الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة غرداية
University of Ghardaia

N° d'enregistrement
/…../…../…../…../…..

كلية العلوم والتكنولوجيا
Faculty of Science and Technology
قسم الرياضيات و الإعلام الآلي
Département of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master

Domaine : Mathematics and Computer Science
Filière : Computer Science
Spécialité : Intelligent Systems for Knowledge Extraction

Topic

ADeep Learning based Autism Stimming Detection From Video : Skeletal Motion Analysis

Présenté par :
Youb Bassaid
Latreche Brahim

Soutenue publiquement le June, 2026

Devant le jury composé de :

ML.Fkair	MCA	**Univ.Ghardaia**	**President**
A.Bouchekouf	MAA	**Univ.Ghardaia**	**Examiner**
M.Adjila	Professeur	**Univ.Ghardaia**	**Examiner**
N. Brahim	MAA	**Univ.Ghardaia**	**Supervisor**

Année universitaire 2025/2026

Dedication

May this modest work reflect my deepest gratitude to all the wonderful people who have lit up my journey.

To my dear parents, who have sacrificed everything for me, and whose prayers and love have carried me this far.

To my brothers and sisters, my lifelong companions, and to all my extended family — my paternal and maternal uncles, aunts, and cousins — for their warm presence and constant encouragement.

To my teachers and supervisors, who passed on to me far more than knowledge: rigor, passion, and humility.

To my sincere friends and fellow travelers, who shared in my hardships and my joys.

And to all those who have loved me unconditionally, simply.

Bassaid YOUB

Dedication

Praise be to God, by whose grace good deeds are accomplished, and by whose guidance goals and objectives are achieved. I praise Him, the Exalted, for His favor in facilitating the completion of this research, crowning with it years of effort and pursuit, and long patience in the sanctuary of science and knowledge.

To the source of generosity and grace; my dear father, who was my support and pillar, and my precious mother, who was my prayers and care; the lights of my path and the guides of my steps. And to all my family and relatives, who strengthened me and were a fortress for me throughout my journey.

To the esteemed professor supervising this work, and to my distinguished professors in this scientific institution; I dedicate to you the fruit of this effort in recognition of your academic merit, and in appreciation of your sound guidance that illuminated for me the paths of research and analysis.

To my fellow students with whom I shared the passion for science and the study halls, and to my loyal friends from outside the university walls, who were my best support and encouragement, and who instilled hope in me at every stage.

To our esteemed university, its leadership and administration, and to all its dedicated staff and employees: Thank you for your commitment to overcoming obstacles and fostering the nurturing academic environment that enabled us to produce this scholarly work.

I dedicate this humble effort to everyone who laid the foundation for my intellectual future.

Latreche Brahim

Acknowledgments

Praise be to God Almighty, who provided us with strength and patience, and bestowed upon us health and well-being to overcome obstacles and difficulties, until this humble effort was crowned with completeness and perfection.

We extend our deepest thanks and gratitude to our esteemed supervisor, Ms. BRAHIM Nacera, and our co-supervisor, Mr. KHARROUBI Ahmed, who showered us with their valuable advice and guidance, and generously provided us with their sound scientific care and open hearts throughout the preparation of this research.

We also wish to express our heartfelt gratitude and special thanks to all the esteemed professors and teachers who taught us throughout our academic journey; we owe them a debt of gratitude for the knowledge and guidance they so generously shared to illuminate our path to success.

We extend our sincere thanks and appreciation to the esteemed professors, members of the discussion committee, for their kindness in accepting to evaluate this work and enriching it with their sound observations and valuable scientific corrections.

In conclusion, we extend our deepest gratitude to our dear parents for their unwavering support and sincere prayers, and to our families, our classmates, and all our loyal friends who encouraged us and were a great support to us until we reached this goal.

ملخص

لا يزال الكشف المبكر والموضوعي عن اضطراب طيف التوحد تحدياً سريرياً بالغ الأهمية، إذ تتسم أدوات التشخيص التقليدية بالتكلفة الزمنية والمادية العالية، والذاتية في التقييم، وكثيراً ما تُطبَّق في مراحل متأخرة تفوت معها فرصة الاستفادة القصوى من مرونة الشبكة العصبية. تهدف هذه الأطروحة إلى الكشف الآلي عن السلوكيات الحركية النمطية المعروفة بـ**السلوك التحفيزي الذاتي** من تسجيلات فيديو طبيعية غير مقيّدة، بهدف تقديم أداة فحص غير تدخلية وقابلة للتطوير. لتحقيق ذلك، جُمِعت مجموعة بيانات مركّبة من مستودعات عامة راسخة وتسجيلات تكميلية من منصات التواصل الاجتماعي، وأُعيد تشريحها بدقة على مستوى الإطار الواحد لتغطية أربع فئات سلوكية: رفرفة الذراعين، وضرب الرأس، والدوران، والحركة الطبيعية. وبعد أن كشف نموذج أولي عن خلل في تسرّب البيانات أدى إلى تضخيم مصطنع للدقة، جرت إعادة تصميم منهجية كاملة مبنية على تقسيم صارم على مستوى الفيديو المصدر. تستعيض المنهجية النهائية عن ميزات البكسل بتمثيل هيكلي يصون الخصوصية، مستخرَج بواسطة YOLOv8x-pose، يُشتقّ منه متجه ميزات بيوميكانيكية لكل إطار يشمل: زوايا المفاصل، والسرعات الزاوية، والطاقة الحركية، وديناميكيات مركز الثقل، ومؤشر الترددية الحركية، ويُغذّي بنيتي دمج مزدوج هما: BiLSTM+PoseC3D وST-GCN+GRU. أثبتت كلتا البنيتين قدرةً عاليةً على التعميم على مجموعة اختبار خارجية مستقلة تماماً، إذ حقّق أفضل إعداد دقةً تجاوزت ثمانين بالمئة ومساحةً تحت منحنى ROC فاقت 0.95. وقد تميّز سلوك ضرب الرأس بأعلى موثوقية في الاكتشاف، في حين مثّلت الفئة الطبيعية أصعب الفئات نظراً للتشابه الحركي مع السلوكيات التحفيزية منخفضة الشدة. تُؤكّد هذه النتائج أن التعلم العميق القائم على الوضعية الهيكلية والمُعزَّز بالمعطيات البيوميكانيكية يُمثّل نهجاً قابلاً للتطبيق في الكشف الآلي عن السلوك التحفيزي الذاتي، فيما تُعدّ توسعة قواعد البيانات والتعلم الضعيف الإشراف أبرز المسارات الواعدة للأبحاث المستقبلية.

الكلمات المفتاحية: اضطراب طيف التوحد، اكتشاف السلوك التحفيزي الذاتي، تقدير الوضعية الهيكلية، YOLOv8، BiLSTM، ST-GCN، دمج التدفقين المزدوج، الميزات البيوميكانيكية، تصنيف الفيديو، التعلم العميق.

Abstract

Early and objective detection of Autism Spectrum Disorder (ASD) remains a critical clinical challenge, as traditional diagnostic instruments are resource-intensive, subjective, and often administered too late to exploit peak neural plasticity. This thesis investigates the automated recognition of stereotypical self-stimulatory behaviors—commonly referred to as *stimming*—from unconstrained video recordings, with the goal of providing a non-invasive, scalable screening tool. To this end, a composite dataset was assembled from established public repositories and supplementary social media recordings, re-annotated at frame-level precision across four behavioral classes: Arm Flapping, Headbanging, Spinning, and Neutral. After an initial prototype exposed a critical data-leakage flaw that artificially inflated accuracy, a complete methodological redesign was undertaken based on rigorous source-video-level data partitioning. The final approach replaces pixel-level features with a privacy-preserving skeletal representation extracted by YOLOv8x-pose, from which a per-frame biomechanical feature vector is computed—encoding joint angles, angular velocities, kinetic energy, center-of-gravity dynamics, and a periodicity proxy—and feeds two dual-stream fusion architectures: **BiLSTM+PoseC3D** and **ST-GCN+GRU**. Both architectures demonstrated strong generalization on a fully independent external test set, with the best configuration achieving over eighty percent accuracy and an area under the ROC curve exceeding 0.95. Headbanging proved the most reliably classified behavior, while the Neutral class remained the most challenging due to kinematic overlap with low-intensity stimming. These results confirm that pose-based, biomechanically informed deep learning is a viable and principled approach to automated stimming detection, with dataset expansion and weakly-supervised learning identified as the most promising avenues for future improvement.

Keywords: Autism Spectrum Disorder, stimming detection, human pose estimation, YOLOv8, BiLSTM, ST-GCN, dual-stream fusion, biomechanical features, video classification, deep learning.

Résumé

La détection précoce et objective des troubles du spectre autistique (TSA) demeure un défi clinique majeur : les outils diagnostiques traditionnels sont coûteux en ressources, subjectifs et généralement administrés trop tardivement pour bénéficier pleinement de la plasticité neuronale. Cette thèse vise la reconnaissance automatique des comportements stéréotypés d'auto-stimulation— communément appelés *stimming*—à partir d'enregistrements vidéo non contraints, dans le but de fournir un outil de dépistage non invasif et extensible. À cet effet, un jeu de données composite a été constitué à partir de dépôts publics établis et d'enregistrements complémentaires issus des réseaux sociaux, puis ré-annoté à la précision de l'image près selon quatre classes comportementales : battement des bras, cognement de la tête, rotation sur soi et mouvement neutre. Après qu'un prototype initial a révélé une fuite de données critique faussant artificiellement les résultats, une refonte méthodologique complète a été entreprise, fondée sur un partitionnement rigoureux au niveau de la vidéo source. L'approche finale substitue les caractéristiques au niveau pixel par une représentation squelettique préservant la vie privée, extraite par YOLOv8x-pose, à partir de laquelle un vecteur de caractéristiques biomécaniques par image est calculé—encodant angles articulaires, vitesses angulaires, énergie cinétique, dynamique du centre de gravité et un indicateur de périodicité—et alimente deux architectures à double flux : **BiLSTM+PoseC3D** et **ST-GCN+GRU**. Les deux architectures ont démontré une bonne généralisation sur un jeu de test externe entièrement indépendant, la meilleure configuration atteignant plus de quatre-vingts pour cent de précision et une aire sous la courbe ROC supérieure à 0,95. Le cognement de la tête s'est avéré le comportement le plus fiablement classé, tandis que la classe neutre est restée la plus difficile en raison du chevauchement cinématique avec les stimmings de faible intensité. Ces résultats confirment que l'apprentissage profond basé sur la pose et informé par la biomécanique constitue une approche viable pour la détection automatisée du stimming, l'extension des données et l'apprentissage faiblement supervisé étant identifiés comme les pistes les plus prometteuses pour les travaux futurs.

Mots-clés : trouble du spectre autistique, détection du stimming, estimation de pose squelettique, YOLOv8, BiLSTM, ST-GCN, fusion double flux, caractéristiques biomécaniques, classification vidéo, apprentissage profond.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADI-R	Autism Diagnostic Interview—Revised
ADOS-2	Autism Diagnostic Observation Schedule, Second Edition
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASD	Autism Spectrum Disorder
AU	Action Unit
BiLSTM	Bidirectional Long Short-Term Memory
BPTT	Backpropagation Through Time
CARS	Childhood Autism Rating Scale
CNN	Convolutional Neural Network
CoG	Center of Gravity
DSM-5-TR	Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition, Text Revision
EDR	Early Detection Rate
EEG	Electroencephalogram
ESSBD	Extended Self-Stimulatory Behavior Dataset
FC	Fully Connected
fMRI	Functional Magnetic Resonance Imaging
FPN	Feature Pyramid Network
GDPR	General Data Protection Regulation
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
HPE	Human Pose Estimation
I3D	Inflated 3D Convolutional Network
ICD-11	International Classification of Diseases, Eleventh Revision
IQ	Intelligence Quotient
k-NN	k-Nearest Neighbors

LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
MBH	Motion Boundary Histogram
MLP	Multi-Layer Perceptron
MS COCO	Microsoft Common Objects in Context
PANet	Path Aggregation Network
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
ROC-AUC	Receiver Operating Characteristic - Area Under the Curve
ROI	Region of Interest
RRB	Restricted and Repetitive Behaviors
SHAP	Shapley Additive exPlanations
SSBD	Self-Stimulatory Behavior Dataset
ST-GCN	Spatial Temporal Graph Convolutional Network
SVM	Support Vector Machine
Tanh	Hyperbolic Tangent
VRAM	Video Random Access Memory
WHO	World Health Organization
XAI	Explainable Artificial Intelligence
YOLO	You Only Look Once

INTRODUCTION

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition affecting an estimated one in every 36 children worldwide, characterised by persistent deficits in social communication and by restricted, repetitive patterns of behaviour [1]. Among the most clinically observable manifestations of these repetitive patterns are self-stimulatory behaviours— commonly referred to as *stimming*—which encompass motor stereotypies such as arm flapping, headbanging, body rocking, and spinning. Although stimming is not in itself diagnostic, its presence, frequency, and intensity constitute significant clinical signals that trained practitioners use to characterise the severity and nature of the disorder [1].

Early diagnosis is widely regarded as the single most consequential factor in long-term outcomes for individuals with ASD. Meta-analytic evidence indicates that behavioural interventions initiated before the age of three yield substantially greater cognitive and adaptive gains than those begun later, with reported effect sizes of $d = 0.74$ for cognitive outcomes [2]. Yet the median age at diagnosis in many healthcare systems remains between three and five years—well beyond this critical developmental window. The primary bottlenecks are structural: gold-standard diagnostic protocols such as the Autism Diagnostic Observation Schedule (ADOS-2) require trained clinicians, controlled environments, and sessions lasting 45 to 60 minutes. In many regions, specialist waiting lists extend beyond a year, and access is severely limited in rural and low-income settings [3].

The convergence of advances in deep learning, computer vision, and human pose estimation has opened a fundamentally different diagnostic pathway. Video recordings of children engaged in natural play or daily activities can now be analysed automatically, without clinical infrastructure, invasive sensors, or specialised facilities. This approach is non-invasive, privacy-preserving when pose-based representations are used, and scalable to populations that would never otherwise reach a specialist. The central technical challenge is translating subtle, variable, and context-dependent behavioural signals into reliable automated classifications—a problem that classical machine learning and handcrafted features have proven insufficient to solve, but that modern spatiotemporal deep architectures are increasingly well-equipped to address.

This thesis presents the design, implementation, and evaluation of an automated system for detecting stereotypical self-stimulatory behaviours from video recordings, targeting four behavioural categories: Arm Flapping, Headbanging, Spinning, and Neutral (non-stimulatory movement). The system follows a two-solution development trajectory. The first solution ,

based on multi-backbone CNN feature extraction combined with a BiLSTM temporal model, revealed a critical data-leakage vulnerability in the experimental pipeline and established empirically that pixel-level features are insufficient for robust generalisation under real-world recording conditions. The second solution pivots to a skeletal representation extracted by YOLOv8x-pose, from which a 35-dimensional hand-crafted biomechanical feature vector is derived per frame. Two dual-stream fusion architectures are then evaluated: BiLSTM+PoseC3D, which pairs a 3D convolutional heatmap stream with a Bidirectional LSTM biomechanical stream, and ST-GCN+GRU, which replaces the 3D CNN backbone with a Spatial-Temporal Graph Convolutional Network.

The thesis is organised as follows:

Chapter 1 — Theoretical Foundations and Key Concepts introduces the clinical and computational background necessary to contextualise the work. It covers the diagnostic criteria and behavioural biomarkers of ASD, the argument for early automated screening, and the deep learning building blocks underpinning the proposed system: Convolutional Neural Networks, recurrent architectures (RNN, LSTM, GRU, BiLSTM), 3D convolutional networks, dual-stream fusion, and human pose estimation via YOLOv8x-pose.

Chapter 2 — State of the Art provides a structured review of existing approaches to automated ASD behaviour detection. It surveys classical handcrafted-feature methods, CNN-based classifiers, spatiotemporal architectures, pose-based models, and transformer networks, and situates the proposed work within the landscape of current limitations: data scarcity, absence of severity annotation, poor cross-domain generalisation, and the lack of clinical validation frameworks.

