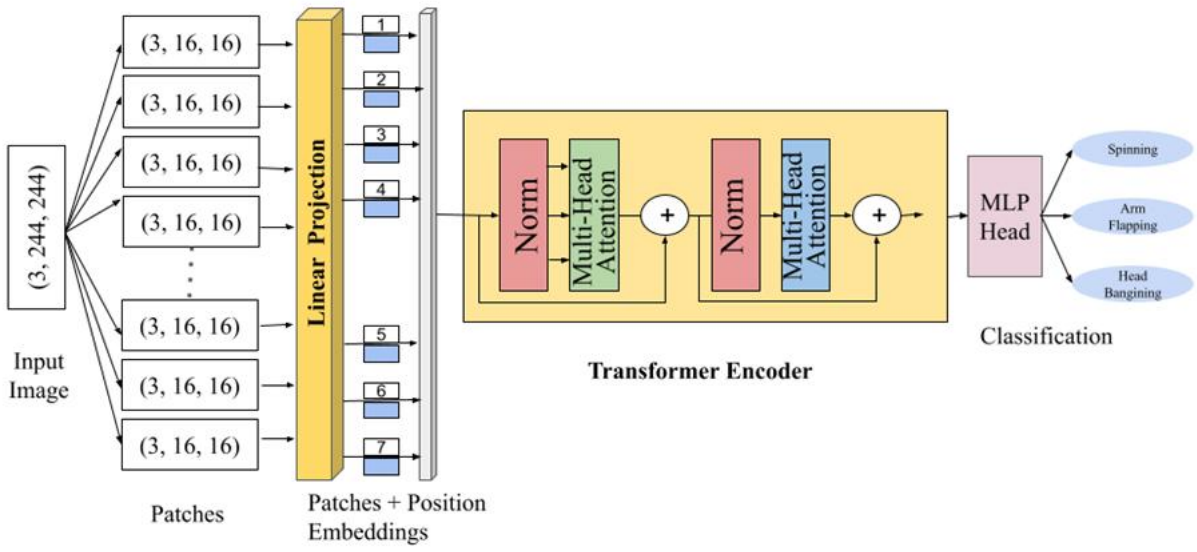
Classification Module: After the skeletal sequence has been processed through the transformer encoder, the final feature representation is prepared for classification. A global average pooling operation is first applied to condense the sequence representation into a fixed-length feature vector. This compact representation is then passed through a series of fully connected layers, which refine the learned features and enhance their discriminative power.

The classification module is structured with fully connected layers consisting of 786 which is the embedding output, intermediate layer with 512 and 4 neurons in final layer. Batch normalization is applied after each layer to stabilize learning, while ReLU activation introduces non-linearity and prevents neuron saturation. To reduce overfitting and improve generalization, dropout regularization is employed at a rate of 0.5 after each dense layer.

The final stage of the classification module is a SoftMax activation function applied to final layer, it generates a probability distribution over the four stimming action classes. The action category with the highest probability is selected as the model's predicted label.

Figure 16

StimViT Architecture



Implementation of Stimming Action Recognition

Our Vision Transformer (ViT)-based model follows a structured pipeline to process skeletal motion sequences and classify them into different stimming behaviors. Each frame consists of 33 key points, each represented by (x, y, z) coordinates. A skeletal sequence of shape (72,33,3) is reshaped to (3,72,33) and resized to (3,244,244), which matches the input expected by ViT, where x, y, and z coordinates of the skeleton are treated as RGB channels of the image. We segment the motion data into patches of size (3,16,16) and transform each patch into an embedding using a linear projection layer. We add positional encodings to these embeddings to retain the temporal order of movements before passing them into the transformer model.

Once the pose sequences are converted into embeddings, they are processed by ViT-Base-Patch16-224, a pre-trained transformer model. It consists of 12 transformer encoder layers,

each with multi-head self-attention and feed-forward networks to analyze long-range dependencies in skeletal motion. The model initially had 12 attention heads designed for image-based tasks, but we modified the input embedding structure to treat x, y, and z coordinates as separate input channels to better suit skeletal motion analysis. The classification head then takes the processed output and assigns the most probable action label.

Since our dataset is relatively small, we freeze the first six transformer layers and fine-tune layers 7-12 along with the classification head. This ensures that lower-level feature representations remain stable, while higher layers adapt to stimming behavior recognition. We also freeze the input embedding layers and positional encodings to preserve the pre-trained spatial structure.

For training, we use the Adam optimizer with a learning rate of 0.0001, which helps the model learn efficiently without making drastic updates that could destabilize training. Since our task involves classifying stimming behaviors into multiple categories, we use categorical cross-entropy as the loss function, which ensures the model learns to differentiate between the three action classes accurately. To prevent overfitting, we implement early stopping, which monitors the validation loss and stops training if the model starts memorizing the training data instead of generalizing well.

Once trained, the model predicts stimming actions by analyzing the probability distribution over the four action categories and selecting the most likely class as the final prediction. Thanks to the self-attention mechanism in ViT, the model can effectively capture long-range dependencies in motion sequences. This makes it particularly useful for identifying repetitive behaviors

Training Configurations

Table 3 shows the training configurations for the models. StimNet, Local StimNet, and StimAuto are developed on the TensorFlow framework, and StimViT is developed on the PyTorch framework. So, StimNet's Hyperparameters differ slightly from those of the other models.

Table 3

Training configurations

	StimNet	**Local StimNet**	**StimAuto**	**StimViT**
Input Type	Joint Distances	Joint Distances Arms/Head/Torso	Latent Embeddings (AE)	Flattened Joints (ViT format)
Framework	TensorFlow / Keras	TensorFlow / Keras	TensorFlow / Keras	PyTorch
Optimizer	Adam	Adam	Adam	AdamW
Learning Rate	1e-3 (ReduceLROnPlateau)	1e-3 (ReduceLROnPlateau)	1e-4 (ReduceLROnPlateau)	1e-4 (fixed)
Loss Function	Categorical Cross Entropy	Categorical Cross Entropy	Categorical Cross Entropy	Cross Entropy

Activation	LeakyReLU + SoftMax	LeakyReLU + SoftMax	LeakyReLU + SoftMax	ReLU + SoftMax
Batch Size	32	8	16	16
Epochs	300 (early stop, 20 patience)	256 (early stop, 20 patience)	256 (early stop, 20 patience)	10
Dropout	0.4	0.3	0.4	0.2

Chapter 5: Results

This chapter will discuss the results of the models we trained and tested for stimming action recognition. We used a combination of standard classification metrics and more advanced measures. The key evaluation metrics in this study include K-fold accuracy, precision, recall, and F1-score, along with additional metrics such as area under the curve (AUC) and Confusion matrix.

Evaluation Metrics Used

Accuracy is the well-known evaluation metric used for classification tasks. It measures the proportion of total correctly classified samples out of the total samples. We take the mean of K-fold accuracy as the final accuracy of the model.

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

Where:

TP (True Positives): Correctly predicted positive cases.

TN (True Negatives): Correctly predicted negative cases.

FP (False Positives): Incorrectly predicted positive cases.

FN (False Negatives): Incorrectly predicted negative cases.

Precision measures how many of the instances predicted as positive are correct. This is important in cases where false positives are costly.

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

A high precision score means that the model does not misclassify negative instances as positive too often.

Recall measures how well the model can identify actual positive cases. It is important in safety-critical applications, where missing an instance of a particular class, for example, harmful actions like headbanging in our case, could have severe consequences.

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

High recall ensures that fewer relevant instances are misclassified.

The F1 score is calculated using precision and recall, making it a more reliable metric when classes are imbalanced. It is the harmonic mean of precision and recall:

F1= $2 \times \frac{Precison \times Recall}{Precison + Recall}$

A high F1 score indicates that both false positives and false negatives are minimized, making it a better choice than accuracy alone when dealing with real-world datasets.

Since our task involves recognizing actions from video sequences, standard classification metrics alone are not enough. We also evaluate the model using metrics specifically designed for temporal action recognition.

Area Under the Curve (AUC) is used to measure how well the model separates different classes. It is derived from the Receiver Operating Characteristic (ROC) curve; it plots the true positive rate (recall) against the false positive rate (FPR)

FPR=$\frac{FP}{FP+TN}$

TPR= $\frac{TP}{TP+TN}$

AUC values closer to 1.0 indicate better model performance, as this means the model effectively distinguishes between different classes.

The confusion matrix helps us see where our model gets things right and where it struggles. Instead of just looking at overall accuracy, it breaks down predictions for each action category,

showing which behaviors are often confused with one another. This is especially useful for understanding if the model is biased toward certain actions or if it consistently misclassifies similar movements. By analyzing these patterns, we can fine-tune the model by adjusting training data, improving preprocessing, or tweaking the architecture to ensure better recognition of all stimming actions.

StimNet

The following results for the StimNet model were obtained after applying data augmentation by increasing the number of videos to 5 per subject. This augmentation strategy helped balance the dataset and improve model generalization during training and evaluation.

The training and validation loss curves for the StimNet model, trained with five videos per subject through data augmentation, show a steady downward trend across epochs, confirming successful convergence as shown in **Figure 17**. Although minor fluctuations appear in the validation loss, the overall pattern indicates stable learning without severe overfitting. This suggests that the StimNet model effectively captured important motion features necessary for stimming action classification.