Chapter 3 — Implementation and Methodology constitutes the core contribution of this thesis. It documents the full development pipeline: dataset construction and re-annotation, solution 1 prototype and failure analysis, the biomechanical feature engineering framework, the dual-stream architectural designs, the experimental protocol including cross-validation and seed sensitivity analysis, the quantitative results on both internal and external test sets, and an honest diagnosis of the underfitting behaviour observed throughout training.

CHAPTER 1

THEORETICAL FOUNDATIONS AND KEY CONCEPTS

1.1 Introduction

The diagnosis of Autism Spectrum Disorder has historically relied on qualitative clinical observations—a process inherently bounded by subjectivity and resource constraints. The advent of deep learning offers a paradigm shift, enabling the translation of complex behavioral phenotypes into objective, quantifiable computer vision biomarkers. Bridging clinical reality with computational intelligence is not just a technological challenge, but a critical imperative for early intervention.

1.2 Autism Spectrum Disorder (ASD)

1.2.1 Definition and Core Characteristics

ASD is clinically-diagnostically characterized as a neurodevelopmental disorder that entails persistent deficits in social communication and social interaction in multiple contexts, along with restricted/repetitive behaviors (RRBs). Following the guidelines stipulated by the ICD-11, such deficits emerge during early development and lead to clinically significant impairment in several realms, including personal, family, social, educational, or occupational functioning. The spectrum approach reflects considerable variation in both the degree of impairments and ASD symptoms, which range from people requiring substantial support due to their developmental problems to highly intelligent individuals with prominent social/pragmatic challenges.

The clinical heterogeneity that defines ASD poses both a challenge and an opportunity for computer vision systems. Models must learn to distinguish between neurotypical variability and truly diagnostic motor/social atypicalities without overfitting to spurious correlations (e.g., clothing or lighting). The core characteristics directly translate to measurable computational targets: **(i)** Reduced social gaze → eye-tracking heatmaps and fixation duration histograms; **(ii)** Atypical prosody → spectral entropy and pitch variability over time; **(iii)** Repetitive motor behaviors → periodicity detection in skeleton joint trajectories.where harmonic peaks in this frequency band are strongly associated with stereotypic movements [4].
Thus, faithful technical operationalization of the clinical definition is the first necessary step toward any automated screening system.

Global Diagnostic Criteria (DSM-5 and ICD-11):

The clinical definition and diagnosis of Autism Spectrum Disorder are globally standardized by two primary systems:

- **DSM-5-TR:** the American Psychiatric Association's Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition [1].
- **ICD-11:** the World Health Organization's International Classification of Diseases, Eleventh Revision [5].

Both frameworks consolidate historical subcategories (such as Asperger's syndrome) into a single, unified spectrum. They share a core diagnostic dyad: persistent deficits in social communication and reciprocal social interaction, paired with restricted, repetitive, and inflexible patterns of behavior or interests.

1.2.2 The Critical Importance of Early Intervention

Early intervention refers to the systematic delivery of evidence-based behavioral, developmental, or educational therapies beginning at or before the preschool period (typically ages 12–36 months), when neural plasticity is maximal. The clinical consensus, derived from longitudinal cohort studies and randomized controlled trials (e.g., the Early Start Denver Model), is that early intervention produces dose-dependent improvements in IQ, language, and adaptive behavior, with meta-analytic effect sizes of $d = 0.74$ (95% CI: 0.58–0.90) for cognitive outcomes compared to late or no intervention [2]. The critical window hypothesis posits that interventions initiated before age 3 capitalize on experience-expectant learning mechanisms, whereas delays beyond age 5 require substantially greater therapeutic intensity to achieve comparable gains.

The imperative for early detection places stringent requirements on system design. Unlike gold-standard ADOS-2 administration (which takes 45–60 minutes), video-based AI must extract diagnostic features from much briefer, often unstructured home recordings. This necessitates computationally efficient architectures (e.g., MobileNetV3 or X3D for real-time processing) and robust domain adaptation to handle variable capture conditions. Moreover, the intervention time window informs model evaluation: ROC-AUC must be supplemented with time-dependent metrics such as the early detection rate (EDR), defined as proportion of true positives identified before a clinical cutoff age $T_{\text{cutoff}} = 30$ months. Recent benchmarks report that hybrid AI–clinician workflows reduce T_{diag} from 40.2 ± 12.8 months (standard care) to 18.5 ± 5.5 months ($p < 0.001$) [3].

1.2.3 Current Diagnostic Methods

Clinical Observations VS Instrumental Methods

Current diagnostic methods for ASD are methodologically divided. *Clinical observation* (unstructured) relies on a trained clinician's narrative judgment during free play or parent–child interaction, guided by DSM-5-TR criteria but not constrained by a manualized script. Conversely, *instrumental methods* employ standardized, validated instruments such as **the Autism Diagnostic Observation Schedule, Second Edition (ADOS-2)** or **the Autism Diagnostic Interview—Revised (ADI-R)**. These instruments impose semi-structured situations (e.g., "ball play," "bubble pop") designed to elicit specific target behaviors, offering inter-rater reliability $\kappa > 0.80$ compared to $\kappa \approx 0.55{-}0.65$ for unstructured observation [6].

Indeed, from the point of view of computer processing, the clinical observation approach is a prototypical example of a semi-supervised learning task: the inputs are highly dimensional and

unstructured (such as lighting differences, different camera angles, occlusions); while labeling (based on the clinician's assessment) is inherently low-dimensional and corrupted with noise. Instrumental approaches, however, fall under the umbrella of structured prediction: here, the protocol of recording establishes M fixed tasks ("Call Name", "Joint Attention", "Imitation"), which allows treating the problem as M binary/ordinal classification problems of time-aligned video clips [3].

1.2.4 From Clinical Criteria to Computer Vision

The Motor Signature of ASD

The motor signature of ASD refers to a constellation of atypical movement patterns that, while not part of the core DSM-5-TR diagnostic criteria, are empirically robust and often precede social–communication symptoms. These include: **(i)**hypotonia (low muscle tone) observable in infancy, **(ii)**delayed gross motor milestones (sitting, crawling, walking), **(iii)**atypical gait (reduced arm swing, increased stride variability), **(iv)**poor postural control, **(v)**stereotyped motor mannerisms (hand-flapping, toe-walking, body rocking). Meta-analyses indicate that motor impairments are present in $\approx 85\%$ of ASD children and have moderate-to-large discriminative ability (Cohen's $d = 0.7-1.2$) between ASD and typically developing peers [7].
In computer vision and movement science, the motor signature is operationalized as a vector of kinematic and dynamic features extracted from pose estimation outputs (e.g., MediaPipe, OpenPose, ViTPose).

Behavioral Biomarkers for ASD

Social-Visual Signature (Gaze and Joint Attention Biomarkers): deficits in social–visual engagement are among the earliest observable signs of ASD, typically emerging between 6–12 months of age. These include: **(i)**reduced preferential attention to biological motion and faces, **(ii)**diminished eye contact (hypogaze), **(iii)**atypical scanning patterns (increased mouth-directed gaze, decreased eye-directed gaze), **(iv)**impairments in joint attention—the triadic coordination of gaze between self, other, and object [8].
In the ADOS-2, joint attention is explicitly elicited through tasks such as "bubble pop" and "mechanical toy."

Facial Signature (Expression Dynamics and Affect Atypia): Clinically, individuals with ASD exhibit reduced spontaneous facial expressivity, fewer social smiles, and incongruent facial affect relative to context. Unlike neurotypical individuals who show rapid, coordinated activation of facial Action Units (AUs), ASD is associated with: **(i)**lower AU6 (cheek raiser/blink) and AU12 (lip corner puller) intensity during positive stimuli, **(ii)**increased asymmetry in AU4 (brow lowerer) during frustration, **(iii)**delayed onset latency of AU1+2 (inner/outer brow raiser) during surprise [9].
these signatures are extracted using AU detection models (OpenFace, MediaPipe Face Landmarks, AUs regression networks).

1.2.5 Significance for Automated Video-Based Detection

The emergence of automated AI systems that use video recordings to detect Autism Spectrum Disorder (ASD) has resulted in a huge paradigm change in the field of neurodevelopmental medicine. Diagnosis of autism involves long and detailed clinical tests. Such processes are normally associated with bottleneck issues leading to delayed diagnosis. Video recording provides solutions to such problems through machine learning algorithms, which help assess facial expressions, body movements, eye movements, and social interaction from videos.

Facilitating Early Detection and Crucial Intervention

A child's brain possesses its highest level of neuroplasticity during the first few years of life. While traditional clinical diagnoses are often not finalized until a child is 3 to 5 years old, video-based AI can identify subtle behavioral markers—such as atypical motor postures, reduced social smiling, or fleeting eye contact—in infants between 6 and 36 months. Catching these signs early allows families to start targeted behavioral therapies during critical developmental windows, fundamentally improving long-term communication and social outcomes.

Democratizing Access and Reducing Healthcare Disparities

Traditional diagnostic pipelines frequently involve long waiting lists (sometimes exceeding a year) and expensive clinical visits. Video-based detection models can run on standard smartphones or consumer-grade cameras, requiring only basic computing power or cloud processing. This effectively: **(i)** Reaches underserved communities in rural or low-income areas that lack specialized autism clinics, **(ii)** Lowers financial barriers, offering a low-cost screening tool accessible to anyone with a phone, saving expensive resources like MRIs for deep clinical follow-ups.

Eliminating Subjectivity with Objective Metrics

Standard clinical tools rely heavily on the qualitative interpretation of an observer. In contrast, computer vision frameworks use technologies like 2D and 3D pose estimation to extract "skeletons" of moving individuals. **(i)** They measure highly precise spatio-temporal dynamics of gestures, gait variability, and midline postural control that are virtually invisible to the naked human eye, **(ii)** They provide granular, continuous risk scores rather than a rigid binary "yes/no" classification, giving clinicians a more nuanced view of the child's behavioral phenotype.

Non-Invasive and Naturalistic Assessments

Children, particularly those on the autism spectrum, can become highly anxious, uncooperative, or overstimulated in unfamiliar clinical environments—or when fitted with physical motion sensors. Video analysis is completely non-invasive. It allows researchers and clinicians to analyze a child's behavior while they play freely at home or in a standard playroom, capturing authentic behavior rather than a stress response.

Preserving Patient Privacy and Enhancing Data Sharing

Using advanced pose-estimation software, AI can strip away identifying visual information—such as a child's face, skin tone, or home environment—leaving behind only a mathematical skeleton of their movements. This anonymity makes it significantly safer to share behavioral data between global specialists for secondary opinions, train medical students, and build larger, less biased research datasets.

The reliable detection of such complicated motor, facial, and eye movement biomarkers within a video stream is a difficult representational task. Typically, hand-engineered approaches often fall short in generalizing across the huge variability witnessed in reality with respect to natural videos captured in home settings, involving variability in illumination, occlusions, and toddler shape. It is only due to hierarchical feature learning offered by Artificial Neural Networks that modern screening tools manage to automatically distinguish these subtle patterns in the visual data. The next section describes fundamental aspects of Deep Learning.

1.3 Fundamentals of Deep Learning

The automated detection of neurodevelopmental conditions such as Autism Spectrum Disorder (ASD) from raw video footage presents a challenge that is fundamentally different from problems addressable by classical statistical methods. The data is inherently *unstructured*: individual pixel values, skeletal joint trajectories, and temporal motion patterns carry no meaning in isolation; their significance emerges only from complex, high-dimensional interactions distributed across space and time. This section introduces the deep learning concepts that serve as the theoretical backbone of the detection system proposed in this thesis—from the basic computational unit of a neural network to the optimization algorithms that enable automatic feature discovery from data.

1.3.1 Artificial Neural Networks(ANNs)

From the Perceptron to Multi-Layer Networks:

Deep Learning is a special kind of Machine Learning technology that employs multiple layers of artificial neural networks in order to emulate the brain's capability to learn from data. As opposed to conventional systems, which need explicit instructions from the programmers (such as "look for hands"), deep learning algorithms automatically identify the features needed to perform a task based on large quantities of raw data [11, 10].

Key Features:

- **Hierarchical Learning:** Information is processed through layers in which higher levels learn more complex ideas (i.e., edges > shapes > faces).
- **End-to-End Process:** Raw data (such as a video recording) can be converted into meaningful results (such as the probability of autism) without human intervention

in between.

- **Performance Scaling:** Performance generally increases as the amount of data grows, in contrast to conventional methods, which reach a ceiling level of accuracy.

Neuron:

A neuron is a cell in the brain that enables the body to think and react to stimuli. In deep learning models, it provides a biological template for designing the artificial neuron that constitutes the core of a neural network [11].

Neuron Function: Neurons act as information collectors by accepting input signals from sensory cells or other neurons and firing electrical pulses if specific levels of stimuli are met [11].

Application of Artificial Neural Networks: Artificial neurons work in a similar manner to biological neurons by receiving numeric inputs, assigning significance to each input, and transmitting output signals to enable the machine to learn data patterns (e.g., autism detection from video frame images) [11].

The Perceptron:

The perceptron, introduced by Frank Rosenblatt in 1958, represents the simplest form of a neural network. It takes multiple binary inputs, applies weights to them, sums the weighted inputs, and passes the result through an activation function to produce an output [12].

Mathematically, a perceptron computes:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1.1}$$

where x_i are inputs, w_i are weights, b is a bias term, and f is an activation function (typically a step function for binary classification).

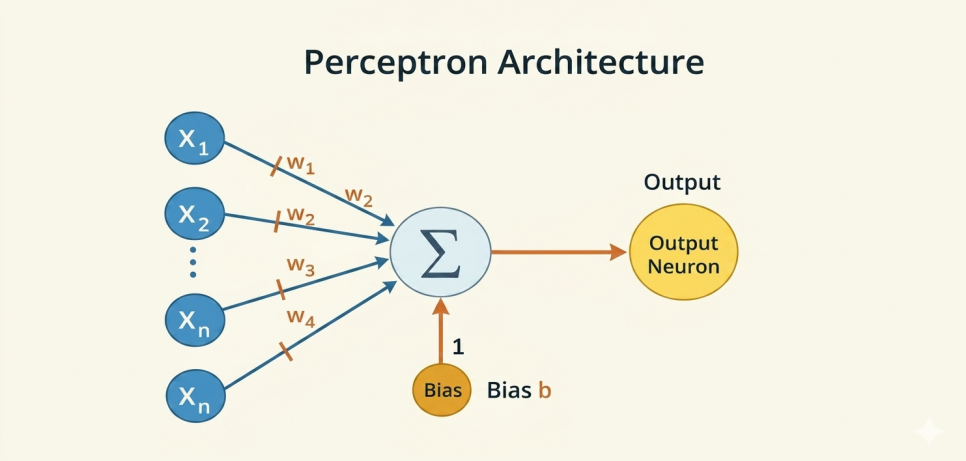

Figure 1.1: Perceptron architecture showing inputs, weights, bias, and output neuron.

Although revolutionary, single layer perceptrons are still limited in that they can only learn linearly separable problems [12].

The limitations of single layer perceptrons were famously shown by Minsky and Papert's analysis of the XOR problem, marking the first "AI winter" of the late 1960s [12].

Multi-Layer Perceptrons (MLPs):

The answer to the limitations of the perceptron lies in the depth of the network. A multi-layer perceptron (MLP) is composed of multiple layers of interconnected perceptrons: the input layer, the hidden layer, and the output layer [12].

The multiple layers of the hidden layer apply a weighted sum function followed by a non-linear activation function. This complex composition of MLPs, with multiple layers of perceptrons, enables the strategic application of non-linear functions to learn non-linear patterns [12].

This architectural advancement of the perceptron, along with the application of the back-propagation algorithm for training the perceptron, enables the MLP to learn non-linear separability.

Key Components of MLPs:

The main components of multi-layer perceptrons include [12] :

- **Input layer:** Receives the raw data (e.g., pixel values, feature vectors)
- **Hidden layers:** Perform intermediate computations and feature transformations
- **Output layer:** Produces the final prediction (classification, regression, etc.)
- **Activation functions:** Introduce nonlinearity (sigmoid, tanh, ReLU)
- **Weights and biases:** Learnable parameters adjusted during training

The combination of learning rules (such as backpropagation and gradient descent) and perceptron architectures forms the foundation of ANNs' expertise in pattern recognition, prediction, and decision-making tasks [12].