Figure 17

Loss functions of StimNet

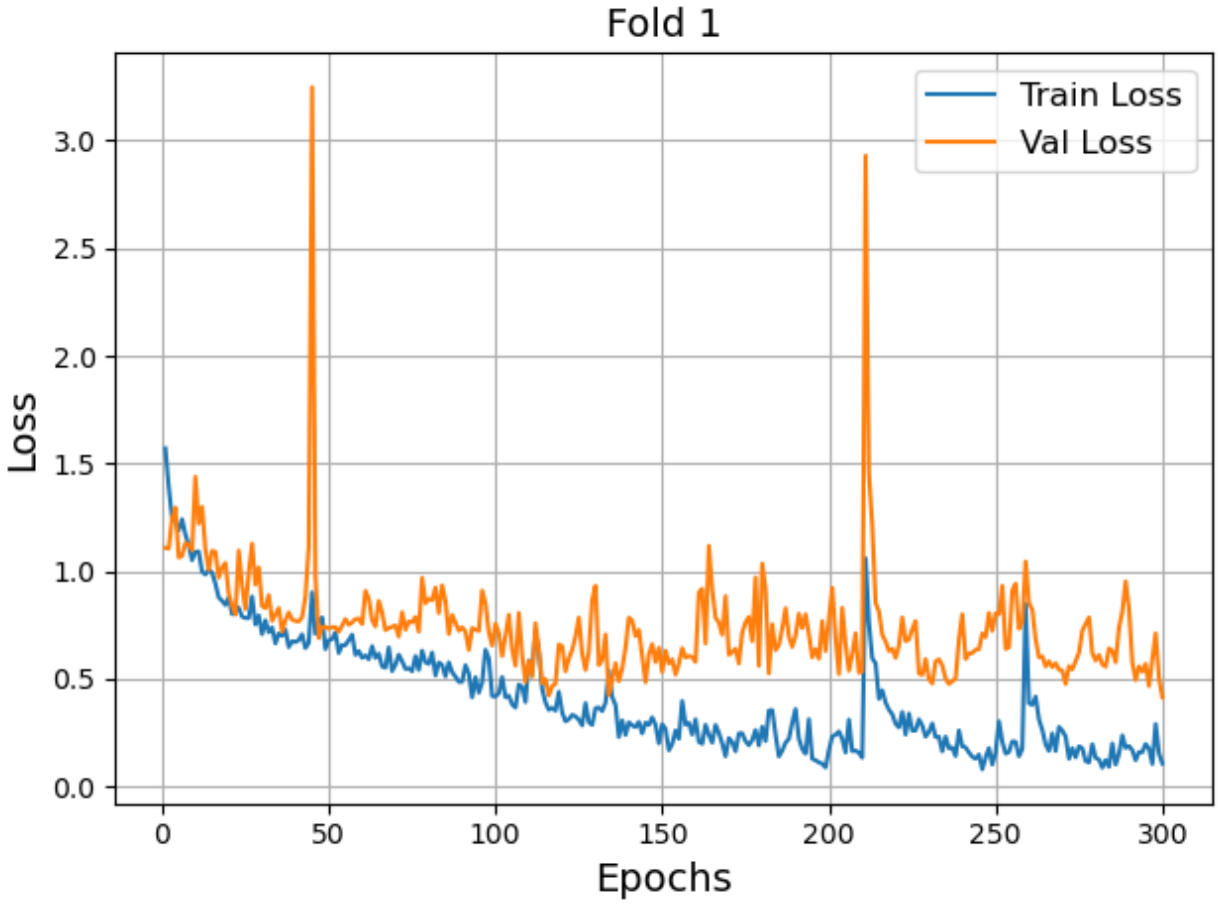


Figure 18 shows the performance metrics across folds, including Accuracy, Precision, Recall, and F1 Score. The results demonstrate that the model consistently maintains high performance across all folds, with metric values largely clustered between 0.85 and 0.92. Although a slight dip in Precision and F1 Score is observed in Fold 3, the overall performance remains stable, indicating strong generalization and robustness. The close alignment of Precision, Recall, and F1 Score across folds further suggests that the model is well-balanced, effectively

handling both false positives and false negatives. These results highlight the reliability of the classification models developed for stimming action recognition.

Figure 18

Accuracy, Precision, Recall, and F1 Score plot across five folds of StimNet

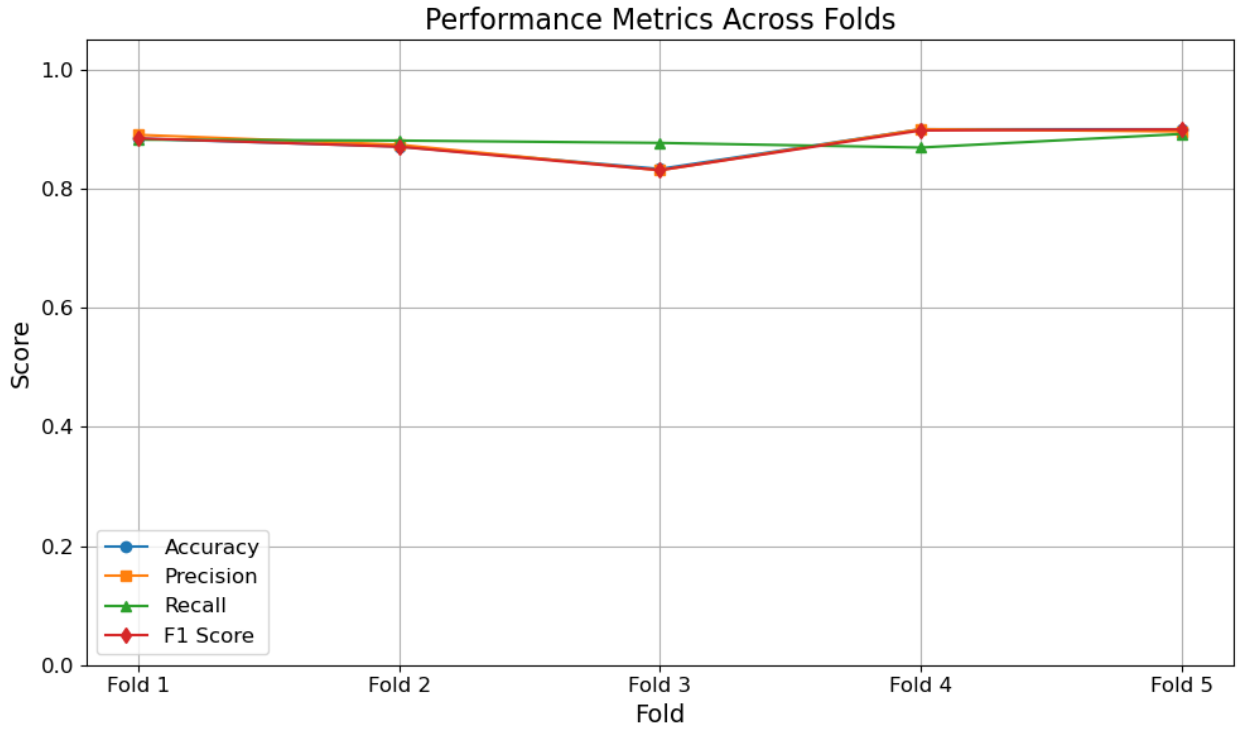


Figure 19 shows the Receiver Operating Characteristic (ROC) curves for each fold during the five-fold cross-validation process. The model consistently achieves high performance across all folds, with Area Under the Curve (AUC) values ranging between 0.90 and 0.92. These results demonstrate that the classifier maintains strong generalization ability and reliable discrimination between classes, with minimal variability across different data splits. The high AUC values across folds indicate the model's robustness and its effectiveness in recognizing stimming actions from skeletal data.

Figure 19

ROC plots of StimNet across five folds

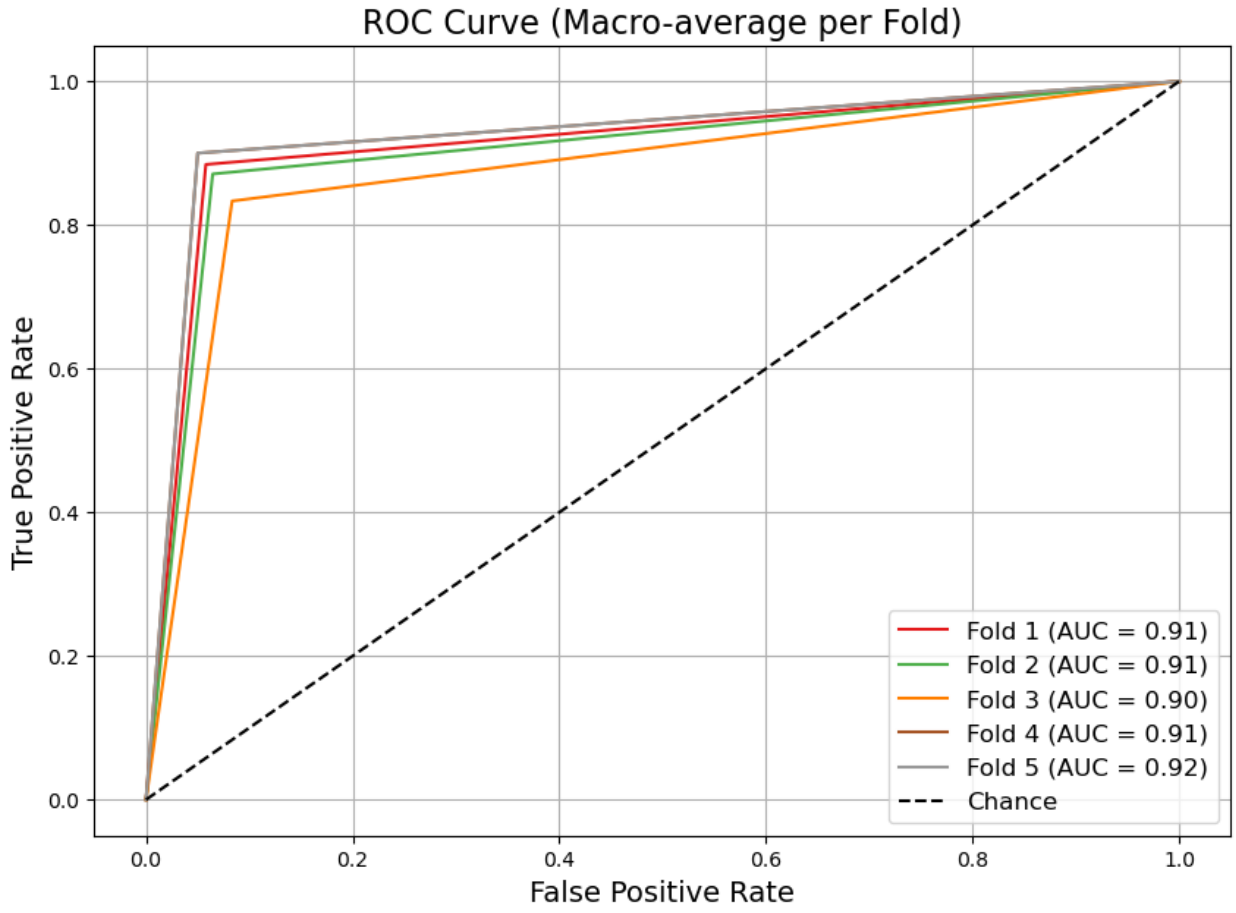
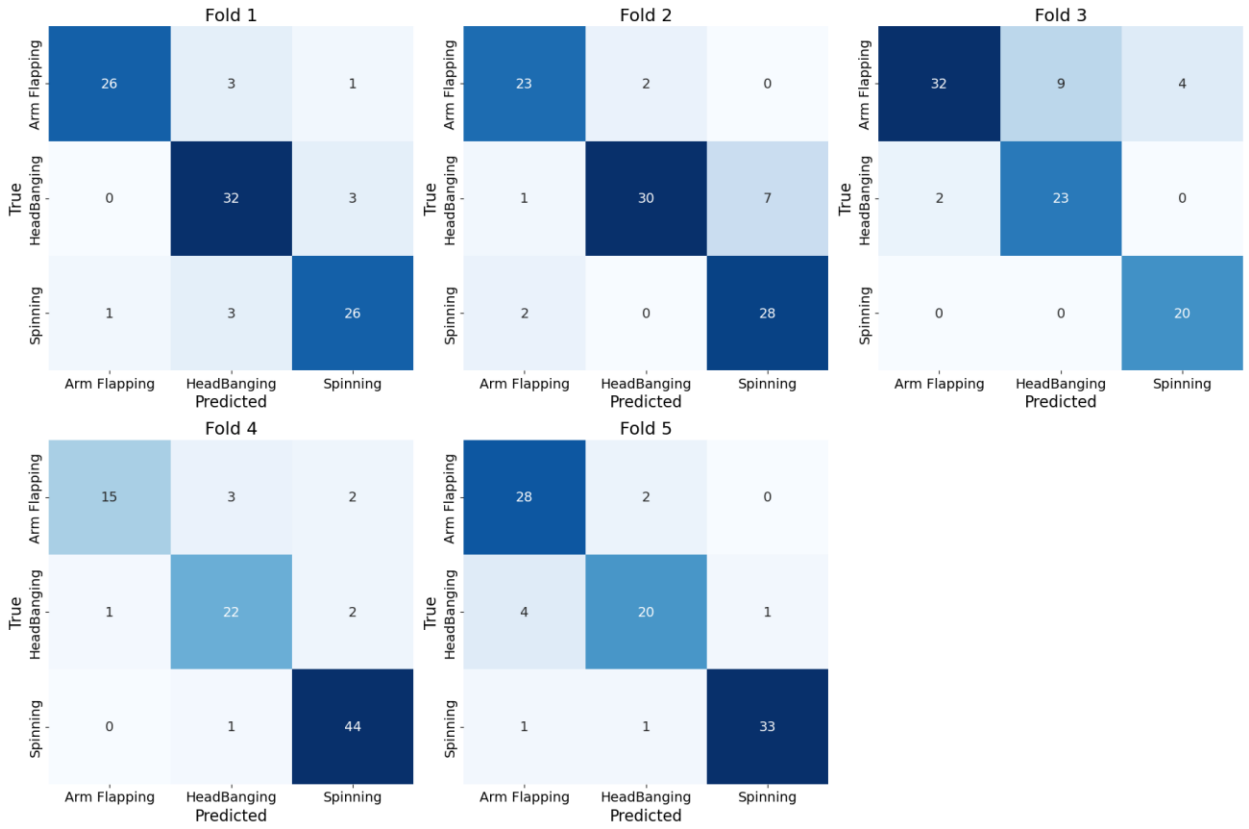