1.3.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Network (CNN or ConvNet) is an advanced type of feed-forward neural networks that have been specifically developed for processing information whose topology is defined on a grid structure, examples include time series data and images which can be considered two-dimensional grids of pixel values. While in fully connected networks, each neuron is directly connected with every other neuron in its adjacent layer, CNNs take advantage of three main concepts that enable significant reduction in learnable parameters [13, 14].

Why CNNs Are Suitable for Images:

Traditional MLPs struggle with image data because of:

1. **Parameter explosion:** A fully connected network processing a modest 224×224 RGB image would require over 150 million parameters just in the first layer
2. **Lack of spatial invariance:** MLPs treat each pixel independently, failing to leverage the spatial structure of images
3. **No translation equivariance:** The same feature appearing in different locations requires separate learned detectors

CNNs address these limitations through three key ideas: **local connectivity**, **weight sharing**, and **pooling**.

Core Concepts of CNNs:

1. **Convolutional Layers:**

   Shift-invariance is achieved in Convolutional Neural Networks through the concept of weight sharing between neurons in the convolutional layer. Each neuron in the CNN model is designed to analyze information from unique local receptive fields. This allows the model to detect spatial structures such as edges, corners, and line end points that are formed by neighboring pixels [16, 15].

   The Function of Convolutional Layers Convolutional layers play an essential role in the Convolutional Neural Network as the basic components. They conduct spatial convolution of input data with a set of learnable filters referred to as kernels. The operation of these layers depends on several key parameters that include:

   - **Convolution kernels (filters):** Small matrices (typically 3×3 or 5×5) that detect specific features like edges, textures, or patterns. Each kernel learns to recognize a particular visual feature.
   - **Stride:** Determines how many pixels the kernel moves at each step. Larger strides reduce computational load but may miss fine details.
   - **Padding:** Adding zeros around the input border to control output spatial dimensions and preserve information at edges.

   In convolution, each kernel scans the image, calculating the dot product of each kernel's weights with the image's region of interest, resulting in a feature map that shows the presence of specific features within the image .

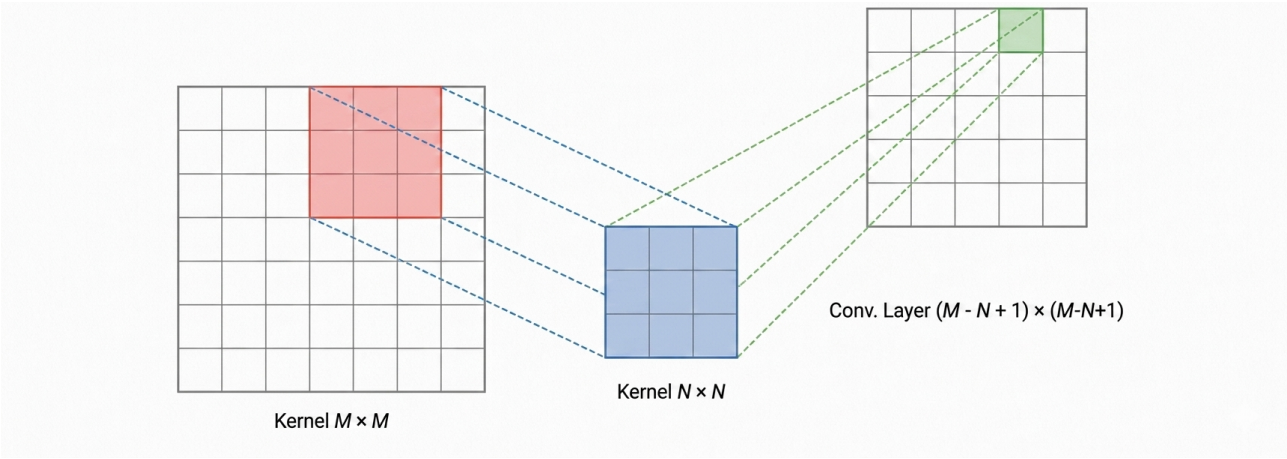

Figure 1.2: Mapping of the convolutional kernel filter to the input image[17].

2. **Activation Functions:**

Because of their fundamental impact on the performance of deep neural networks, activation functions have remained a highly active and critical area of research. The development of these functions generally follows a distinct historical progression:

Early Saturating Functions: In early neural network architectures, functions like Sigmoid and Hyperbolic Tangent (Tanh) were the standard choices. Between the two, Tanh was typically preferred because its output.

- **Sigmoid:**

$$\sigma(x) = \frac{1}{1+e^{-x}} \text{ for binary classification outputs} \tag{1.2}$$

- **Tanh:**

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \text{ for outputs ranging between } -1 \text{ and } 1 \tag{1.3}$$

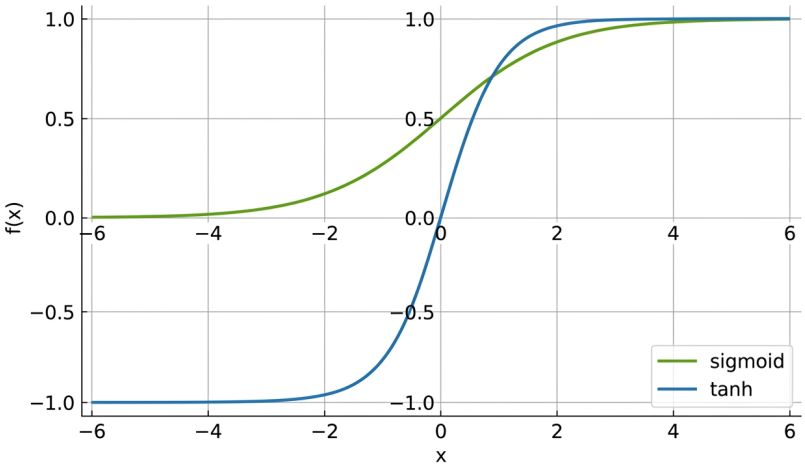

Figure 1.3: Curve of activation functions

The Vanishing Gradient Problem: As networks grew deeper, researchers found that these "saturating" functions severely restricted learning. Because their gradients approach zero at the extremes, they caused the vanishing gradient problem, preventing deeper layers from updating effectively.

The ReLU Solution: To overcome this bottleneck, Nair and Hinton (2010) introduced the Rectified Linear Unit (ReLU). By acting as a non-saturating function, ReLU successfully bypassed the vanishing gradient issue. Due to this breakthrough, it has been widely adopted and remains a cornerstone in modern.

- **ReLU:**

$$f(z) = \max(0, z) \tag{1.4}$$

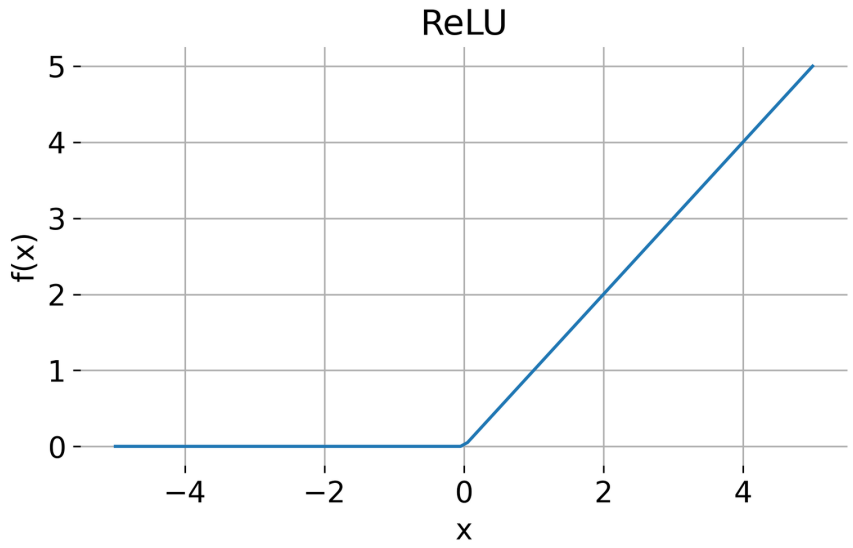

Figure 1.4: Curve of activation functions (ReLU)

The rectified linear activation function. This activation function is the default activation function recommended for use with most feedforward neural networks. Applying this function to the output of a linear transformation yields a nonlinear transformation.

3. **Pooling Layers:**

   In earlier designs such as LeNet-5, spatial downsampling was first employed by applying average pooling, which is currently referred to as "sub-sampling." This helped simplify computation and slightly increase translation invariance[18].

   In subsequent models, various types of pooling mechanisms were exhaustively compared. Studies revealed that Max Pooling is generally more effective than average pooling in detecting objects since it retains only the most prominent features and is faster to converge during training[19].

   Max Pooling became widely accepted in revolutionary models like AlexNet, where the idea of "overlapping" max pooling was also proposed to avoid overfitting[20].

Finally, the idea of Global Average Pooling was conceived as an architectural replacement for dense layers and significantly reduced the number of parameters involved in training, avoiding overfitting during classification.

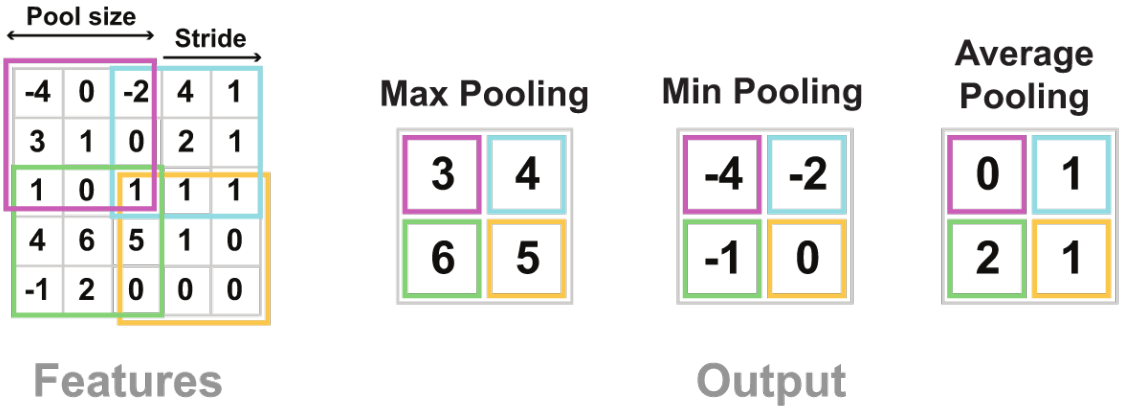

Figure 1.5: pooling

Pooling progressively reduces the spatial size of representations, allowing deeper layers to have larger receptive fields while controlling overfitting.

4. **Fully Connected Layers:**

   FC layers are considered to be the ideal architecture of deep feedforward neural networks (Sarker, 2021). FC layers can be regarded as continuous linear operators in finite-dimensional vector spaces that carry out global feature mapping to transform the input data into an abstract space of higher level.

5. **2D Convolution with Activation Function:**

   The complete 2D convolutional layer operation, including the bias term and non-linear activation, is mathematically expressed as:

$$\mathbf{F}_k[i,j] = \sigma\left(\sum_c \sum_m \sum_n \mathbf{w}_k[c,m,n]\,\mathbf{X}[c,\ i{+}m,\ j{+}n] + b_k\right) \tag{1.5}$$

   where:

   - $F_k[i,j]$ — output feature map value at position (i,j) for kernel k
   - $\sigma(\cdot)$ — non-linear activation function (e.g., ReLU, sigmoid, tanh)
   - c — input channel index (e.g., RGB channels or heatmap joint channels)
   - m,n — spatial indices within the kernel window (typically $m,n \in [0,k_h-1]\times[0,k_w-1]$)
   - $w_k[c,m,n]$ — learnable weight of kernel k at channel c, position (m,n)
   - $X[c,i+m,j+n]$ — input value at channel c, spatial position $(i+m,j+n)$
   - b_k — learnable bias term for kernel k

1.3.3 Introduction to Sequence Models

The problem of automatic video data analysis for diagnosing autism spectrum disorder (ASD) is fundamentally difficult due to the property of the behavior characteristics, especially motor stereotypes, which do not reside in single images, but in how these evolve over time. While it is true that CNNs are highly effective at feature extraction from individual images, they do not have the ability to track motion, and thus cannot separate transient motion from stereotyped movements. This problem can be solved with sequence models, which explicitly model temporal dependencies, and thus include temporal context in the process of learning the features. Temporal models are particularly useful when order matters in the input data, e.g., in time series analysis, text analysis, audio signal processing, and video frames analysis. In the case of video data analysis for ASD detection, both temporal and spatial patterns are equally important.

The Concept of Temporal Dependency

Temporal dependency refers to the relationship between an event at time t and events that occurred at previous times $\{t-1, t-2, \ldots, t-k\}$. In many real-world sequences, the immediate past is not sufficient; long-range dependencies (e.g., the subject at the beginning of a paragraph affecting a pronoun at the end) are crucial.

Mathematically, for a sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, the goal is often to model:

$$P(\mathbf{y}_t | \mathbf{x}_1, \ldots, \mathbf{x}_t) \tag{1.6}$$

This requires the model to maintain a *hidden state* that acts as a "memory" of the past [22].

Formally , a sequence model maps an ordered input sequence $X = (x_1, x_2, \ldots, x_T)$, where each $x_t \in \mathbb{R}^d$ denotes the feature vector extracted from frame t, to a sequence of latent representations $H = (h_1, h_2, \ldots, h_T)$ or to a global decision y. The defining property of such models is the recurrence relation

$$h_t = f_\theta(h_{t-1}, x_t), \tag{1.7}$$

where h_t denotes the hidden state at time t, h_{t-1} the previous state, and f_θ a parameterized non-linear function. Equation (1.7) embodies the principle of *temporal context*: the interpretation of the current observation x_t is conditioned on the accumulated history h_{t-1}. For video-based ASD analysis, this property is indispensable, since a hand-flapping or body-rocking event is only meaningful when considered as a repetition over consecutive frames rather than as an instantaneous posture [21].

Recurrent Neural Networks (RNNs)

RNNs are the foundational architecture for sequence modeling. They process sequences one element at a time while maintaining a hidden state $\mathbf{h}_t$ that is updated recurrently:

$$\mathbf{h}_t = \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{1.8}$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y \tag{1.9}$$

However, vanilla RNNs suffer from the **vanishing/exploding gradient problem** during backpropagation through time (BPTT), making it nearly impossible to learn long-range dependencies (typically beyond 10 steps) [23].

Long Short-Term Memory (LSTM) Networks

LSTMs, introduced by Hochreiter and Schmidhuber [24], are designed to overcome the vanishing gradient problem. They introduce a **cell state** $\mathbf{c}_t$ that acts as a conveyor belt, and three gates to control the flow of information:

- **Forget Gate** ($\mathbf{f}_t$): Decides what to discard from the previous cell state.
- **Input Gate** ($\mathbf{i}_t$): Decides what new information to store.
- **Output Gate** ($\mathbf{o}_t$): Decides what to output based on the cell state.

The update equations are:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{1.10}$$

where $\odot$ is element-wise multiplication and σ is the sigmoid function.

Gated Recurrent Unit (GRU)

To simplify the LSTM architecture while retaining its ability to solve the vanishing gradient problem [25]. introduced the Gated Recurrent Unit (GRU). Unlike LSTMs, GRUs do not have a separate cell state; they instead use a combined hidden state and only two gates:

- **Update Gate** ($\mathbf{z}_t$): Decides how much of the past knowledge needs to be passed along to the future.
- **Reset Gate** ($\mathbf{r}_t$): Determines how much of the previous state to forget.

The update equations for a GRU are:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z) \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_h) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned} \tag{1.11}$$

Because of its simpler structure, a GRU is computationally more efficient than an LSTM and often yields comparable performance on many sequential modeling tasks.

From RNNs to Gated Architectures

The evolution from RNNs to gated architectures (LSTM and GRU) represents a paradigm shift: from naive sequential propagation to **controlled information flow**. The key insight is that the model should learn *what to remember, what to forget, and when to output*. This gating mechanism allows gradients to flow unchanged through many time steps, effectively learning dependencies spanning hundreds of steps.

Bidirectional LSTM Network (Biomechanical Stream)

In many applications, especially biomechanical analysis (e.g., gait recognition, motion capture), past *and* future context are both informative. A Bidirectional LSTM (BiLSTM) [26] processes the sequence in two directions:

- A forward LSTM reads the sequence from $\mathbf{x}_1$ to $\mathbf{x}_T$ producing hidden states $\overrightarrow{\mathbf{h}}_t$.
- A backward LSTM reads from $\mathbf{x}_T$ to $\mathbf{x}_1$ producing $\overleftarrow{\mathbf{h}}_t$.

The final hidden state at time t is the concatenation: $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$. This provides each time step with full context of the entire sequence, which is crucial for tasks like joint angle prediction or activity recognition.

3D Convolutional Residual Network (Heatmap Stream)

For spatiotemporal data such as video or heatmap sequences (e.g., from thermal cameras or pose estimation heatmaps), 3D Convolutional Neural Networks (3D CNNs) are effective. A 3D CNN performs convolution across both spatial dimensions (height, width) and the temporal dimension.

A **Residual Network (ResNet)** is extended to 3D by using 3D kernels. The residual block learns a mapping:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x} \tag{1.12}$$

where $\mathcal{F}$ represents the 3D convolutional layers. This "skip connection" alleviates the vanishing gradient problem in deep networks. In the *Heatmap Stream*, this 3D ResNet extracts hierarchical spatiotemporal features from a sequence of heatmaps (e.g., probability maps of joints over time) [27].

Dual-Stream Fusion

Dual-stream fusion combines complementary information from two distinct input streams before making a final prediction. A classic example is two-stream networks for action recognition [28]:

1. **Spatial Stream**: Processes raw RGB frames (appearance).
2. **Temporal Stream**: Processes optical flow (motion).

In our context, the **Heatmap Stream** (3D ResNet) processes location/confidence maps, while the **Biomechanical Stream** (BiLSTM) processes sequential physical parameters (e.g., joint angles, forces). Fusion can occur at:

- **Early fusion**: Concatenate inputs before processing.
- **Late fusion**: Combine the final feature vectors from both streams (e.g., concatenation, summation, or weighted average).
- **Intermediate fusion**: Merge features at multiple layers.

The fused representation is then passed to a classifier or regressor.

1.4 Human Pose Estimation and Skeleton Extraction

1.4.1 The Role of Skeletal Data in Behavioral Analysis

In the context of machine screening for ASD, identifying the "motor signature" of a given individual demands separating out the pure kinematics from the noise present in raw video pixels. It is common knowledge that direct application of deep neural networks to RGB pixels often leads to such models learning spurious correlations between unimportant environmental details and achieving high levels of accuracy through sheer memorization[29].

Skeletal data helps overcome this weakness since it acts as a tight bottleneck structure. By projecting the patient's body movements on the low-dimensional graph with interconnected joints, all the confounding information becomes deliberately ignored. Thus, the generated skeleton remains invariant to background clutter, different lighting setups, as well as variations in the style and color of patients' clothes. This forces the model to identify exact dynamic patterns in the form of hand-flapping or the characteristic walking pattern of atypical gait.

An important aspect of using skeletal data relates to privacy protection and ethics in clinical practice, namely anonymization by design. Processing only the pose graphs automatically removes any face features or biometric markers from the analysis while remaining true to the requirements of motor biomarkers necessary for proper diagnosis. This allows full adherence to various regulatory frameworks, e.g., HIPAA or GDPR [30].

1.4.2 Evolution of YOLO-based Pose Estimation

The field of Human Pose Estimation (HPE) has traditionally been plagued by complex, multi-step approaches that require heavy computational resources. The integration of YOLO into the world of HPE stands out as an important moment in this area because it focuses more on consistent, real-time operation while not sacrificing structural accuracy [31].

First developed by Redmon [32], as a pure bounding box detector, the YOLO framework challenged the traditional world of two-step detectors (Faster R-CNN, among others), as it redefined object detection as one problem of regression. As it progressed through various iterations, YOLO frameworks integrated anchor boxes, Feature Pyramid Networks, and PANets to refine multi-scale semantic integration.

Adapting this technique to keypoints required a fundamental change in thinking, as it went from using a spatial bounding box to fine-grained keypoint regression. As opposed to other methods that use computationally costly heatmap generation (such as SimpleBaseline or HRNet), current YOLO-based HPE solutions perform direct vector regression for keypoints through their detector's head.

1.4.3 Architecture and Mechanics of YOLOv8x-pose

In the current paper concerning ASD motor signature extraction, we use the YOLOv8x-pose, proposed by Ultralytics [33]. YOLOv8 makes use of the modified CSPDarknet53 backbone architecture paired with a novel anchor-free decoupled head for objectness, classification, and keypoint regression tasks.

From the perspective of a macro-level design philosophy, YOLOv8-pose relies on a top-down framework for human pose estimation. It starts with regressing a highly accurate bounding box encompassing the human figure, followed by the regression of keypoints belonging to the aforementioned object's structure. It should be noted that, compared to bottom-up approaches, top-down methods tend to provide higher quality results when there is just a single subject in an image and no need for grouping multiple detected joints into people.

As regards model architecture selection, YOLOv8x-pose seems like the best possible fit due to its size parameters. Namely, while edge devices may benefit from smaller models that would enable faster processing speeds, clinical computer vision applications prefer precise detection of subtle phenotypic traits (micro-tremors or wrist angles of particular ASD stereotypical behaviors). That is why YOLOv8x-pose was chosen since, in addition to a high number of parameters (i.e., complexity), it has a large receptive field enabling highly accurate estimation of keypoints in cases of motion blur or occlusions.

As for the format used in this paper for representing the output data, YOLOv8x-pose uses the MS COCO pose format [34]. For each detected person, a tensor consisting of N = 17 keypoints is provided. Specifically, each point is represented as follows:

$$K_i = (x_i, y_i, v_i) \quad \text{for} \quad i \in \{1, 2, \ldots, 17\} \tag{1.13}$$

where the coordinates x_i and y_i describe normalized 2D coordinates of that joint in the image, and v_i, ranging from 0 to 1, is the confidence (pseudo-visibility) score calculated by the model. As such, the obtained continuous stream of coordinates (X_1, ..., X_T) will form the basis of further analysis by diagnostic classifiers.

1.5 Conclusion

This chapter has established the foundational concepts necessary for understanding video-based ASD detection using deep learning. We have seen that:

ASD is defined by its distinctive and identifiable patterns of behavior in social communication and repetitive behaviors according to DSM-5 and ICD-11 criteria [1, 5]. These patterns in behavior, which include abnormalities in eye contact, gestures, facial expressions, and motor mannerisms, are precisely those which computer vision can potentially identify and measure.

The existing methods for diagnosing ASD, though validated in clinical settings, are not without certain shortcomings. While clinical methods are based on observations by qualified professionals in clinical settings, instrumental methods like fMRI and EEG are objective but costly and not always accessible to everyone. This is why researchers are actively seeking alternative methods.

Deep learning technology offers several potential tools in automatically identifying features in visual data. CNNs are particularly effective in automatically identifying spatial features in individual frames, while sequence models like LSTMs and Transformers can automatically identify temporal patterns in sequences of frames. These technologies put together can potentially model spatio-temporal patterns in behavior.

Video analysis architectures (CNN+LSTM, 3D CNNs, Two-Stream, Video Transformers) provide different ways of tackling the spatio-temporal problem of feature extraction, which come with their own trade-offs in terms of computational efficiency, interpretability, and ability to learn different behavioral patterns.

Human Pose Estimation acts as a vital structural bottleneck for behavioral analysis. By converting raw pixels into skeletal graphs via top-down frameworks like YOLOv8-pose, we deliberately ignore confounding environmental noise and facial biometrics. This approach preserves patient privacy while forcing models to strictly learn the dynamic kinematic markers of ASD.

CHAPTER 2

STATE-OF-THE-ART

2.1 Introduction

Before the rise of deep learning, researchers relied on a variety of traditional approaches to identify early signs of autism. These methods typically focused on handcrafted features, clinical observations, or classical machine-learning techniques designed to capture behavioural cues from videos or sensor data. Although often limited in accuracy and scalability, they laid the foundation for today's modern, data-driven approaches and remain an important part of the evolution of automated ASD detection.

This chapter provides an overall review of current literature regarding automated Autism Spectrum Disorder (ASD) detection using video-based analysis of human behaviour. The review is organised into three parts: an overview of pre-deep-learning (traditional) methods, a classification of the different deep learning approaches, and finally a summary of how these methods have been applied in practical settings.

2.2 Classical Methods

In the pre-deep learning era, the methods used to identify behaviors in autism spectrum disorder cases involved classical computer vision methodologies and human-designed signal processing pipelines. These traditional methods formed the basis of the concepts within the field and defined the key behaviors that needed to be identified through automation, namely repetitive motor behaviors, stereotypical gestures, and irregular movement patterns. While these methods have largely been superseded by deep learning, their workings offer insight into the need to shift paradigms.

2.2.1 Methods based on hand-crafted features

Some of the earliest computational methods for behavioral analysis related to ASD made use of descriptors such as optical flow features and statistics about the gradients in raw video data. Among the most commonly used descriptors were HOG (Histogram of Oriented Gradients) and HOF (Histogram of Optical Flow), the former providing information about the orientation of edges, while the latter described the magnitude and direction of pixel movements. One variation of these was dense trajectories, which described the HOG, HOF, and MBH (Motion Boundary Histogram) descriptors for densely tracked trajectories of pixels in time.

These techniques quantified collective statistics of movement, which included the frequency, magnitude, and directionality of stereotypical motions such as hand flapping, head banging, and body rotation. The Self-Stimulatory Behavior Dataset (SSBD), created by Rajagopalan et al. in 2013, has been the benchmark data set for testing handcrafted motion features in this setting, where there are 75 naturally occurring videos recorded from YouTube showing children with autism spectrum disorder displaying stereotypic motor actions [21]. Initial histogram-based motion analysis on the SSBD yielded recognition rates between 75% and 87%.

Although easy to interpret and computationally efficient, handcrafted feature techniques

were inherently constrained in several ways. Their vulnerability to camera motion, background interference, illumination changes, and individual occlusions made them unreliable when implemented in realistic settings. Since handcrafted feature techniques used fixed representations, they could not account for the extensive variation in autism behavioral expression among different people, ages, and recording settings. Furthermore, since handcrafted feature techniques did not model high-order temporal dependencies, they were unable to represent the periodicity and self-similarity inherent in stereotypic motion sequences, like pathological stimming compared to regular childhood motions.

2.2.2 Pose-Based Traditional Techniques

In parallel with optical flow techniques, another stream of research focused on pose estimation, where the extraction of coordinates of joints from video footage represented an indirect measure of human movement dynamics based on a simplified 2D skeleton model. Even though initial 2D skeletons could be compromised by the performance of the then-available pose estimation techniques, their semantically meaningful and robust against the environment background representation was superior to that of the raw pixel-based representations.

Based on rule-based classifiers of skeletons, there were attempts to encode clinical markers, such as angular speed of the elbow joint during arm flapping or the tilt of the head over time, into threshold-based criteria. Clinical studies utilizing kinematic features for the autism detection task showcased the clinical relevance of the proposed measures, while also illustrating the fragility of rule-based formulation, wherein small variations in the position of the camera, proximity of the patient, or body proportions among age categories might invalidate the predefined rules.

A far more significant challenge involved the reliance on skeleton extraction accuracy. The traditional methods of pose detection prior to the advent of deep learning had considerable trouble with children because of their relatively small body size and rapid, jerky movements characteristic of young children who have ASD. The occlusion of certain joints due to stimming behavior — such as those involving hand-to-torso or hand-to-face contact — led to consistent detection failures, which then propagated downstream to produce classification errors.

2.2.3 Machine Learning with Features Engineering

The third approach in the pre-deep-learning era involved supervised machine learning classification algorithms, such as SVM, Random Forest, and k-NN, along with the use of engineered features. The main improvement in these approaches was that they learned discriminative decision boundaries from labeled training examples and thus generalized beyond programmed behavioral definitions.

Feature sets used for machine learning included the temporal frequency distribution using Fourier transformation of joint trajectories (which is related to the periodic nature of repetitive behaviors), optical flow intensity histograms that captured the activity associated with self-stimulation, and kinematic information about the motion of the upper limbs. One interesting

finding in this area was that specific frequency bands associated with the repetitive rate of behavior like arm flapping or body rocking had high diagnostic value.

These techniques, however, had one crucial structural limitation: the ability of a feature extraction pipeline to represent data is necessarily limited by its own design assumptions. Complex spatiotemporal relations, such as those involving the relationship between the rhythmicity of arm movement, torso motion, and overall motion context within the scene, cannot be properly represented through manually engineered features. Additionally, proper feature selection was necessary for these classifiers to work correctly, and their performance was significantly affected by variations in the distribution of input data.

2.3 Deep Learning Approaches

The incorporation of deep learning marked a watershed moment for the field of automated analysis of behaviors associated with autism spectrum disorders. Through the extraction of hierarchical feature representations directly from video inputs, rather than using pre-designed features, deep architectures enabled significant improvements in accuracy and robustness. The subsequent subsections analyze the most notable types of deep neural network architectures that have been utilized in addressing the problem at hand.

2.3.1 CNN-based Behavioral Classification for Videos

Deep architectures like Convolutional Neural Networks (CNN) have been employed in early ASD behavioral video analysis due to their ability to capture spatially rich feature hierarchies within video frames. The typical process involved modifying standard backbones used in image classification such as VGG-16, MobileNet, and ResNet to behavioral video recognition tasks by using the feature vector from each frame to generate a sequence of frame-level features via averaging or pooling.

As an example of how the CNN-based models have developed into hybrid models, we can mention the paper by Jabbar(2026) [35]. In this paper, based on the SSBD dataset but with the inclusion of a Normal Behavior category resulting in a classification problem of arm flapping, head banging, spinning, and normal behavior, the authors conducted an evaluation of five different architectures, namely, VGG-16, MobileNet, EfficientNet-B7, 3D CNN-LSTM, and the authors' own CNN-GRU architecture. In particular, each video was prepared by performing bounding box detection via OpenCV, rescaled to 224×224 pixels, and reduced to 30 equally spaced frames.

CNN-GRU model, which combines two convolutional layers for the extraction of spatial features and one gated recurrent unit (GRU) layer for temporal modeling, attained 96% validation accuracy and average k-fold accuracy of $92.94\% \pm 0.38\%$. This model outperformed all baselines in terms of accuracy with statistical significance (paired t-tests with Bonferroni corrections). The choice of using GRU over LSTM was because the former offered a relatively simple gating scheme and lesser computation cost but had similar temporal modeling capability

to the latter. Arm flapping and head banging performed best in per-class accuracy with F1 score of 0.98 each, whereas spinning proved to be the hardest because of its resemblance in motions to the other repeated classes.

Pure CNNs without the incorporation of temporal modeling are inherently inadequate in the understanding of the dynamics of behavioral processes. The hybrid CNN-GRU and CNN-LSTM models help achieve both strong spatial and sequential state modeling capabilities but are still dependent on the size of the temporal input sequence used. Furthermore, there is a problem with generalization since the study was carried out using just one set of data comprising 75 videos.

2.3.2 Spatiotemporal Deep Models: 3D CNNs and Two-Stream Networks

As a way to circumvent the fundamental limitation of frame-level CNN-based models, the concept of processing the video content as an integrated three-dimensional volume, which accounts for both space and time, gave rise to a group of spatiotemporal deep models. Two main approaches exist in this domain: 3D CNNs and two-stream networks.

3D CNNs implement convolution in three dimensions using spatiotemporal kernels and thus allow simultaneous representation of appearance and motion. Representative examples are C3D, I3D (Inflated 3D ConvNet), and SlowFast Networks. I3D networks were proposed by inflating the weights of an existing 2D ImageNet network to 3D space and demonstrated exceptional results on benchmarks of action recognition. This architecture was employed in several ASD-focused studies reviewed in Yang(2025) [36]. SlowFast networks utilize two distinct pathways in video processing, a slow one, operating at high spatial and low temporal resolutions, and a fast one, performing opposite.

The RGBPose-SlowFast Network presented in Prakash(2025) [37] and analyzed in Yang(2025) [36] used this strategy for ASD behavior recognition on the SSBD dataset, with 91% accuracy. In this model, skeletal joint heatmaps were used as an extra input modality for one of the network branches.