Figure 20 presents the confusion matrices for each fold during five-fold cross-validation, showing the model's classification performance across the three stimming action classes: Arm Flapping, Head Banging, and Spinning. The diagonal elements represent the number of correct predictions for each class, while off-diagonal elements indicate misclassifications. Across all folds, the model demonstrates strong predictive capability, particularly for Head Banging and Spinning, with minimal confusion between these classes. Some degree of misclassification is observed between Arm Flapping and the other categories, which may be attributed to

overlapping motion patterns. Overall, the confusion matrices confirm the model's consistent ability to accurately distinguish between stimming actions across different data splits.

Figure 20

StimNet confusion matrix for five folds



Localized StimNet

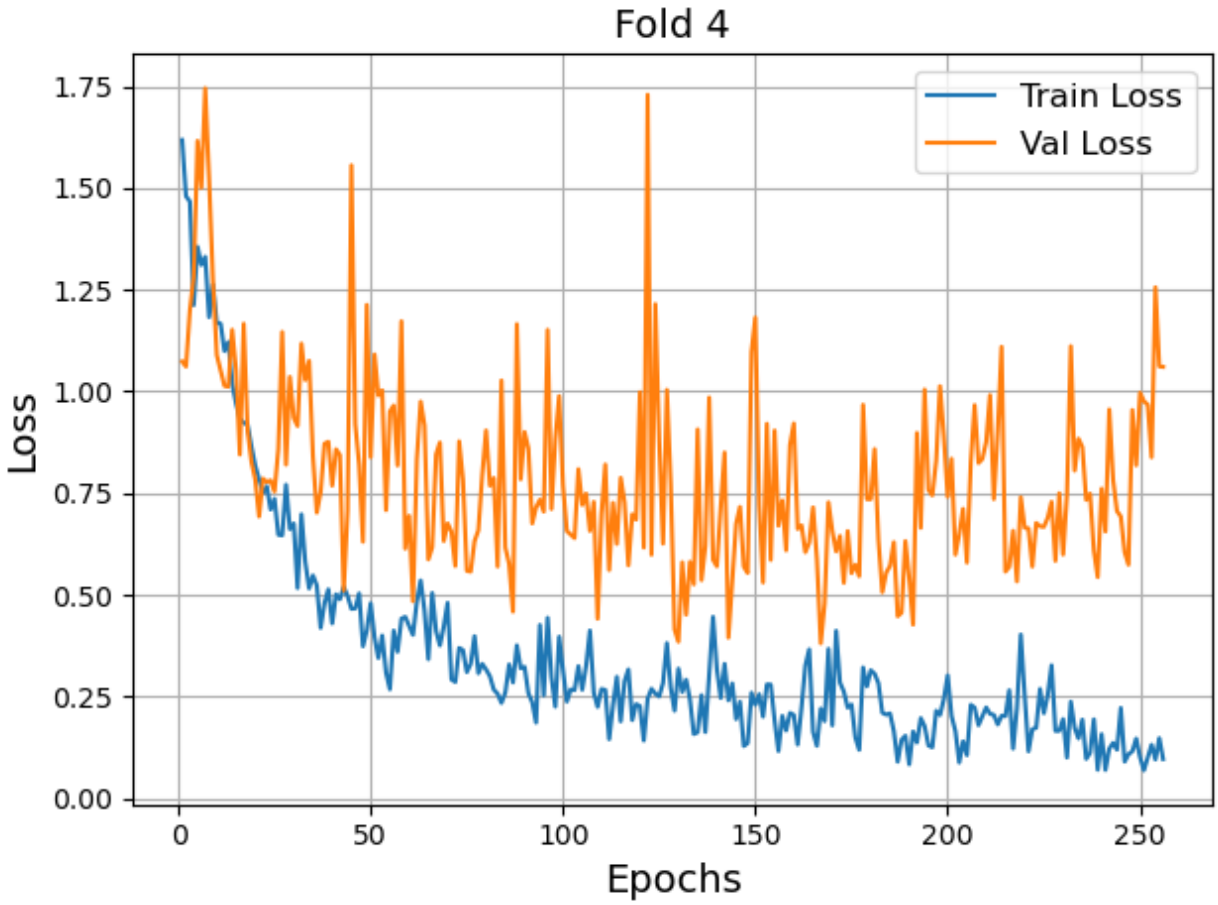
The following results for the second CNN-based model were achieved with a data augmentation level of 3 videos per subject. This augmentation helped improve model robustness and supported better learning on the small available dataset.

The loss curves for the second CNN-based model also demonstrate successful convergence, with both training and validation losses gradually decreasing over time as shown in **Figure 21**. While some fluctuations are present in the validation loss, the model achieves a stable

gap between training and validation losses, supporting its ability to generalize reasonably well across different folds.

Figure 21

Loss functions of Local StimNet



Compared to the StimNet model, this model exhibits slightly higher variability across folds, particularly with noticeable dips in Fold 2 and Fold 5. **Figure 22** shows performance metrics across folds, including Accuracy, Precision, Recall, and F1 Score.

Figure 22

Accuracy, Precision, Recall, and F1 Score plot across five folds of Local StimNet

Figure 23 presents the ROC curves across the five folds for the second model. The Area Under the Curve (AUC) values range from 0.87 to 0.92, slightly more variability than the StimNet model. While Folds 3 and 4 achieve strong AUC scores above 0.90, Folds 2 and 5 show a modest decrease. Overall, the model demonstrates good classification ability, but with less consistency across folds than StimNet.