Two-stream networks, by contrast, use a more straightforward strategy. This is done by keeping the appearance (RGB frames) and motion (optical flow) streams completely separate and using a separate CNN branch for each, with feature fusion taking place late in the process. Two-stream I3D networks combined with SVM achieved 98.3% accuracy when working on the SSBD dataset, relying on temporal coherency between adjacent frames as a self-supervised factor [28]. Fusion of two-stream I3D networks outperformed single modality models on all ASD benchmarking tests.

The main drawbacks of spatiotemporally based deep networks are their computational complexity and the necessity of large-scale training datasets to avoid overfitting. These factors represent serious problems for their implementation in low-resource medical facilities, as well as being problematic due to the necessity of using richly labeled data sets when training on ASD behavioral data.

2.3.3 Pose-Based Deep Learning Models

Motivated by the success of deep learning in human pose estimation, a significant body of work has applied skeleton-based representations to ASD behavioral analysis, combining state-of-the-art pose extraction tools with temporal deep learning models. This approach offers several advantages over raw RGB-based methods: skeleton representations are invariant to irrelevant visual factors such as lighting, clothing, and background, provide a compact and privacy-preserving alternative to raw video, and encode the clinically meaningful spatial relationships between body joints explicitly.

Modern pose estimation backbones — including OpenPose (used in at least 10 of the 16 skeleton-based studies reviewed by Yang(2025) [36]), AlphaPose (which provides top-down 2D multi-person keypoint detection with 17-joint predictions), HRNet (which maintains full-resolution feature maps throughout processing for superior keypoint localization), and MediaPipe (Google's open-source framework) — now provide reliable 2D joint estimates even from monocular RGB video without depth sensors. The Yang(2025) [36] review identified HRNet as achieving particularly high accuracy in head-related ASD behavioral analysis, with one study reporting 98.2% accuracy using HRNet combined with OpenFace.

Approaches involving temporal modeling based on extracted skeletal sequences have used both LSTM and Transformer networks. CNN-LSTM models, which utilize CNNs to analyze spatial information regarding joints and LSTM networks to model the temporal information associated with skeletal sequences, were responsible for no less than eight papers mentioned in the review conducted by Yang et al., achieving F1-scores of up to 0.91 on SSBD for analyzing head poses. More advanced methods using multi-scale bidirectional LSTMs on frame sequences extracted from different CNN streams include the DASD model by Aldhyani and Al-Nefaie (2025) [38].

The DASD architecture deserves close consideration as a showcase for state-of-the-art approach to multi-stream pose-based networks. Designed specifically for classifying the act of hand flapping using SSBD binary videos, the network includes three streams based on pretrained CNN architectures such as EfficientNetV2B0, ResNet50V2, and DenseNet121, respectively, followed by a combination of channel attention and spatial attention along with a multi-scale bidirectional LSTM with three layers containing 256, 128, and 64 bidirectional units, additionally enhanced with temporal attention. Finally, the classification branch is composed of two dense layers with L2 regularizer, batch normalization, and residual connection. The network yielded 96.55% accuracy with 100% specificity (no false positives) on the test dataset, far outperforming any backbone in isolation like the EfficientNetV2B0 with 75.86%, and beating any prior attempt on the binary classification of SSBD dataset. Per-class F1 scores for hand flapping and normal class are estimated to be 0.97 and 0.96, respectively.

Nonetheless, the validation set consisted of merely 29 instances, and such a sample size would cause the effect of a single instance being incorrectly classified to influence sensitivity and specificity values by ±6 to ±8 percent. Cross-validation and cross-dataset testing were also not conducted, which reduces the validity of the results obtained. Moreover, the three CNN

backbone branches remained frozen, and no fine-tuning of the network with respect to autism spectrum disorder cases took place.

The issue common to many pose-based methods is their reliance on the accuracy of the skeleton extraction procedure. According to the Yang(2025) [36] review paper, the omission of joints in the skeleton owing to occlusions, poor lighting conditions, and unfavorable camera angles could have a detrimental impact on the results produced by the machine learning models. One study noted an increase in the accuracy of predictions by 10 percent by incorporating skeleton denoising and predicting the locations of missing joints based on the information from adjacent frames. The reliance on 2D poses rather than 3D poses is another shortcoming that should be addressed within the field.

2.3.4 Transformers and Self-Attention Based Approaches

Since the application of Transformers, an architecture designed originally for natural language processing, to computer vision, new opportunities have arisen for modeling long-range temporal dependencies in behavior analysis from videos. In contrast to RNN-based approaches, where the temporal receptive field is limited by the state transition, the self-attention approach allows each frame to attend to all frames at once, potentially modeling any range of temporal dependencies.

Several Transformer-based architectures were proposed for ASD behavioral analysis in recent years. One approach uses a feed-forward Transformer coupled with NFA to classify repetitive head movements on multiple datasets, with an accuracy of 94–95%. Video Swin Transformer, the video modification of the Shifted Window approach of the Swin Transformer, has been utilized for recognition of ASD stimming behavior on an expanded SSBD/ESBD dataset, with the accuracy of 94.43%, F1 score of 96.21% and the use of natural language as an extra modality, which makes it unique in its field.

Attention-based systems have also been introduced into hybrid models rather than being used alone in Transformer networks. The multi-stream architecture proposed by Aldhyani & Al-Nefaie (2025) [38] involves the use of both channel attention (which recognizes the most behaviorally significant feature channels using dual global average and max pooling and then passes them through a shared MLP) and spatial attention (which emphasizes important spatial areas in the frames by means of channel-wise average and max projections and subsequent 7×7 convolutional operations). In addition, the authors introduce an exclusive temporal attention component that calculates frame weights according to a softmax function. The use of attention at multiple levels — on the spatial feature maps created by each CNN branch and on the temporal model constructed by the bidirectional LSTM — reflects the trend towards fine-grained localization.

The limitations of transformer-based systems mainly include the high sample and computation complexities. The self-attention mechanism is quadratic to the number of frames in the sequence, and thus long videos will consume much computing resources. Furthermore, transformer models are much more computation-intensive in nature and need to be trained with significantly larger numbers of labeled samples than CNN models to prevent overfitting issues.

Due to the nature of scarce labeled autism behavior videos available, this limitation becomes quite important since there are generally no more than 100 subjects available in most public datasets.

2.3.5 Explainable AI for ASD Detection

Interpretability is an essential challenge associated with the use of deep learning-based models in the diagnosis of ASD. The difficulty of explaining how a neural network model makes decisions has been described as a black-box problem and represents one of the primary reasons why these models have not yet been adopted in practice, nor have they received regulatory approval or the trust of researchers. XAI methods are a set of emerging techniques used to improve interpretability in machine learning applications.

According to the systematic review conducted by Lawan(2025) [39], SHAP (Shapley Additive exPlanations), LIME (Local Interpretable Model-Agnostic Explanations), and Grad-CAM (Gradient-weighted Class Activation Mapping) are the most common XAI approaches in ASD detection tasks based on a sample of 34 studies using neuroimaging, behavioral, eye-tracking, genomic, and multimodal data.

SHAP, used in seven of the papers, ensures that the scores for feature attributions are consistent worldwide based on cooperative game theory, allowing the determination of the input features, such as the neuroimaging ROIs, behavioral survey features, or genomic markers, that contribute the most to model predictions. Some example applications are the use of SHAP together with XGBoost to determine the MID2 gene as a potential biomarker of ASD across 55 GEO datasets, as well as the determination of muscular strength and gray matter volume as significant predictors of the severity of ASD through multimodal imaging.

The LIME approach, used in six research works, provides local and case-specific interpretations through constructing an approximation of the decision boundary using an interpretable model in the vicinity of each particular prediction. This approach has been especially utilized for behavioral questionnaires data sets (UCI ASD-Test being a prominent example). The major strength of this method lies in the fact that it allows conducting clinical behavioral screenings effectively and accurately at a reduced cost. However, the approach's main weakness concerns its instability between different samples.

Grad-CAM, utilized in four papers, constructs a visual saliency map by calculating the gradient of the class score with respect to the activations of a particular conv layer in order to demonstrate how the input contributes to the model's classification. As such, Grad-CAM allows clinicians to validate brain areas responsible for ASD diagnosis using the available neuroanatomical information. This technique can be applied only to CNNs but not to attention-based or graph-based networks.

Although the importance of the application of XAI techniques in making clinical use of ASD diagnosis is clear, there exist multiple important gaps which were identified by Lawan(2025) [39]. Firstly, no XAI evaluation metric that could be used specifically to analyze whether the explanations produced by a particular XAI approach fit ASD diagnostics was created.

Currently, XAI approaches are evaluated solely on a descriptive level and not by applying quantitative metrics. Secondly, the instability of LIME and high computation cost of SHAP hinder the practical application of those two techniques. Finally, only one of the examined papers uses an XAI technique in combination with the video behavior data.

2.4 Comparing

2.4.1 Traditional Approaches to the Shift towards Deep Learning

Some key similarities between traditional approaches included:

- They were computationally cheap and easy to prototype since they required very few computational resources.
- Their feature representations were also meaningful and semantically connected to behavioral characteristics. This allowed for easier integration with the clinical domain.
- They performed well on straightforward cases with stable recordings.

Despite all these strengths, the limitations of classical methods became so severe as to necessitate the shift towards deep learning. The first failure was their sensitivity to low-level factors such as motion, illumination, background, partial occlusion, making them ineffective under realistic video environments characteristic of ASD assessment in the wild. The second limitation was that the predetermined features could not be automatically adjusted to accommodate the wide range of behaviors that can occur among patients at various development levels and within different cultural backgrounds. Finally, the last but most crucial problem was their inability to cope with the complicated spatiotemporal dynamics involved in abnormal repetitive actions that set them apart from normal infantile behaviors. As pointed out by Yang(2025) [36], deep learning models consistently surpassed the performance of classical baselines across all experimental settings, with the sole exception of MLP with HOF features in the most controlled cases.

2.4.2 Deep Learning Techniques

In reviewing these major architectural classes individually, it is useful to consider them against various criteria, including empirical performance, computational expense, privacy implications, data demands, and ecological validity. In Table 2.1 we present an overview of the critical empirical findings from various studies conducted using the SSBD dataset, which is the most common publicly available test set for autism behavior videos.

Method	Architecture	Dataset	Accuracy	Key Strength
Rajagopalan (2013/2014)	Histogram-based features	SSBD	86.60%	Interpretable baseline
Singh (2025)	CNN–LSTM	SSBD	92.62%	Temporal + spatial fusion
Jabbar (2026)	CNN–GRU	SSBD (4-class)	92.94%	Efficient spatiotemporal
Alkahtani (2023)	LSTM + VGG-19	SSBD	95.00%	Deep temporal modeling
Asmetha & Senthilkumar (2025)	Transformer Network	SSBD	95.01%	Long-range attention
Aldhyani & Al-Nefaie (2025)	Multi-stream CNN + Attention	SSBD	96.55%	Multi-stream + attention
Yang (2025)	**Temporal coherency DNN + SVM**	**SSBD**	**98.30%**	**Self-supervised coherency**
Video Swin Transformer	Transformer + Language	SSBD + ESBD	94.43%	Multimodal, long-range
RGBPose-SlowFast	Dual-pathway 3D CNN	SSBD + HGD	91.00%	Spatial + temporal fusion

Table 2.1: Performance comparison of selected deep learning methods on SSBD and related benchmarks.

CNN-based models incorporating GRU or LSTM temporal components provide a realistic tradeoff between accuracy and feasibility. The CNN-GRU model proposed by Jabbar(2026) [35] reached 92.94% mean cross-validated accuracy with a minimal level of variance ($\sigma = 0.38\%$), indicating high generalization capabilities across the SSBD dataset, and is computationally affordable enough for use in cloud and near-real-time environments. The less complex gates utilized by GRU compared to LSTM further mitigate the risks of overfitting with limited amounts of data, which is essential in view of typical data shortages for ASD.

Models combining spatial and temporal information, such as I3D and SlowFast, demonstrate similar results and are better adapted for working with untrimmed videos that require detection and temporal localization of behavioral events simultaneously. However, significantly higher computational complexity associated with these algorithms hinders their deployment, and their superior performance compared to hybrid CNN-RNN models becomes less evident when applied to short trimmed video fragments specific to SSBD.

The pose-based DL architecture, typified by the DASD multi-stream approach of Aldhyani and Al-Nefaie (2025) [38], performs the best in the area of narrow single behavior binary classification (accuracy of 96.55% for hand flapping compared to normal behaviors), while also being privacy-preserving and less susceptible to unrelated changes in visual appearance. The drawbacks are that they rely on proper pose detection and are currently restricted to binary or highly restricted behavior categories due to lack of sufficient training examples.

Transformer-based architectures have the greatest ability to capture complex temporal dynamics, due to the use of self-attention for global context in time series. They show strong performances (accuracies from 94–95% for Transformer-NFA, 94.43% for Video Swin Transformer) comparable to the best hybrid CNN-RNN models in practice, but still require large amounts of data to achieve these scores. The addition of a language modality along with video data in the Video Swin Transformer model is novel but still experimental.

A common theme throughout all architectural families is that multimodal fusion, either between streams for different modality types like RGB and optical flow, multiple streams using different backbone-based feature extractors, or streams incorporating behavioral video and clinical metadata information, consistently beats the unimodal method. The multi-stream approach adopted by Aldhyani and Al-Nefaie (2025) [38] provides an especially strong example here: in isolation, each stream based on a particular backbone (EfficientNetV2B0 at 75.86%, ResNet50V2 at 89.66%, and DenseNet121 at 93.10%) was significantly beaten by their combination (96.55%), thus showing that complementary features do work together.

Generalizability to real-world clinical video data continues to be the most important issue unsolved by all the proposed architectures. The current standard methodology of training and testing on one of the widely used benchmarks, for instance, SSBD, tests only the within-distribution capabilities of the network. As the Yang(2025) [36] review highlights, there is significant evidence to show that while accuracy is quite high on datasets with specific constraints or control settings, generalization across domains is much less understood.

2.5 Limitations and Challenges of the Research

It is evident from the cumulative results obtained in the previous sections that there are certain structural gaps that characterize the state-of-the-art of the field. It should be mentioned that these gaps are not just methodological drawbacks of specific techniques but represent systematic challenges for dataset development, evaluation, clinical implementation, and ethics.

2.5.1 Lack of Data and Ethical Limitations

Open-source ASD behavioral video datasets are exceedingly rare and limited in size. The SSBD, the leading benchmark, consists of just 75 videos, each 90 seconds long, gathered from YouTube. Other datasets like PAVIS (1,837 sequences), ASDPose (319 tests), and MMASD (1,315 instances) are also present; however, the majority of the former are not freely accessible, and all three have less than 100 observations per subject. The lack of publicly available data is not due to insufficient attention devoted to this issue; rather, there are genuine ethical and legal limitations on collecting video recordings of children with ASD, obtaining parental permission, and using this type of information, which falls under the restrictions of privacy laws. Indeed, several studies mentioned in Yang(2025) [36] required removing videos from their experiments due to privacy issues.

The impact of this limitation is far-reaching; the effects include increased variance, higher

chances of overfitting, reduced statistical power of cross-experiment comparisons, and inability to analyze cross-population generalizability due to small sample sizes. While data augmentation and transfer learning help somewhat in compensating for the issue, they do not resolve the problem of distribution bias inherent to the limited amount.

2.5.2 Lack of Severity Classification

The vast majority of ASD detection algorithms, regardless of their architecture, treat the issue as a binary problem, classifying between ASD and typically developing (TD) subjects. Although this problem formulation works well from an operational perspective, it does not capture the clinical reality of autism spectrum disorders (ASD), which is that ASD is a spectrum with high heterogeneity in symptomatology and severity. As noted by diagnostic tools like the Autism Diagnostic Observation Schedule (ADOS) or the Childhood Autism Rating Scale (CARS), severity can be represented in ASD patients using continuous/ordinal values. Notably, no publicly available ASD behavioral video datasets contain severity values annotated by experts according to the ADOS/CARS criteria; therefore, severity levels cannot be distinguished, which is clinically crucial for diagnosis and intervention plans.

Integrating severity-labeled datasets with advanced deep learning models able to predict multiple severity levels or continuous severity is among the most promising research avenues moving forward. ASDPose, a dataset containing videos, skeletal data, and cognitive assessments from ADOS-2 sessions, is one example of how the link between severity and deep learning models can be explored in future research.

2.5.3 Lack of Good Generalization Across Domains

Another issue that arises in conjunction but not identical to the problem outlined above is the question of how well a model performs across domains. The controlled nature of the recording conditions on which current ASD benchmarks rely (fixed camera positions, cooperative participants, etc.) does not align at all with the free nature of naturalistic video recording (videos taken by parents in home settings, clinical evaluations performed in varying conditions, observation in school settings, for instance). Yang(2025) [36] explicitly highlight this issue, stating that while models may do well on SSBD and PAVIS, they often fall apart when faced with more varied naturalistic videos. Cross-benchmark testing, where a model would be tested on one benchmark based on training on another, is virtually nonexistent in the literature.

2.5.4 Absence of Multimodal Fusion for Video

The current ASD behavioral video models predominantly rely on the analysis of motor behaviors through RGB video recordings and skeletal representation but fail to incorporate the valuable information provided by other modalities. Eye-gaze patterns and eye contact behaviors, key aspects of ASD evaluation clinically, have been examined in isolation based on eye tracking data (refer to the review of Lawan(2025) [39]) but not combined with video motor

analysis. The vocalizations and prosodic patterns that can be identified via audio analysis, facial actions and affects through facial action coding, physiological signals from wearable sensors all have information pertaining to ASD identification but cannot be captured by a skeletal motor analysis alone. The MMBD and DREAM datasets discussed by Yang(2025) [36] contain multiple modalities such as physiology, gaze vector and therapy information, although very rarely incorporated into deep learning models. Only one paper, Video Swin Transformer with Natural Language Descriptions, is an exception to this trend.

2.5.5 Absence of Clinical Validation and XAI Criteria

While much time and effort has been put into creating various ASD behavioral diagnosis AI solutions, none of the proposed models have been clinically validated by being tested against human ratings of autism symptoms in actual clinics. Moreover, no clinical validation is feasible at the present stage due to both regulatory issues associated with the application of artificial intelligence to clinical care and the absence of XAI frameworks allowing for the validation of AI models' diagnostic decisions.

The need for XAI is particularly pressing. According to Lawan et al. (2025) [39], there is no unified XAI assessment framework corresponding to the principles of ASD diagnosis; the effectiveness of the interpretation frameworks is evaluated descriptively, and not clinically; and clinician-oriented feedback is not incorporated into the assessment framework of any of the examined systems. The lack of XAI techniques aimed at video content, which is present only in one paper, is especially important considering the key role of video-based assessment in the area. Taken together, these deficiencies mean that clinicians are unable to estimate the validity, ethicality, or relevance of ASD behavior analyses performed using artificial intelligence.

2.6 Conclusion

This chapter has been organized to give a systematic overview of the state-of-the-art approaches for automatic ASD behavior detection from videos, covering the development trajectory from handcrafted feature extraction and conventional machine learning approaches up to the advanced spatiotemporal models, pose representations, and even attention mechanisms.

While traditional techniques had an advantage of interpretability and computational efficiency, they were unable to reach the levels of robustness and generalizability needed for practical clinical applications. Deep learning models, driven by the inadequacy of predefined feature sets in modeling the spatiotemporally rich dynamic motor behaviors associated with autism spectrum disorder, have enabled a notable increase in accuracy levels. In terms of deep learning models, CNN-RNN networks performed well with relatively reasonable computational resources; spatiotemporal architectures provided more sophisticated temporal modeling capabilities, but at an increased computational cost; deep learning using poses ensured privacy-preserving and background-independent representation of motion signals; and transformer-based models utilized powerful temporal attention mechanisms but consumed significantly more

data and computational resources. The research evidence shows that fusion and multi-stream models outperformed single modality models in recognition accuracy; the DASD method proposed by Aldhyani and Al-Nefaie (2025) [38], which achieved an impressive 96.55% accuracy score on a binary hand-flapping recognition task while maintaining full specificity, is one such model.

The four studies forming the foundation of this review — Yang(2025) [36], Jabbar(2026) [35], Aldhyani and Al-Nefaie (2025) [38], and Lawan(2025) [39] — provide insight into both the state-of-the-art achievements in the domain and the areas where the field is currently lagging. Among the major issues to be solved are the extreme lack of annotated severity levels, clinically validated datasets, cross-domain generalization experiments, integration between different behavioral streams (motion, eye tracking, audio analysis, physiological data), the development of novel approaches to interpreting video XAI models, and the absence of consistent assessment criteria, allowing for comparative evaluations.

This list constitutes the rationale behind the proposed approach and the design space in which it operates. The current study addresses some of these problems by creating an architecture capable of extracting spatiotemporal features from multimodal sources and focusing on behavioral elements using an attention mechanism. This model was tested for different behavioral categories and various conditions of the dataset and was designed with the explainability principle in mind from the very beginning rather than being an afterthought.

CHAPTER 3

IMPLEMENTATION AND METHODOLOGY

3.1 Introduction

Detecting stereotypical self-stimulatory behaviors — commonly referred to as "stimming" — in individuals with Autism Spectrum Disorder (ASD) poses a non-trivial computer vision challenge: the target behaviors are subtle, visually variable across subjects, and occur against uncontrolled real-world backgrounds. This chapter describes the methodology developed to address this challenge, targeting four behavioral categories: ArmFlapping, Headbanging Spinning, and Neutral (non-stimulatory movement).

The methodology presented in this chapter was not arrived at directly, but rather through an iterative process of experimentation, failure analysis, and progressive refinement. Each design decision is grounded in empirical evidence, and the reasoning behind each methodological transition is documented alongside the technical implementation.

3.2 Data Engineering

3.2.1 Data Acquisition

We constructed a composite dataset from three distinct sources. Our first source is the Self-Stimulatory Behavior Dataset (SSBD), a well-known benchmark for projects of this nature. While the paper introducing the SSBD indicates it should contain 75 videos, we found only 56 available in the online database. For our second source, we utilized the Extended SSBD, which contributed 70 videos. Finally, we gathered the remaining 52 videos manually from social platforms such as YouTube, TikTok, and Facebook using queries for autism stimming behavior. This manual collection was conducted to increase variability in recording conditions, including illumination levels and subject demographics.

These three sources provided us with 178 raw videos depicting the target behaviors. Ensuring the appropriate handling of this dataset was a primary concern throughout our research. First, we intended for all materials to be used solely for non-commercial research purposes. Moreover, we paid special attention to ensuring that no personal information was retained regarding the individuals featured in the videos.

3.2.2 Annotation and Standardization

We manually re-annotated all 178 source videos using a custom XML schema we designed to capture the temporal boundaries and metadata of behavioral events. For each annotated segment, we recorded the start time, end time, behavioral category (Arm Flapping, Headbanging, Spinning, or Neutral), and a subjective intensity rating. Re-annotation was deemed necessary for three reasons: first, the existing annotations from SSBD and ESSBD were not consistently formatted; second, we determined that the original annotations were imprecise, with segment boundaries that did not align tightly with the actual onset and offset of behaviors; and third, neither original dataset included annotations for neutral, non-stimulatory movements.

To facilitate efficient and accurate re-annotation, we developed a dedicated Python-based graphical user interface (GUI) annotation tool **Figure 3.1** . The tool enabled frame-precise interactive marking of segment boundaries during video playback, assigning a behavioral category and intensity rating, and export the results directly to our custom XML schema. By using this purpose-built tool, we significantly accelerated the workflow while ensuring consistency across the entire dataset.

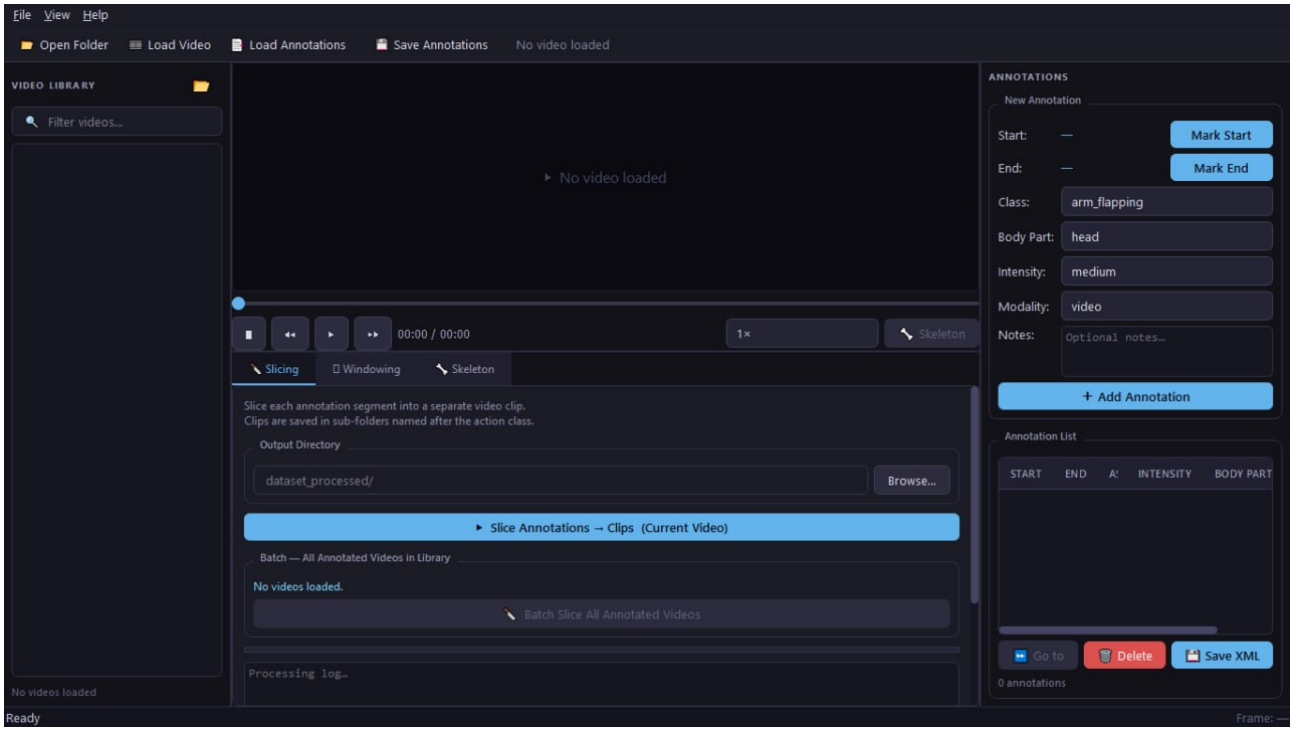

Figure 3.1: interface (GUI) annotation tool

We consider the inclusion of a dedicated Neutral class a critical design decision in our work. We recognized that a classifier trained only on positive behavior examples tends to assign a target label to any input; therefore, by explicitly annotating periods of typical, non-stimulatory motion, we provide the model with a contrasting signal to reduce false positives.

3.3 Initial Prototype (solution 1: Raw Video Features)

3.3.1 Preprocessing

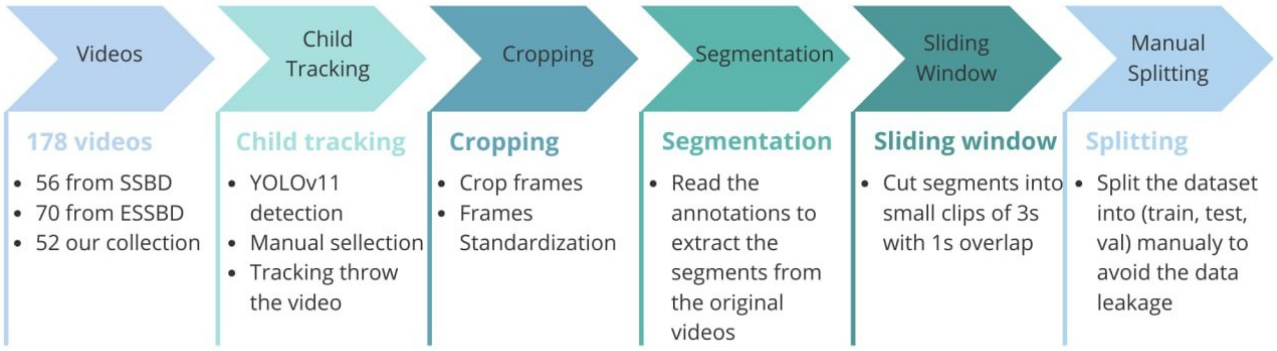

Figure 3.2: Preprocessing pipline

We designed the initial preprocessing pipeline to isolate the subject of interest—typically a child—from potentially cluttered backgrounds. We performed person detection on each video frame using YOLOv11, a state-of-the-art single-stage object detector. Because the videos often contained multiple individuals or background distractors, we determined that an automated detection pass alone was insufficient for reliable subject isolation.

To address this, we introduced a manual bounding box selection step at the start of each video. We visually identified the target child in the frame that contained the largest number of people, and selected the corresponding bounding box from those detected by YOLOv11. This selection initialized a tracking routine where we propagated the chosen bounding box forward through the remaining frames, re-anchoring it to the detector's output at each step. We then cropped the tracked bounding box region from each frame and resized it to a uniform spatial resolution.

Following the cropping process, we performed temporal segmentation by reading the previously generated annotations to isolate specific behavioral events. We then applied a sliding window to these segments, generating 3-second clips with a 1-second stride to augment the sample size while preserving temporal continuity. Finally, we conducted a manual split of the dataset into training, validation, and test sets. We chose this manual splitting strategy specifically to ensure that segments from the same video or subject did not appear in different subsets, thereby strictly avoiding data leakage and ensuring the integrity of our model evaluation.

3.3.2 Feature Extraction and Temporal Modeling

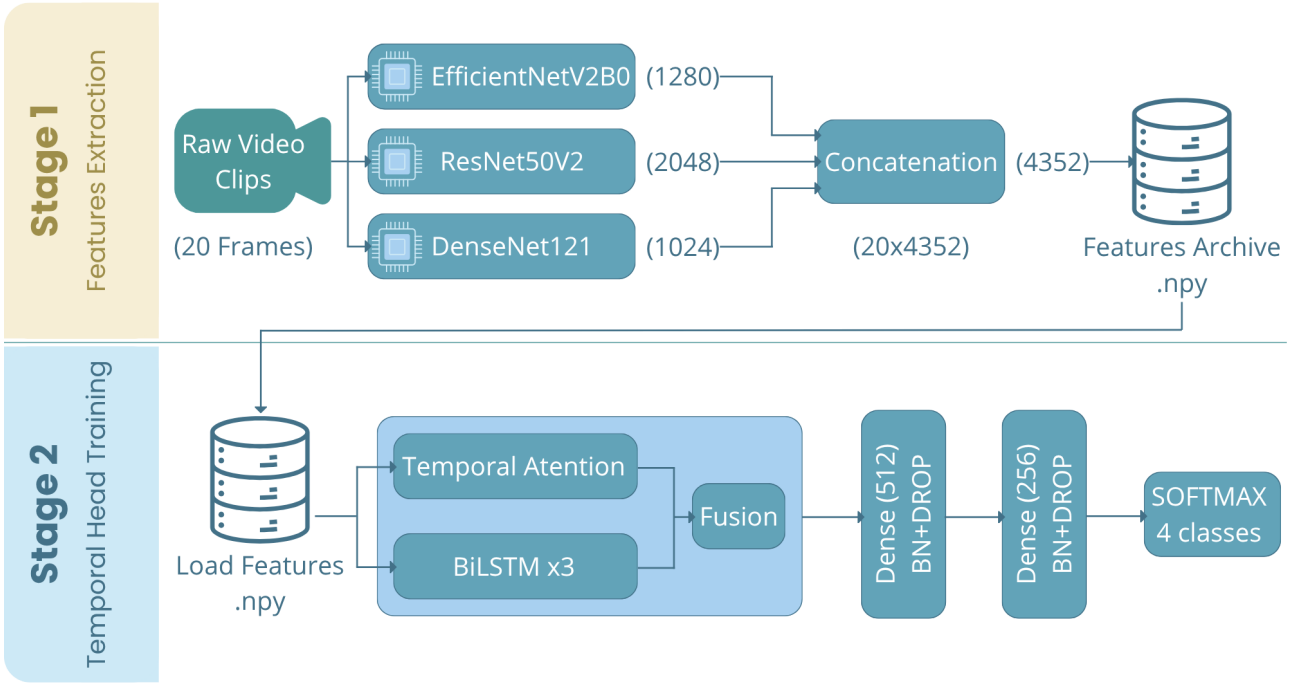

Figure 3.3: solution 01

For the feature extraction stage, we employed a multi-backbone convolutional architecture to capture rich visual representations from each frame. We independently applied three pre-trained CNN backbones to every frame: EfficientNetV2B0 (producing a 1,280-dimensional feature vector), ResNet50V2 (producing a 2,048-dimensional vector), and DenseNet121 (producing a 1,024-dimensional vector). We then concatenated the outputs of these three backbones to form a single composite feature vector with a dimensionality of 4,352 per frame. To accommodate our hardware constraints—specifically a 6 GB VRAM limit—we opted not to hold all three large CNN models in GPU memory simultaneously. Instead, we performed feature extraction sequentially: we used each backbone to process the entire dataset independently and saved the resulting frame-level features as NumPy (.npy) arrays on disk. In the second stage, we loaded and concatenated these pre-computed arrays to produce a final per-clip tensor of shape $(20, 4352)$, representing 20 frames per clip.Our temporal modeling head consumed these per-clip feature tensors. We designed the architecture with three stacked Bidirectional LSTM (BiLSTM) layers, each containing 256 hidden units, and augmented them with a temporal attention mechanism to learn the relative importance of individual frames. We fused the attention output with the BiLSTM output and passed the result through two fully connected layers—Dense(512) and Dense(256). To improve stability and prevent overfitting, we applied Batch Normalization and Dropout regularization to each, while incorporating a residual connection to bridge the two dense layers. Finally, we utilized a Softmax layer with four outputs to produce the class probability distribution.