Figure 23

ROC plots of Local StimNet across five folds

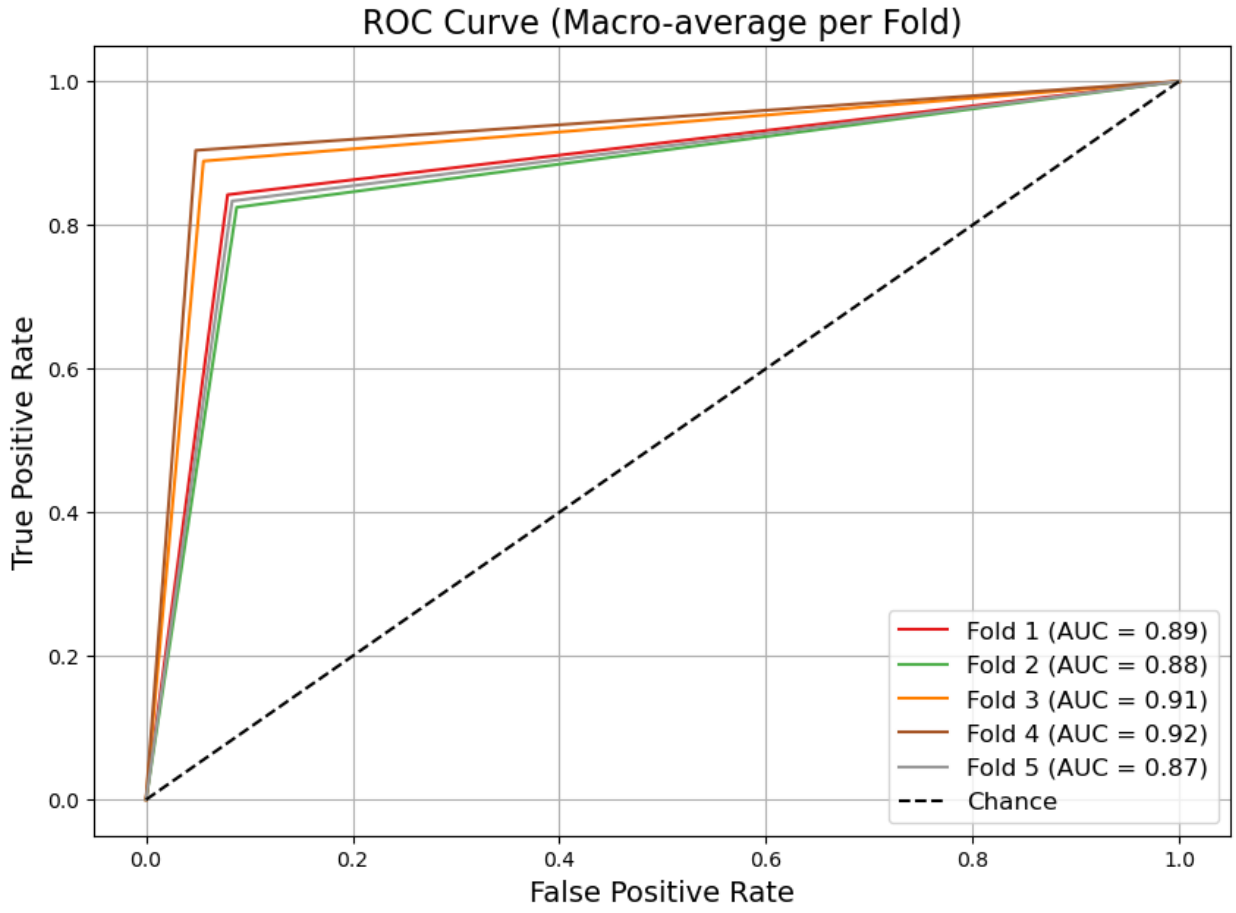
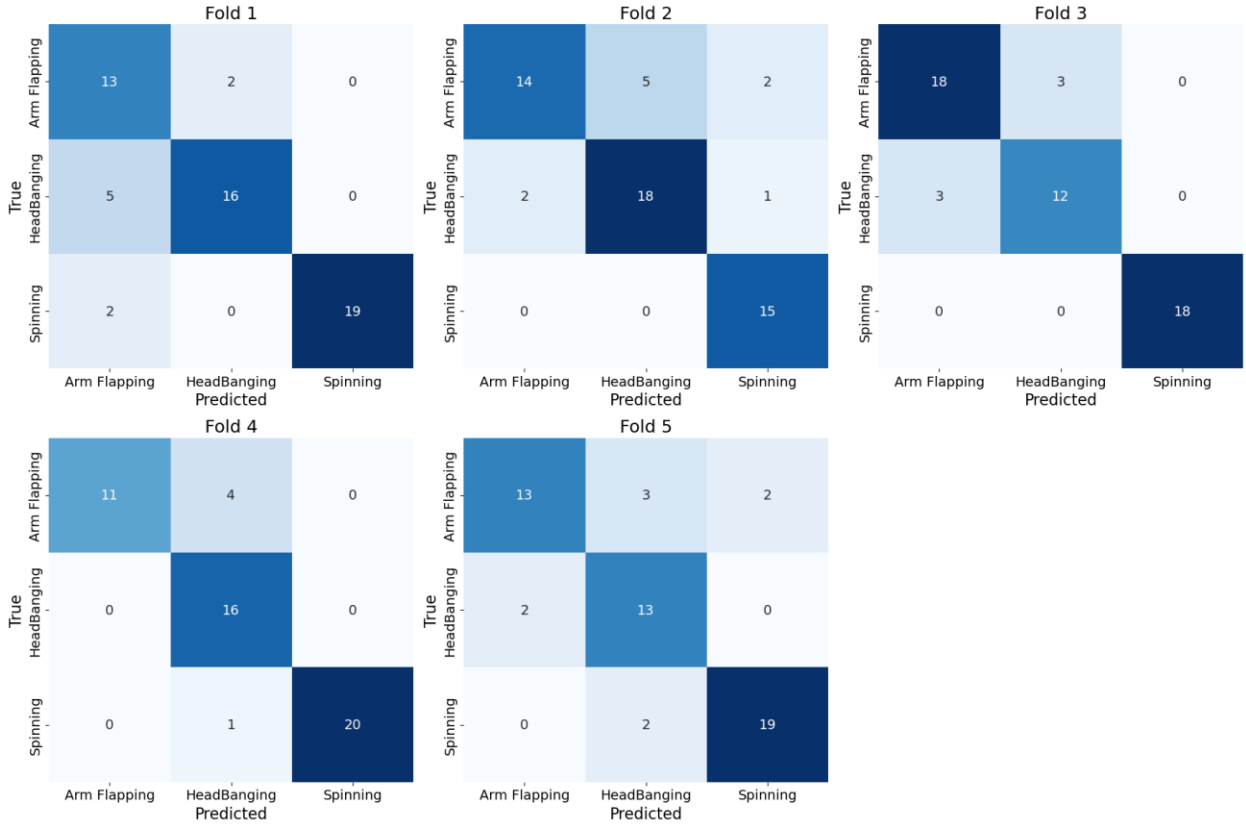


Figure 24 presents the confusion matrices across five folds for the second model. Overall, the model performs reasonably in classifying the three stimming actions: Arm Flapping, Head Banging, and Spinning. However, compared to StimNet, there is more frequent misclassification between Arm Flapping and Head Banging, particularly in Folds 2, 3, and 5. The model consistently performs well in recognizing spinning across all folds, with few errors.

Figure 24

Local StimNet confusion matrix for five folds



StimAuto: Autoencoder-based Model

The Autoencoder model was trained using a data augmentation strategy that expanded the dataset to 10 videos per subject. This higher augmentation level was necessary to improve the model's ability to learn latent features from limited input data.

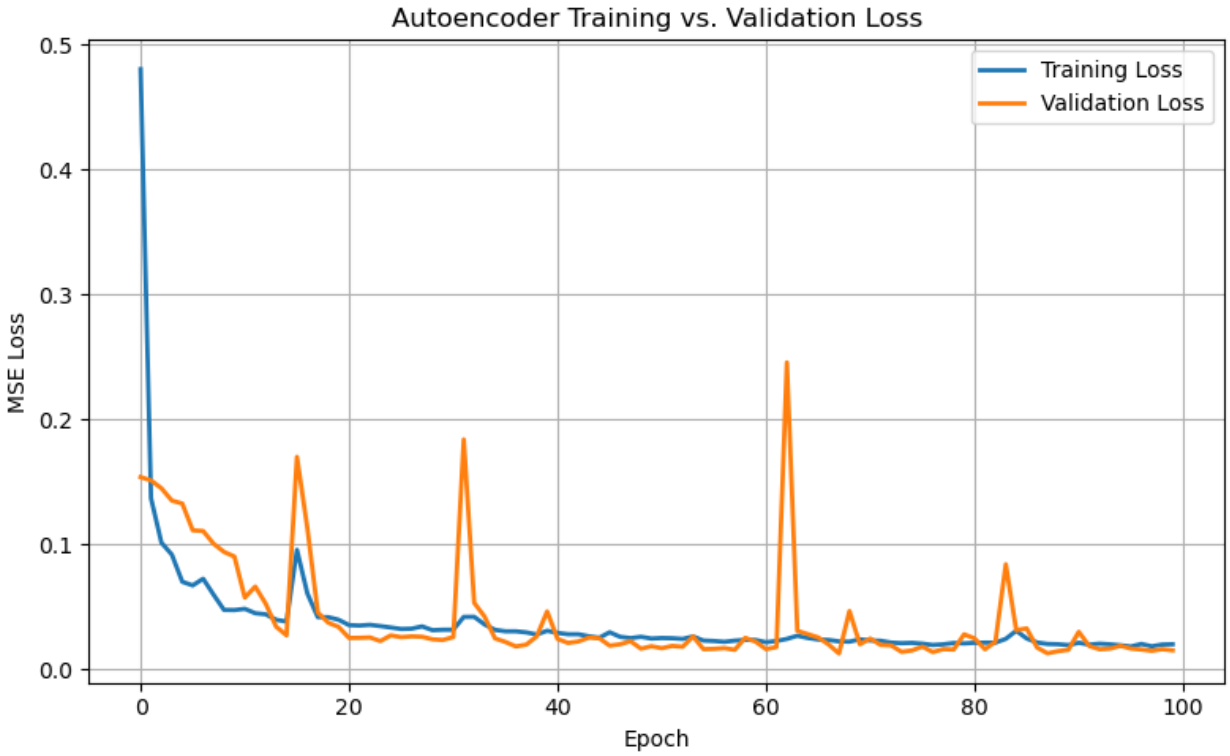
Autoencoder Training Loss (Reconstruction):

The Autoencoder model was trained using a reconstruction loss during the first phase to learn compressed latent representations of skeletal action sequences. **Figure 25** shows the Autoencoder pretraining stage's training and validation loss curves, optimized using mean squared error (MSE) loss. The steady decline and stabilization of both losses indicate that the

Autoencoder successfully learned to reconstruct the input skeletons, confirming effective latent space learning prior to classification.

Figure 25

Loss functions of Autoencoder

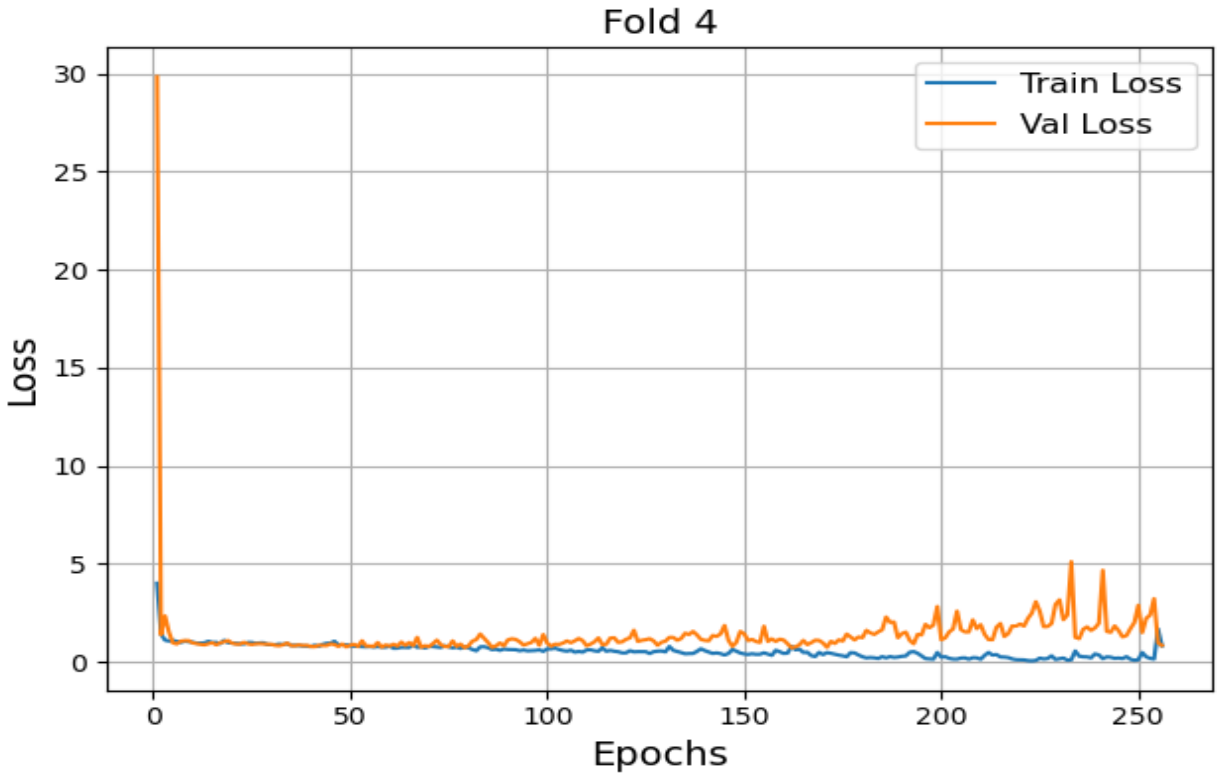


Autoencoder Encoder + MLP Classification Loss:

Following the Autoencoder pretraining, the encoder was used in combination with a multilayer perceptron (MLP) classifier for action recognition. **Figure 26** shows the training and validation loss curves for this second phase, where the model was optimized for classification using cross-entropy loss. The consistent downward trend in both training and validation losses demonstrates that the encoded features were effectively utilized for stimming action classification, achieving reasonable convergence.

Figure 26

Loss functions of StimAuto



The following figures present the evaluation results across five-fold cross-validation, including performance metrics, ROC curves, and confusion matrices of StimAuto.

Figure 27

Accuracy, Precision, Recall, and F1 Score plot across five folds of StimAuto

Figure 27 shows the performance metrics across folds, including Accuracy, Precision, Recall, and F1 Score. Compared to the previous models, the Autoencoder model exhibits higher variability across folds, with noticeable drops in Fold 2 and Fold 5. The metrics range approximately between 0.70 and 0.82. The peak performance is observed in Fold 3, but overall, the model shows less stability and lower scores than StimNet and the second model.

Figure 28

ROC plots of StimAuto across five folds

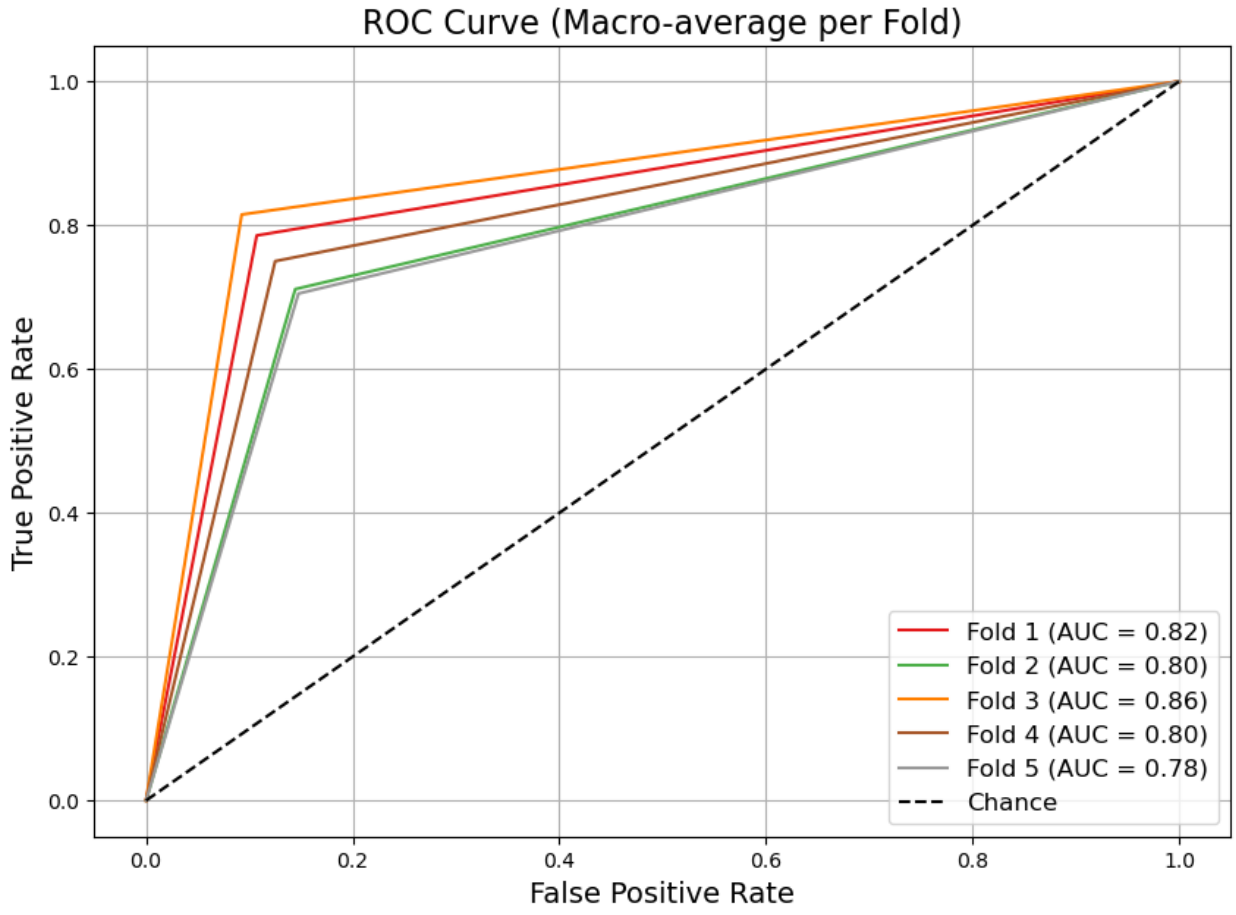


Figure 28 presents the ROC curves for each fold. The Area Under the Curve (AUC) values range from 0.78 to 0.86, which are lower than those achieved by the previous models. Folds 1, 2, and 4 show AUCs close to 0.80, while Fold 5 has the lowest AUC at 0.78. This indicates that while the Autoencoder model can distinguish between classes, its overall discriminatory ability is weaker and less consistent across folds than the other models.

Figure 29

StimAuto confusion matrix for five folds

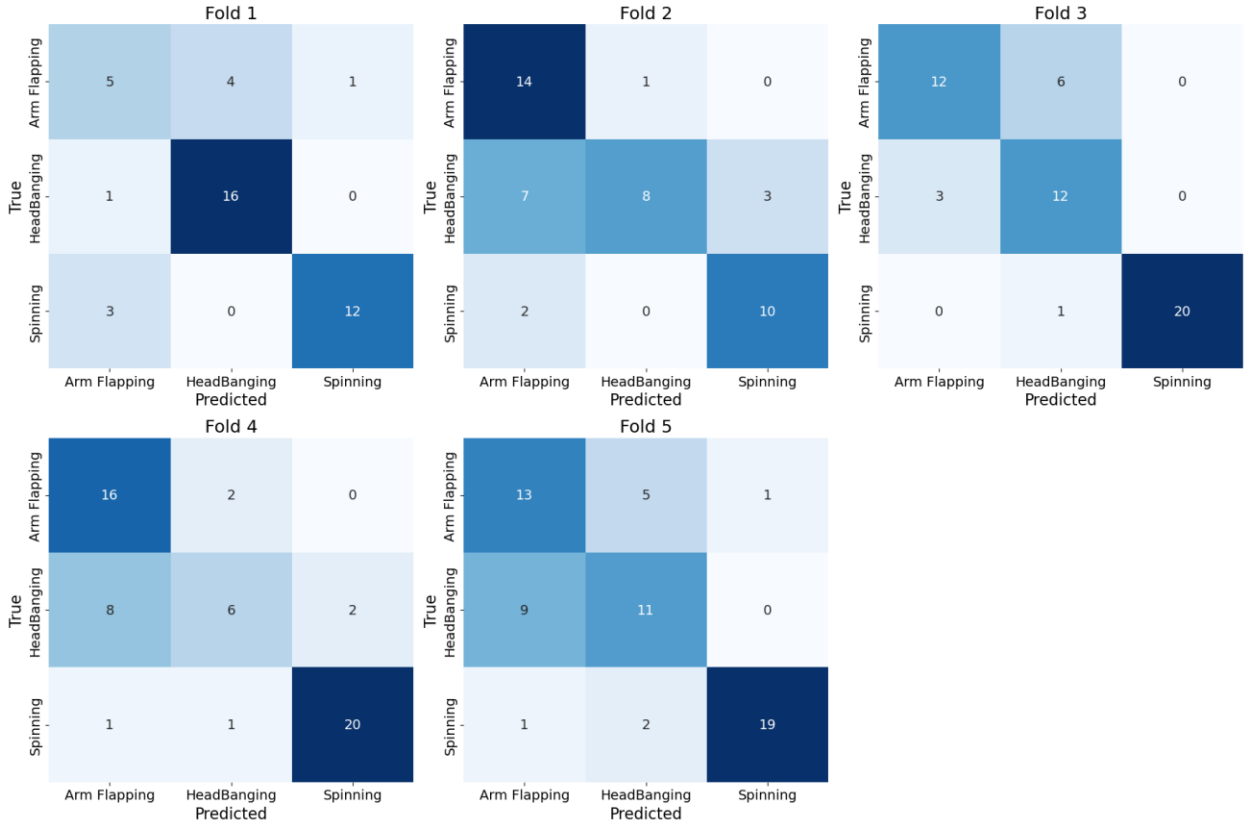


Figure 29 displays the confusion matrices for each fold. The Autoencoder model shows more frequent misclassifications, particularly between Arm Flapping and Head Banging. Although the model consistently accurately identifies Spinning actions, confusion between Arm Flapping and Head Banging is more pronounced across folds. These results suggest that while the model captures simpler spinning actions well, it struggles to differentiate more complex motion patterns.