3.3.3 Failure Analysis: Data Leakage and Quality Loss

We initially reported a solution 1 test accuracy of 96%, a result that appeared, at the time, to represent a major success. However, we did not discover the flaw in this result through a simple inspection of the training pipeline, but rather through empirical evidence. When we evaluated the trained model on a completely external test set—composed of videos that had never been part of any split—the accuracy fell to approximately 50%, barely above the chance level for a four-class problem. This dramatic discrepancy between the internal test accuracy (96%) and external performance (50%) prompted a systematic investigation of our data preparation pipeline, which subsequently revealed the data leakage.

We identified the root cause: we had generated the 3-second clips from source videos before performing the train/validation/test split. Because a single source video yields many overlapping clips that share nearly identical visual content, randomly assigning these clips to different splits resulted in the training and test sets containing clips from the same source videos. In effect, we had been evaluating the model on examples it had already seen in a slightly different temporal window during training. Once we corrected the dataset split—ensuring that all clips derived from a given source video were assigned exclusively to a single partition—we observed the test accuracy drop sharply to approximately 40%.

We also identified a secondary limitation concerning our cropping-and-resizing pipeline. We found that cropping a bounding box region from a standard-definition video and resizing it to the required CNN input dimensions introduced significant resolution loss and blurring. We concluded that this degradation was particularly problematic for the CNN backbones, which rely on fine-grained textural and edge information. The combination of our data leakage correction and these resolution-degraded inputs established a clear performance ceiling, which motivated us to pursue the complete methodological redesign described in the following section.

3.4 Final Methodology (solution 2: Skeletal Pivot)

3.4.1 High-Resolution Pose Estimation

In our solution 2 methodology, we abandoned the cropping-based preprocessing approach entirely. Instead, we applied YOLOv8x-pose—the extra-large variant of the YOLOv8 pose estimation model—directly to full-resolution, uncropped video frames. We selected this model specifically for its superior keypoint localization accuracy. This change allowed us to preserve the original image quality and eliminate the resolution degradation that had constrained our solution 1 feature quality.

We utilized YOLOv8x-pose to detect 17 anatomical keypoints per person, corresponding to the standard COCO body pose skeleton: nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles. For each clip, we ran pose estimation on every frame and retained the resulting keypoint sequences for feature engineering.

Before computing features, we applied a data filtering step to ensure the integrity of our skeletal inputs. We discarded clips from the dataset if:

- YOLOv8x-pose failed to detect any person in one or more frames, resulting in missing keypoints.
- More than one person was detected in any frame, which created ambiguity regarding which skeleton corresponded to our target subject.

While this filtering step reduced the total number of usable clips, we found that it substantially improved the consistency and reliability of our dataset. Table 3.1 presents the distribution of the filtered dataset across splits.

Split	Arm Flapping	Headbanging	Neutral	Spinning	Total
Train	146	140	153	142	581
Validation	30	26	28	30	114
Test	30	30	30	28	118
Total	**206**	**196**	**211**	**200**	**813**

Table 3.1: Dataset distribution after pose-based filtering (solution 2).

In addition to the internal dataset described above, we constructed a completely independent **external test set** to evaluate model generalization. This set consists of 25 raw videos manually collected from social media platforms (YouTube, TikTok, and Facebook), covering all four target behavior categories. The raw video counts per class were: Arm Flapping (50), Headbanging (16), Neutral (37), and Spinning (37). After applying the same filtering pipeline (YOLOv8x-pose followed by discarding clips with no person or multiple persons), we obtained 140 valid clips, all of which passed the filtering successfully. Table 3.2 presents the distribution of this external set. The external test set will be used in **Section 3.7.4** to assess generalization performance.

Class	Raw Videos	Valid Clips
Arm Flapping	50	50
Headbanging	16	16
Neutral	37	37
Spinning	37	37
Total	**140**	**140**

Table 3.2: Distribution of the external test set after pose-based filtering.

3.4.2 Biomechanical Feature Engineering

We transformed the skeletal keypoint sequences extracted by YOLOv8x-pose into a set of hand-crafted biomechanical features designed to capture the distinctive motion signatures of

each target behavior. For a clip of T frames, we generated a feature tensor of shape $(T, 35)$, encoding several categories of motion data.Before we compute any feature, we apply a validity check to every keypoint. We only use a joint in our calculations if its confidence score, as returned by YOLOv8x-pose, exceeds a threshold of 0.15. We treat joints below this threshold as missing and exclude them from relevant computations; we implemented this rule to prevent low-confidence detections from corrupting the feature values. We have organized the 35 features computed per frame into the following groups:

Joint Angles — 5 dimensions

Five joint angles are computed using the law of cosines on the vectors defined by three keypoints per joint, with results reported in degrees:

- **Left elbow:** L shoulder → L elbow → L wrist
- **Right elbow:** R shoulder → R elbow → R wrist
- **Left knee:** L hip → L knee → L ankle
- **Right knee:** R hip → R knee → R ankle
- **Neck angle:** ear → shoulder → hip (using the side with the higher confidence score)

These five angles directly encode the postural configurations that distinguish each behavior — for example, the rhythmic elbow flexion–extension during arm flapping, and the characteristic forward head tilt during headbanging.

Angular Velocities — 5 dimensions

For each of the five joint angles above, the absolute frame-to-frame difference is computed:

$$\Delta\theta_t = |\theta_t - \theta_{t-1}| \tag{3.1}$$

This captures how rapidly each joint angle changes between consecutive frames, encoding the rhythmicity and speed of movement — hallmarks of stimming behaviors that distinguish them from slower, purposeful motion.

Kinetic Energy Proxy — 1 dimension

A scalar measure of overall body motion intensity is computed as the sum of Euclidean displacement magnitudes across all 17 joints between consecutive frames:

$$KE_t = \sum_{j=1}^{17} \|\mathbf{kp}_{t,j} - \mathbf{kp}_{t-1,j}\| \tag{3.2}$$

This provides a single global indicator of how much the entire body is moving at each time step, complementing the joint-specific velocity features.

Per-Joint Velocity Magnitudes — 17 dimensions

For each of the 17 COCO skeleton keypoints, the Euclidean displacement magnitude between consecutive frames is computed individually. If a joint is missing (below the confidence threshold) in either frame, its displacement is set to zero. This yields a 17-dimensional velocity profile that reveals which specific body parts are most active — for instance, high wrist velocity with low lower-body velocity is characteristic of arm flapping.

Center of Gravity — 2 dimensions

The center of gravity (CoG) of the skeleton at each frame is estimated as the confidence-score-weighted average position of all detected joints:

$$CoG_t = \frac{\sum_j w_j \cdot \mathbf{kp}_{t,j}}{\sum_j w_j} \tag{3.3}$$

where w_j is the confidence score of joint j. Only joints above the confidence threshold contribute, ensuring that low-quality detections do not distort the CoG estimate. The result is a 2D coordinate (x, y) representing the body's overall positional centroid in the frame.

CoG Velocity — 2 dimensions

The frame-to-frame displacement of the CoG position is computed as a 2D velocity vector:

$$\Delta CoG_t = CoG_t - CoG_{t-1} \tag{3.4}$$

This captures whole-body translational dynamics across time, which is particularly informative for spinning behavior, where the CoG traces a characteristic oscillatory or rotational trajectory.

Shoulder Rotation Angle — 1 dimension

The orientation of the shoulder axis is computed as the angle of the vector from the left shoulder to the right shoulder keypoint, using the two-argument arctangent function:

$$\theta_t = \arctan 2(\Delta y,\ \Delta x) \tag{3.5}$$

The result, expressed in degrees, captures the in-plane rotation of the torso and is a primary discriminative cue for spinning behavior.

Shoulder Rotation Angular Velocity — 1 dimension

The absolute frame-to-frame change in shoulder rotation angle is computed, with a wrap-around correction applied for angular differences greater than 180° to prevent discontinuities caused by angle periodicity. This scalar captures the rotational speed of the torso and reinforces the spinning detection signal.

Frequency Proxy — 1 dimension

A rolling variance of the kinetic energy proxy is computed over a sliding temporal window. High variance in kinetic energy is indicative of rhythmic, oscillatory motion where bursts of high displacement alternate with brief pauses — a pattern characteristic of arm flapping and headbanging. This feature provides a compact proxy for the periodicity of movement without requiring an explicit frequency estimation.

Normalization

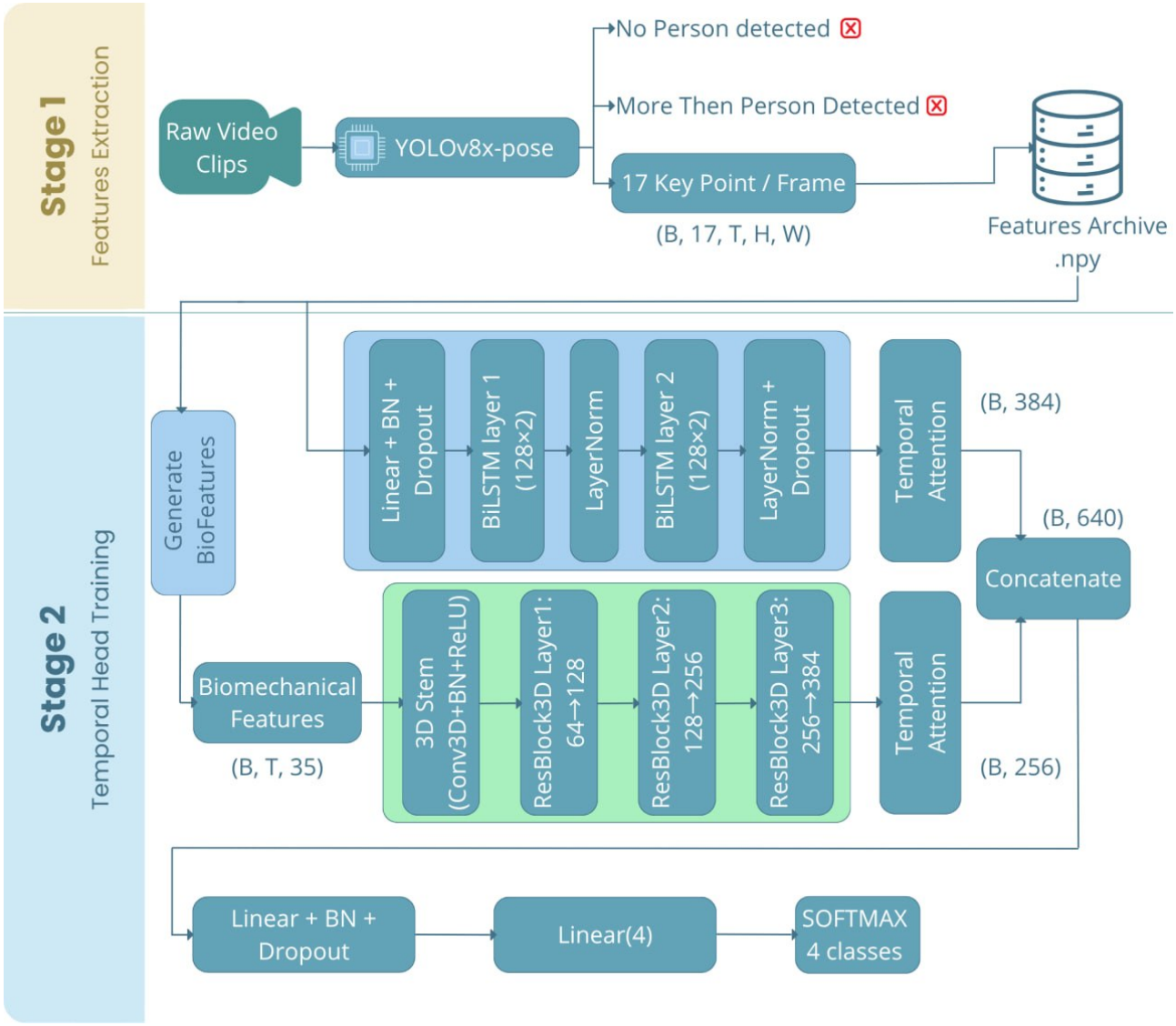

Figure 3.4: solution 02

Two distinct normalization stages are applied:

Stage 1 — Keypoint normalization (before feature computation): Each frame is centered on the hip midpoint (or, if one hip is missing, the visible hip; if both are missing, the mean of all visible joints). The skeleton is then scaled by the torso length — defined as the

Euclidean distance between the shoulder midpoint and hip midpoint — with a fallback to a default scale if the torso length is too small to avoid numerical instability. This makes the pose representation invariant to the subject's distance from the camera and their absolute position in the frame, so that all features reflect body shape and motion rather than camera perspective.

Stage 2 — Biomechanical feature normalization (StandardScaler): The training set features (shape $N \times T \times 35$) are flattened across clips and frames (to $N \cdot T \times 35$) and each feature dimension is clipped to its 1st–99th percentile range computed from the training data, eliminating extreme outlier spikes (for example, in kinetic energy caused by detection jitter). A 'StandardScaler' is then fitted exclusively on the training data to compute the per-feature mean and standard deviation. These statistics are applied to all splits to produce zero-mean, unit-variance features:

$$x' = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{3.6}$$

This ensures that all 35 feature dimensions contribute on comparable scales during model training, improving gradient stability and preventing features with large absolute magnitudes from dominating the learning signal.

3.5 Proposed Dual-Stream Architecture (PoseC3D + BiLSTM)

The final solution 2 model is a dual-stream fusion network that processes two complementary representations of each video clip in parallel: a pose heatmap volume processed by a 3D convolutional PoseC3D-style network, and the hand-crafted biomechanical feature sequence processed by a Bidirectional LSTM. The outputs of both streams are fused prior to final classification into four stereotypy classes: Arm Flapping, Headbanging, Spinning, and Neutral.

3.5.1 Heatmap Stream (3D Convolutional PoseC3D)

The heatmap stream receives a volumetric input of shape $(B, 17, T, H, W)$, where each of the 17 COCO keypoints is represented as a 2D Gaussian heatmap rendered on a spatial grid of $H \times W$ pixels, and T is the number of frames in the clip.

The stream begins with a **3D Stem** — a Conv3D layer followed by Batch Normalization and ReLU activation — which extracts low-level spatiotemporal features from the input volume. The stem output is then passed through a hierarchy of **Residual 3D Blocks (ResBlock3D)**. Each residual block applies a bottleneck design: a 1×1 convolution for channel reduction, a 3×3 convolution for joint spatiotemporal modeling, and a 1×1 convolution for channel expansion. A skip connection is added around each block, with a projection shortcut applied when the spatial dimensions or channel count change.