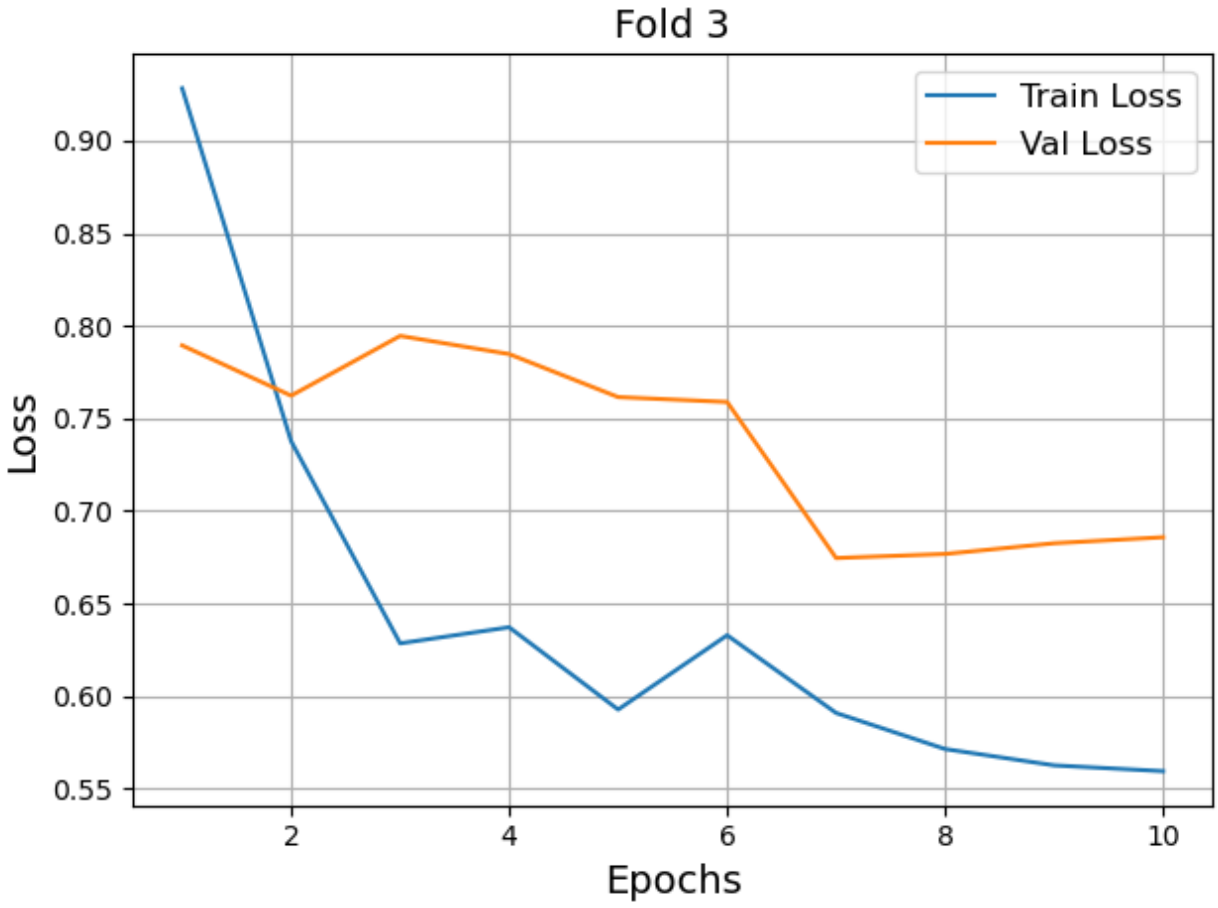
StimViT: Vision Transformer Model

The StimViT model results were obtained after applying a data augmentation level of 5 videos per subject. Adequate augmentation was essential for training the data-intensive Vision Transformer architecture effectively.

Figure 30 shows the training and validation loss curves for the StimViT model. The model, trained with five videos per subject, shows rapid convergence within a small number of epochs. Although slight fluctuations are present in the validation loss, both curves stabilize quickly, suggesting that the model efficiently captured global patterns from skeletal data. This rapid convergence highlights the StimViT model's strength in leveraging transformer-based architectures for the stimming action recognition task.

Figure 30

Loss functions of StimViT



The following figures present the evaluation results across five-fold cross-validation, highlighting the performance metrics, ROC curves, and confusion matrices.

Figure 31

Accuracy, Precision, Recall, and F1 Score plot across five folds of StimViT

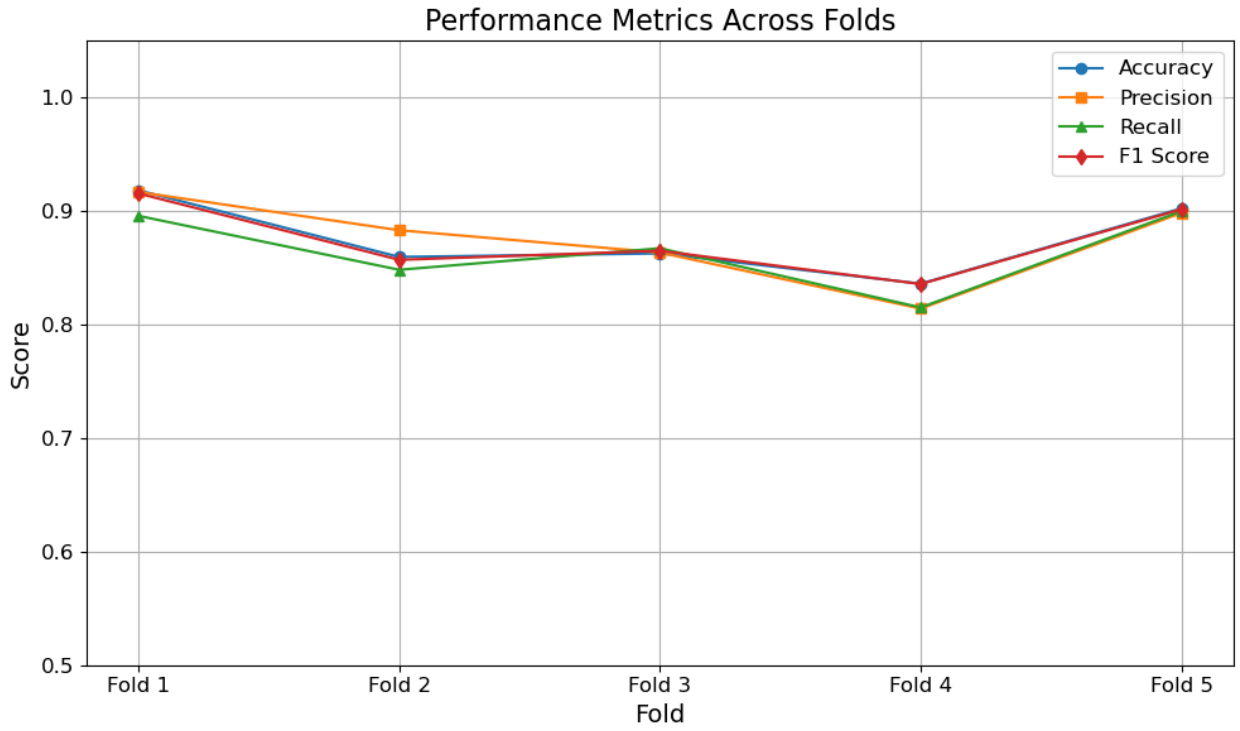


Figure 31 shows the performance metrics across folds for StimViT, including Accuracy, Precision, Recall, and F1 Score. The model consistently maintains high performance across folds, with metric values mostly ranging between 0.85 and 0.93. Although a slight dip is observed in Fold 4, the overall variation is smaller compared to the Autoencoder model. The close alignment of all four metrics across folds indicates balanced classification, effectively managing both false positives and false negatives.

Figure 32

ROC plots of StimViT across five folds

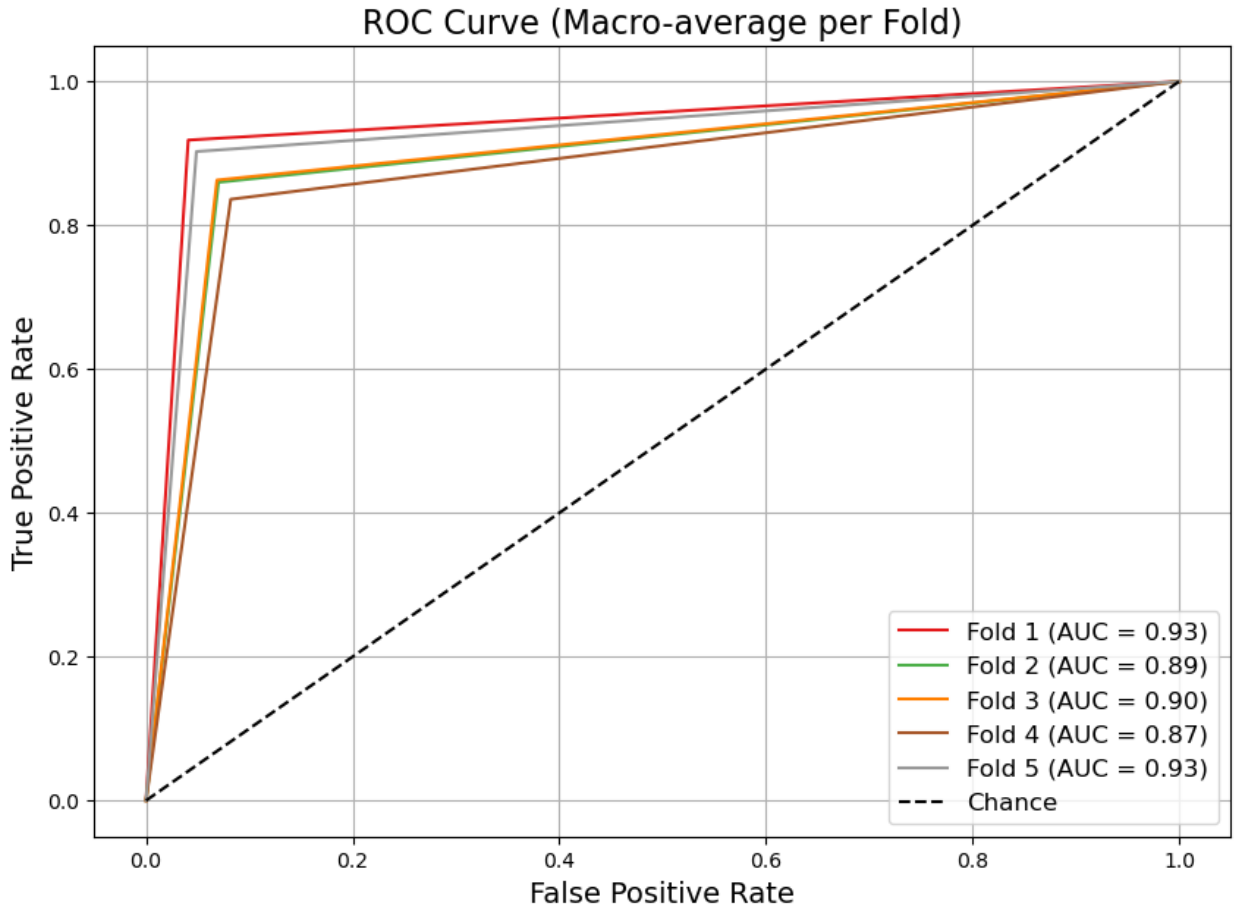


Figure 32 presents the ROC curves across the five folds. The Area Under the Curve (AUC) ranges from 0.87 to 0.93, with the highest performance observed in Fold 1 and Fold 5 (AUC = 0.93). The model demonstrates strong discriminatory ability, with minor variability between folds. StimViT achieves one of the highest and most consistent AUC performances compared to previous models, confirming its robustness in distinguishing stimming actions.

Figure 33

StimViT confusion matrix for five folds

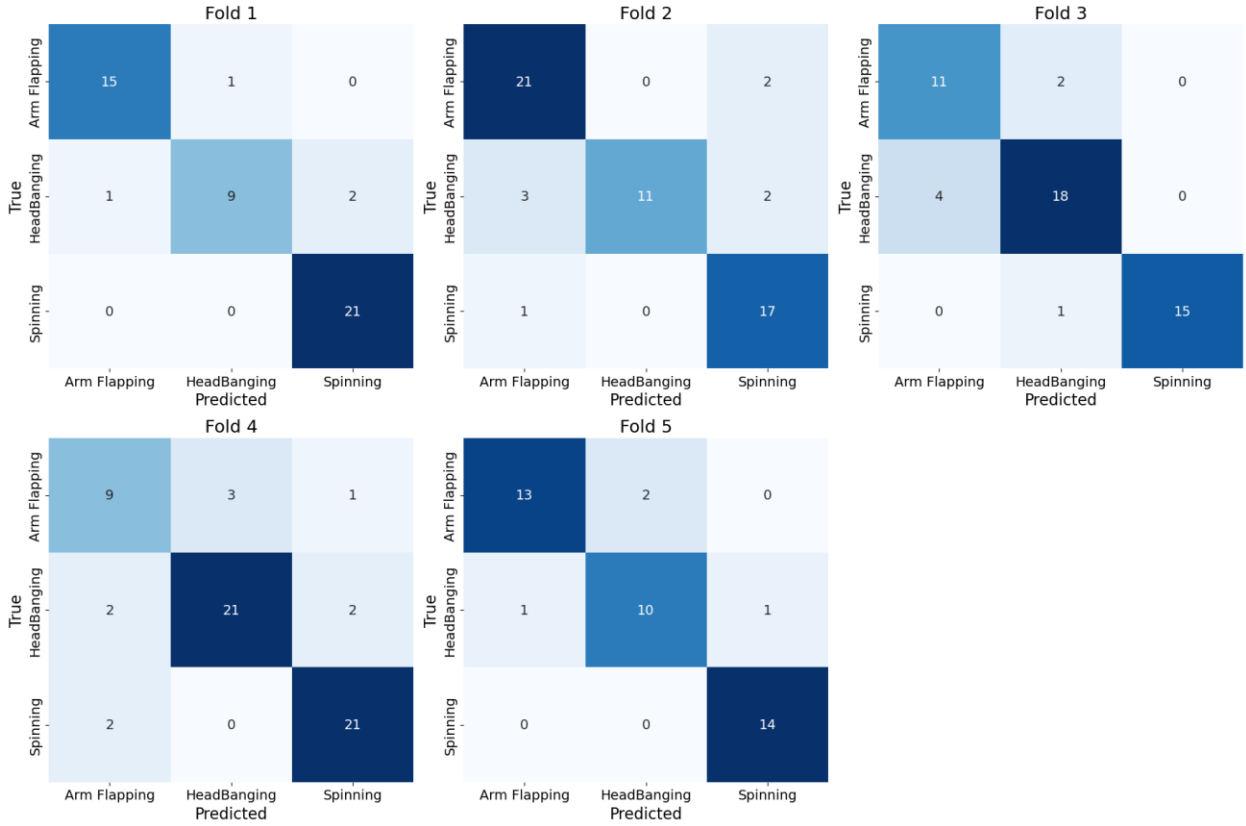


Figure 33 displays the confusion matrices for each fold. The StimViT model shows strong classification performance, particularly for Head-Banging and Spinning actions. Some confusion is observed between Arm Flapping and Head-Banging, but there are fewer misclassifications compared to the Autoencoder model. The overall matrices indicate that StimViT maintains reliable classification across all folds, successfully differentiating between the three stimming behaviors.

Overall Model comparisons

Figure 34 compares the accuracies using the box plot for four models, StimNet, Local StimNet, StimAuto, and StimViT, evaluated across five folds. StimNet and StimViT exhibit the highest median accuracies, both with minimal variation across folds, indicating strong

consistency and generalization. Local StimNet also achieves competitive accuracy but shows slightly more variability compared to StimNet and StimViT. In contrast, StimAuto demonstrates significantly lower median accuracy and greater spread across folds, reflecting higher variability and less stable performance. Overall, the box plot highlights that StimNet and StimViT provide the most reliable and accurate classification of stimming actions among the models studied.

Figure 34

Box plot: Five-fold Accuracies of Four Models

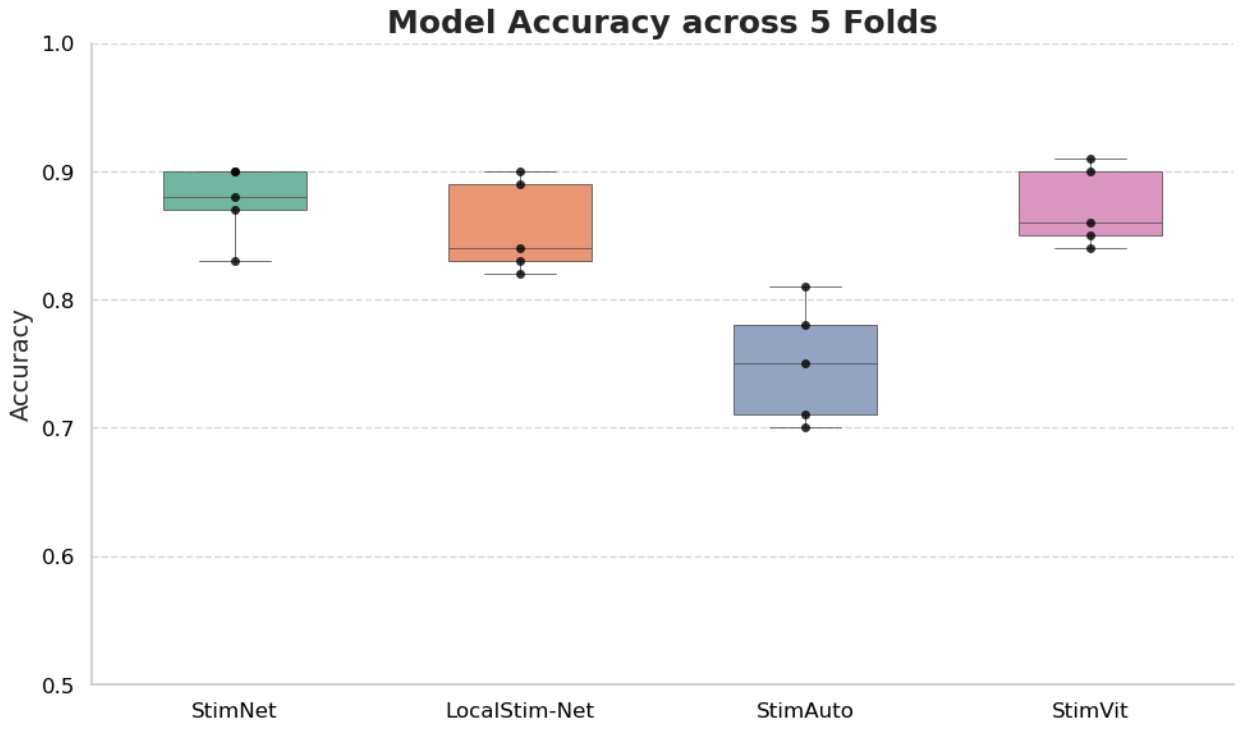


Table 4 summarizes the mean performance of all four models across five-fold cross-validation, reporting the Average Accuracy, F1 Score, and AUC values. StimNet achieves the highest overall scores, followed closely by StimViT, both demonstrating strong classification performance and robustness. Local Stim-Net also maintains competitive performance but slightly trails behind in all three metrics. Although StimAuto demonstrates convergence during training, it shows the lowest average performance across all metrics, highlighting its limitations for the

stimming action recognition task compared to CNN- and Transformer-based approaches. These results are consistent with the trends observed in the box plot comparison.

Table 4

Comparison of Results

S No	Model	Average Accuracy (5-fold)	Average F1 Score (5-fold)	Average AUC (5-fold)
1	StimNet	87.77%	87.69%	90.95%
2	Local StimNet	85.85%	85.85%	89.51%
3	StimAuto	75.28%	73.96%	76.43%
4	StimViT	87.2%	87.48%	90.41%

Table 5 compares existing studies on stimming action recognition with the models proposed in this research. Prior works have employed various machine learning and deep learning techniques such as Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs), and Temporal Convolutional Networks (TCNs). These studies generally used datasets like SSBD or private clinical datasets, achieving varying levels of classification accuracy. In contrast, our models, based on skeletal data extracted through MediaPipe and trained using 1D CNNs, Autoencoders, and Vision Transformers, achieve higher or comparable accuracies on the SSBD-extended dataset, demonstrating the effectiveness of deep learning-based skeletal action recognition for stimming behavior classification.

Table 5

Comparison of Our Models with Existing Literature.

Name of Paper	Methodology	Objective	Accuracy	Dataset	Classes
Rajagopalan et al.	SVM	Classify Stimming Actions	50.70%	SSBD	3
Negin et al.	YOLOv3 + HOF + MLP	Classify Stimming Actions	78%	ASD diagnostic centers (private)	3
Ali et al.	3D-CNN	Classify Stimming Actions	75.62%	SSBD	3
Wei et al.	LSTM, TCN, MS-TCN	Classify Stimming Actions	83%	SSBD (Extended)	3
Stim-Net	MediaPipe + 1D CNN	Classify Stimming Actions	**87.77%**	SSBD (Extended)	3
Local Stim-Net	MediaPipe + 1D CNN (Multi-Head)	Classify Stimming Actions	**85.54%**	SSBD (Extended)	3
StimAuto	MediaPipe + Autoencoder + MLP	Classify Stimming Actions	**75.28%**	SSBD (Extended)	3
StimViT	MediaPipe + Vision Transformer + MLP	Classify Stimming Actions	**87.48%**	SSBD (Extended)	3

Chapter 6: Conclusion and Future Work

This research advances the field of stimming action recognition in children with autism by introducing skeletal data as a primary input for classification. To the best of my knowledge, this is the first study to explore the use of skeletal representations for stimming behavior recognition, whereas prior works have relied entirely on raw RGB video. The findings suggest that skeletal data provides a more structured and noise-resistant input, leading to improved classification accuracy over conventional video-based methods. Additionally, this research is the first to integrate Vision Transformers (ViTs) and Autoencoders into stimming action recognition, moving beyond CNN and LSTM-based approaches. Fine-tuning the ViT-Base-Patch16-224 model on skeletal sequences allowed it to effectively learn both spatial and temporal dependencies in movement patterns, which improved its ability to distinguish different stimming behaviors with greater accuracy.

One of the contributions of this study is its approach to data augmentation. This study applies a set of transformations, such as scaling, rotation, and resizing, across all samples. These enhancements improve the model’s generalization, making it more adaptable to variations in camera angles and subject positioning. In addition to augmentation, preprocessing techniques were carefully designed to address common challenges such as background noise, variations in the child's orientation, and inconsistent camera perspectives. These refinements resulted in a more reliable dataset, ensuring that the training data better reflects real-world conditions.

To further update the dataset, a teammate collected additional video samples, leading to the formation of the ASDB (Autism Syndrome Behaviors) dataset. This dataset added one more class, Finger Flicking, to the existing three classes. Only three classes, Arm Flapping, Head Banging, and Spinning, are used from this dataset.

The results demonstrate that skeletal-based action recognition significantly enhances classification accuracy, making it a promising approach for real-time monitoring applications in therapeutic and home settings.

Suggestions

While this study establishes a strong foundation for automated stimming action recognition, there are several areas where further improvements can be made. One important direction is refining the data preprocessing pipeline by integrating an automated method for detecting and cropping the child before extracting skeletal data. This additional step would help eliminate irrelevant background noise and improve the consistency of skeletal representations.

Expanding the dataset, particularly for the Hand Action class, is also crucial, as this category currently has limited subjects. Increasing the dataset size for underrepresented classes would enhance the model's ability to generalize across different individuals and environments.

Further exploration of advanced deep learning architectures like pre-trained autoencoders could also improve performance. Additionally, optimizing the model for real-time deployment on edge devices would be a valuable step toward making this technology more accessible for practical applications in clinical and home environments.

Given the challenges of acquiring large, annotated datasets, self-supervised and semi-supervised learning techniques could be explored to reduce dependence on fully labeled data while improving model generalization. Techniques such as self-supervised learning or synthetic data generation using generative adversarial networks (GANs) may help improve model generalization and enable more effective training with limited real-world samples.

Future research can address these challenges and further advance automated stimming action recognition, making it a more practical and scalable solution for assisting caregivers and clinicians in monitoring children with autism.

References

Ali, A., Negin, F., Thümmler, S., & Bremond, F. (2022). Video-based Behavior Understanding of Children for Objective Diagnosis of Autism: *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 475–484. https://doi.org/10.5220/0010839200003124

Bishop, S. L., Richler, J., & Lord, C. (2006, October). Association Between Restricted and Repetitive Behaviors and Nonverbal IQ in Children with Autism Spectrum Disorders. *Child Neuropsychology*, *12*(4–5), 247–267. https://doi.org/10.1080/09297040600630288

Buescher, A. V. S., Cidav, Z., Knapp, M., & Mandell, D. S. (2014, August 1). Costs of Autism Spectrum Disorders in the United Kingdom and the United States. *JAMA Pediatrics*, *168*(8), 721. https://doi.org/10.1001/jamapediatrics.2014.210

DeWeerdt, S. (2023, March 10). *Repetitive behaviors and 'stimming' in autism, explained*. Spectrum | Autism Research News. https://doi.org/10.53053/ERTG7729

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* (arXiv:2010.11929). arXiv. https://doi.org/10.48550/arXiv.2010.11929

Do, J., & Kim, M. (2024). *SkateFormer: Skeletal-Temporal Transformer for Human Action Recognition* (arXiv:2403.09508). arXiv. https://doi.org/10.48550/arXiv.2403.09508

Han, H., Zeng, H., Kuang, L., Han, X., & Xue, H. (2024). A human activity recognition method based on Vision Transformer. *Scientific Reports*, *14*, 15310. https://doi.org/10.1038/s41598-024-65850-3

Hinton, G. E., & Salakhutdinov, R. R. (2006b). Reducing the Dimensionality of Data with Neural Networks. *Science (New York, N.Y.)*, *313*, 504–507. https://doi.org/10.1126/science.1127647

Ibh, M., Grasshof, S., Witzner, D., & Madeleine, P. (2023). TemPose: A new skeleton-based transformer model designed for fine-grained motion recognition in badminton. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 5199–5208. https://doi.org/10.1109/CVPRW59228.2023.00548

Karadag, O. O. (2024). SkelVIT: Consensus of Vision Transformers for a Lightweight Skeleton-Based Action Recognition System. *Signal, Image and Video Processing*, *18*(8–9), 5619–5629. https://doi.org/10.1007/s11760-024-03259-1

Ke, Q., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2017). *A new representation of skeleton sequences for 3D action recognition*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3288–3297). https://doi.org/10.1109/CVPR.2017.351

LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). https://doi.org/10.1038/nature14539

Leo, M., Bernava, G. M., Carcagnì, P., & Distante, C. (2022). Video-Based Automatic Baby Motion Analysis for Early Neurological Disorder Diagnosis: State of the Art and Future Directions. *Sensors*, *22*(3), Article 3. https://doi.org/10.3390/s22030866

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). *MediaPipe: A Framework for Building Perception Pipelines* (arXiv:1906.08172). arXiv. https://doi.org/10.48550/arXiv.1906.08172

Maenner, M. J., Warren, Z., Williams, A. R., Amoakohene, E., Bakian, A. V., Bilder, D. A., Durkin, M. S., Fitzgerald, R. T., Furnier, S. M., Hughes, M. M., Ladd-Acosta, C. M., McArthur, D., Pas, E. T., Salinas, A., Vehorn, A., Williams, S., Esler, A., Grzybowski, A., Hall-Lande, J., . . . Shaw, K. A. (2023, March 24). Prevalence and Characteristics of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2020. *MMWR. Surveillance Summaries*, *72*(2), 1–14. https://doi.org/10.15585/mmwr.ss7202a1

Mokhtari, N., Nédélec, A., & Loor, P. (2022). *Human Activity Recognition: A Spatio-temporal Image Encoding of 3D Skeleton Data for Online Action Detection* (p. 455). https://doi.org/10.5220/0010835800003124

Leekam, S., Tandos, J., McConachie, H., Meins, E., Parkinson, K., Wright, C., Turner, M., Arnott, B., Vittorini, L., & Couteur, A. L. (2007, October 3). Repetitive behaviours in typically developing 2-year-olds. *Journal of Child Psychology and Psychiatry*, *48*(11), 1131–1138. https://doi.org/10.1111/j.1469-7610.2007.01778.x

Negin, F., Ozyer, B., Agahian, S., Kacdioglu, S., & Ozyer, G. T. (2021). Vision-assisted recognition of stereotype behaviors for early diagnosis of Autism Spectrum Disorders. *Neurocomputing*, *446*, 145–155. https://doi.org/10.1016/j.neucom.2021.03.004

Rajagopalan, S. S., Dhall, A., & Goecke, R. (2013). Self-Stimulatory Behaviours in the Wild for Autism Diagnosis. *2013 IEEE International Conference on Computer Vision Workshops*, 755–761. https://doi.org/10.1109/ICCVW.2013.103

Serna-Aguilera, M., Nguyen, X. B., Singh, A., Rockers, L., Park, S., Neely, L., Seo, H., & Luu, K. (2024). *Video-Based Autism Detection with Deep Learning* (arXiv:2402.16774). arXiv. http://arxiv.org/abs/2402.16774

Sharma, A., & Tanwar, P. (2024). Model for autism disorder detection using deep learning. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *13*, 391. https://doi.org/10.11591/ijai.v13.i1.pp391-398

Tariq, Q., Daniels, J., Schwartz, J. N., Washington, P., Kalantarian, H., & Wall, D. P. (2018). Mobile detection of autism through machine learning on home video: A development and prospective validation study. *PLOS Medicine*, *15*(11), e1002705. https://doi.org/10.1371/journal.pmed.1002705

Vrskova, R., Kamencay, P., Hudec, R., & Sykora, P. (2023). A New Deep-Learning Method for Human Activity Recognition. *Sensors*, *23*(5), Article 5. https://doi.org/10.3390/s23052816

Washington, P., Kline, A., Mutlu, O. C., Leblanc, E., Hou, C., Stockham, N., Paskov, K., Chrisman, B., & Wall, D. (2021). *Activity Recognition with Moving Cameras and Few Training Examples: Applications for Detection of Autism-Related Headbanging* (p. 7). https://doi.org/10.1145/3411763.3451701

Wei, P., Ahmedt-Aristizabal, D., Gammulle, H., Denman, S., & Armin, M. A. (2023). Vision-based activity recognition in children with autism-related behaviors. *Heliyon*, *9*(6), e16763. https://doi.org/10.1016/j.heliyon.2023.e16763

Wu, K., Peng, H., Chen, M., Fu, J., & Chao, H. (2021). Rethinking and Improving Relative Position Encoding for Vision Transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10013–10021. https://doi.org/10.1109/ICCV48922.2021.00988

Yan, S., Xiong, Y., & Lin, D. (2018). *Spatial temporal graph convolutional networks for skeleton-based action recognition*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.12328

Yang, F., Wu, Y., Sakti, S., & Nakamura, S. (2020). Make Skeleton-based Action Recognition Model Smaller, Faster and Better. *Proceedings of the 1st ACM International Conference on Multimedia in Asia*, 1–6. https://doi.org/10.1145/3338533.3366569

Zuckerman, K. E., Lindly, O. J., & Sinche, B. K. (2015, June). Parental Concerns, Provider Response, and Timeliness of Autism Spectrum Disorder Diagnosis. *The Journal of Pediatrics*, *166*(6), 1431-1439.e1. https://doi.org/10.1016/j.jpeds.2015.03.007