The residual blocks are organized into three hierarchical layers of increasing channel depth and regularization:

- **Layer 1:** $64 \rightarrow 128$ channels, 2 blocks

- **Layer 2:** 128 → 256 channels, 1 block with dropout
- **Layer 3:** 256 → 384 channels, 1 block with stronger dropout

Following the hierarchical layers, a **Temporal Attention** module is applied. Adaptive temporal pooling summarizes the time dimension, and a softmax attention mechanism learns to weight the contribution of individual frames. The final output of the Heatmap Stream is a **384-dimensional** feature vector per clip.

3.5.2 Bio Stream (Bidirectional LSTM + Attention)

The bio stream receives the 35-dimensional hand-crafted biomechanical feature sequence of shape $(B, T, 35)$, as described in **Section 3.4.2**. The sequence is first passed through a linear projection layer followed by Batch Normalization and Dropout, mapping the raw features into a higher-dimensional learned space. The projected sequence is then processed by two stacked Bidirectional LSTM layers (128 hidden units per direction, yielding 256-dimensional hidden states), with Layer Normalization applied after each BiLSTM layer and a Dropout applied before the second layer. A temporal attention mechanism summarizes the sequence into a fixed **256-dimensional** embedding per clip.

3.5.3 Feature Fusion and Classification

The 384-dimensional embedding from the Heatmap Stream and the 256-dimensional embedding from the Bio Stream are concatenated to form a unified **640-dimensional** representation. This fused vector is passed through a linear projection layer with Batch Normalization and ReLU activation, followed by a Dropout layer for regularization. A final linear classification layer produces the four-class logits.

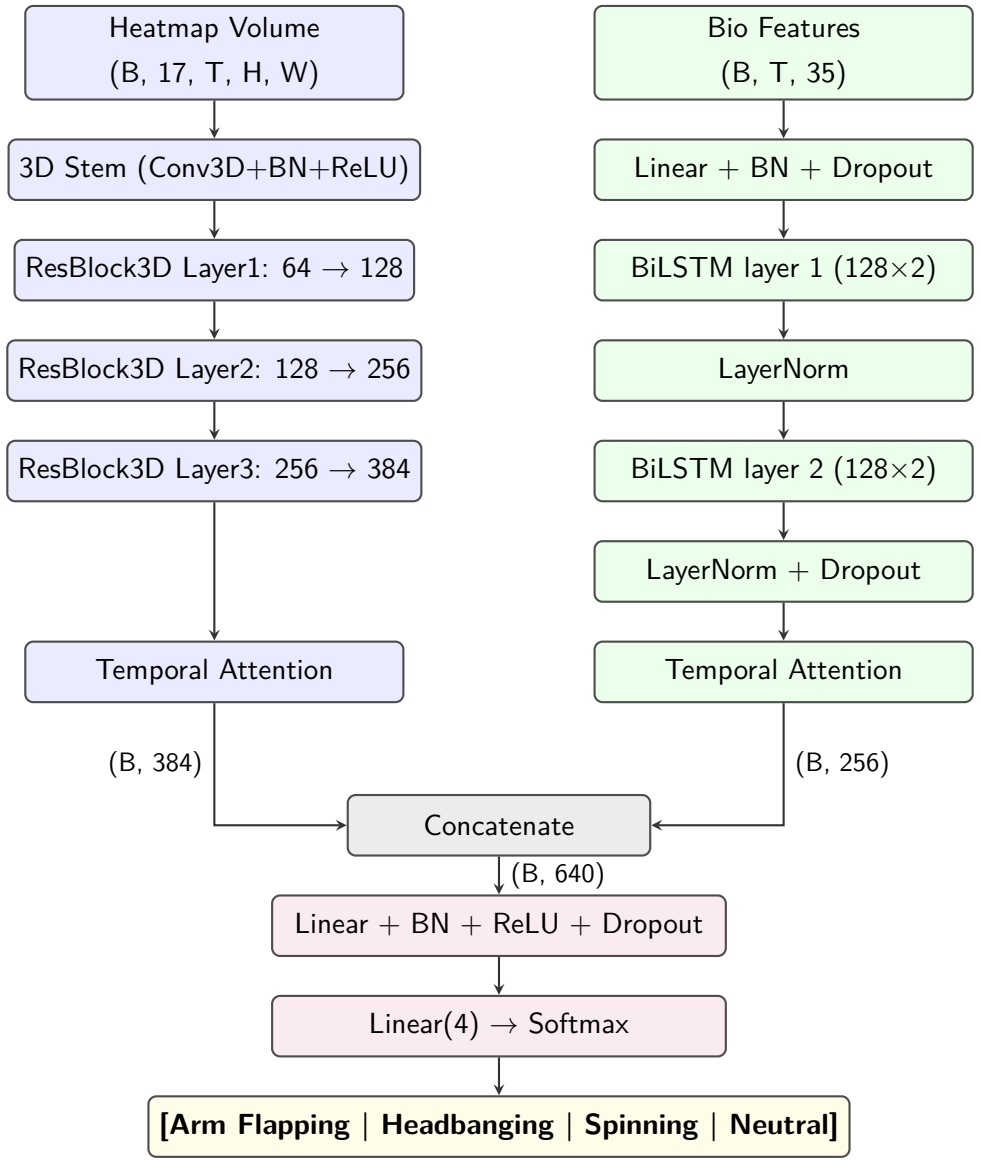

Figure 3.5: Two-stream architecture combining spatial Heatmap Volumes and temporal Bio Features, both utilizing Temporal Attention prior to fusion.

3.5.4 Alternative Architecture: ST-GCN+GRU

We developed an alternative dual-stream architecture that combines graph-based spatiotemporal modeling with recurrent sequence learning to capture complementary information from skeletal heatmaps and biomechanical features.

In the Heatmap Stream, we first encode each joint through a joint embedding module (a Conv3D layer followed by Batch Normalization and ReLU), followed by adaptive average pooling to reduce spatial dimensions. Three ST-GCN blocks then model the spatial relationships between joints and their temporal motion dynamics, with increasing channel depths (64→128→128→256). We finally apply a temporal attention layer to aggregate the most informative time steps into a compact 256-dimensional feature vector.

In parallel, our Biomechanical (Bio) Stream projects the 35-dimensional biomechanical features into a 128-dimensional space using a linear layer with Batch Normalization, ReLU, and Dropout. The projected sequence is then processed by two stacked GRU layers (256 hidden

units each), followed by Layer Normalization and a temporal attention mechanism to obtain a fixed 256-dimensional representation.

Finally, we concatenate the outputs of both streams (resulting in a 512-dimensional vector) and pass them through a fusion module: a linear layer projecting to 256 dimensions, followed by Batch Normalization, ReLU, Dropout, and a final linear classification layer with softmax activation.

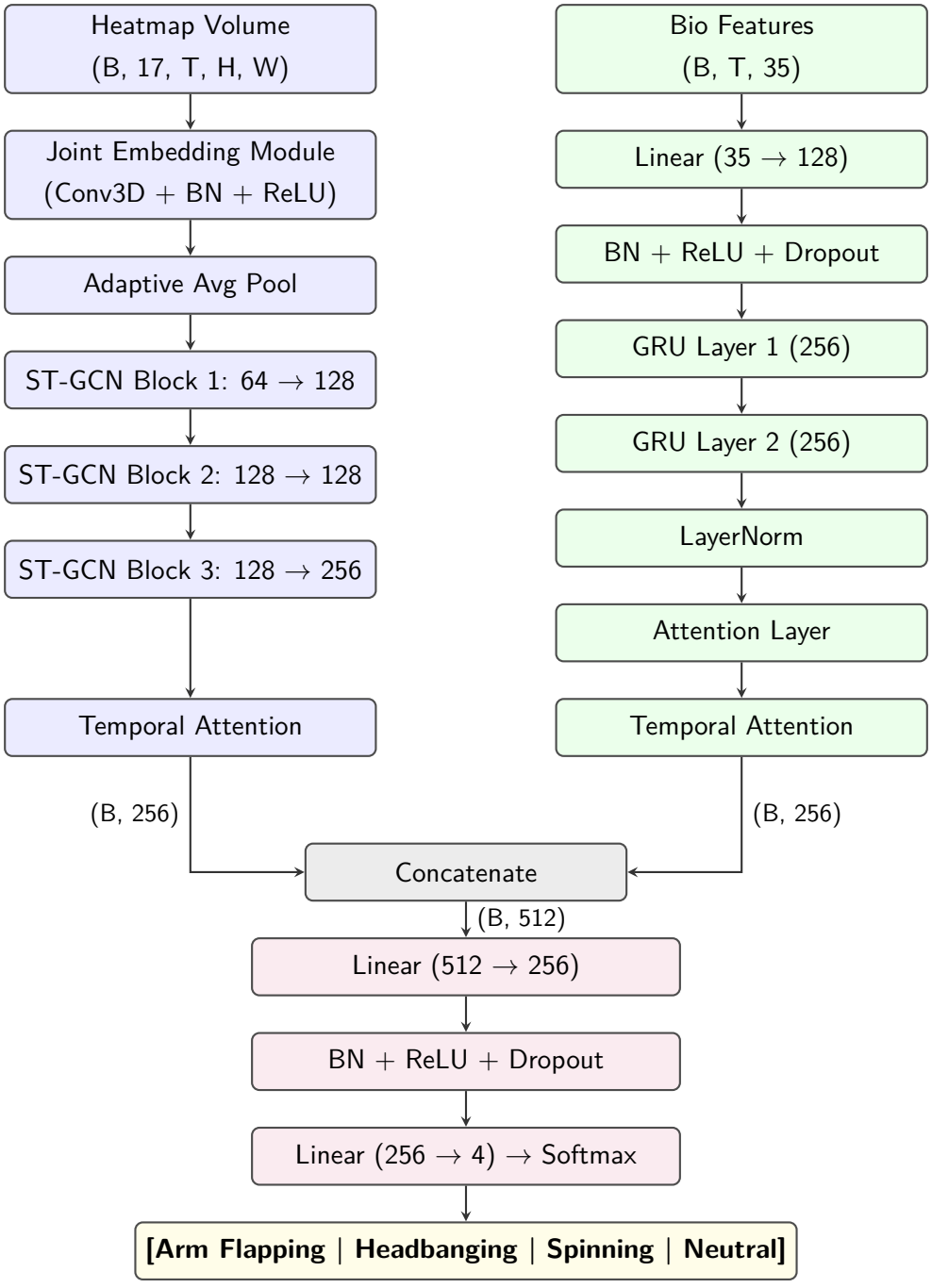

Figure 3.6: Two-stream architecture using ST-GCN for Heatmap Volumes and GRU for Bio Features.

3.6 Experimental Setup and Training

3.6.1 Environment and Hyperparameters

We performed both experiment stages using the following environment configurations. We implemented the first solution using the Google Colab platform, which allowed us to utilize high VRAM GPUs and execute the first feature extraction stage within the TensorFlow/Keras framework. In contrast, we performed the second stage of the experiment on a local PC equipped with an NVIDIA RTX 3060 GPU (6 GB of VRAM), an Intel Core i5 10th-generation CPU, and 24 GB of RAM. Because this limited amount of VRAM made it impossible to continue working with the TensorFlow/Keras framework, we chose the PyTorch framework, as it allowed us to perform fine-tuning of architectural elements with greater precision.

Moreover, the limited VRAM capacity forced us to design a two-stage feature extraction procedure to distribute computations effectively between stages. We used the Adam optimizer for both stages, and we determined the learning rate and batch size empirically through experimentation.

Component	Specification
solution 1 Environment	Google Colab
solution 1 Framework	TensorFlow / Keras
solution 1 GPU	NVIDIA Tesla T4 (15 GB VRAM)
solution 1 CPU	2 vCPU Intel Xeon
solution 1 RAM	∼13 GB
solution 2 Environment	Local Machine
solution 2 Framework	PyTorch
solution 2 GPU	NVIDIA RTX 3060 (6 GB VRAM)
solution 2 CPU	Intel Core i5 (10th Generation)
solution 2 RAM	24 GB
Optimizer	Adam
Dropout Rate	0.4
Clip Duration	3 seconds
Temporal Overlap	1 second
Pose Keypoints	17 (COCO skeleton via YOLOv8x-pose)
Biomechanical Features	35 per frame

Table 3.3: Experimental environment and key hyperparameters.

3.6.2 Validation Strategy

We utilized the corrected dataset split described in **Section 3.3.3** which maintains strict source-video-level separation between training, validation, and test sets—as our baseline evaluation

protocol for all solution 2 experiments.

We conducted a series of experiments to examine the effect of the random seed on model performance. After testing different seed values, we found that results varied non-trivially across seeds, which highlighted the relatively small size of our validation set and the sensitivity of training dynamics to initialization.

To reduce this seed-dependence and improve the generalization of the model, we subsequently adopted a cross-validation strategy. We merged the training and validation sets into a single pool and applied k-fold cross-validation (k=5) over this combined set. We chose this strategy to expose the model to a broader range of training examples during each fold and to produce a more stable performance estimate. While our cross-validation model achieved a comparable accuracy of approximately 65% on the internal test set, we observed that it demonstrated a notably stronger generalization to the external test set, as we discuss in **Section 3.7**.

3.7 Results Summary

In this section, we provide the quantitative outcomes obtained through all experimental settings considered in our work. Specifically, they consist of: (i) BiLSTM+PoseC3D architecture with 5-fold cross-validation, (ii) single split of BiLSTM+PoseC3D using three random seeds (64, 90, and 100), (iii) ST-GCN+GRU architecture with 5-fold cross-validation, and (iv) ST-GCN+GRU architecture with single split and random seed 64. All experimental evaluations were carried out on the internal test set (118 samples), while the best-performing configurations for each of the architectures were additionally tested on an external test set (140 samples).

3.7.1 Comparative Overview of All Experiments

Table 3.4 and 3.5 presents a complete summary of all experimental conditions.

Configuration	Test Accuracy	Test AUC	Best Val AUC	Notes
BiLSTM+PoseC3D (CV mean)	0.615 ± 0.042	0.820 ± 0.024	0.992 ± 0.005	5-fold mean
BiLSTM+PoseC3D (Fold 1)	0.678	0.850	0.994	Single fold
BiLSTM+PoseC3D (Fold 2)	0.568	0.793	0.989	Single fold
BiLSTM+PoseC3D (Fold 3)	0.653	0.821	0.999	Single fold
BiLSTM+PoseC3D (Fold 4)	0.585	0.843	0.983	Single fold
BiLSTM+PoseC3D (Fold 5)	0.593	0.795	0.993	Single fold
BiLSTM+PoseC3D (Seed 64)	0.627	0.811	0.869	Single split
BiLSTM+PoseC3D (Seed 90)	0.602	0.843	0.851	Single split
BiLSTM+PoseC3D (Seed 100)	0.653	0.815	0.907	Single split

Table 3.4: Comparative summary of BiLSTM+PoseC3D experimental configurations (internal test set).

Configuration	Test Accuracy	Test AUC	Best Val AUC	Notes
ST-GCN+GRU (CV mean)	0.639 ± 0.011	0.832 ± 0.021	0.986 ± 0.009	5-fold mean
ST-GCN+GRU (Fold 1)	0.653	0.868	0.991	Single fold
ST-GCN+GRU (Fold 2)	0.653	0.821	0.981	Single fold
ST-GCN+GRU (Fold 3)	0.627	0.828	0.998	Single fold
ST-GCN+GRU (Fold 4)	0.636	0.806	0.971	Single fold
ST-GCN+GRU (Fold 5)	0.627	0.835	0.990	Single fold
ST-GCN+GRU (Seed 64)	0.669	0.852	0.907	Single split

Table 3.5: Comparative summary of ST-GCN+GRU experimental configurations (internal test set).

Note: The exceptionally high AUC values (≥ 0.99) in the "Best Val AUC" column resulted from merging the training and validation sets prior to cross-validation, which introduced source-video leakage between folds.

Observations from the comparison:

- **Best overall test accuracy on internal set:** ST-GCN+GRU (Seed 64) achieved 66.9%, the highest among all configurations.
- **Best overall test AUC on internal set:** ST-GCN+GRU (Fold 1) achieved the highest AUC at 0.868.
- **Cross-validation stability:** ST-GCN+GRU showed lower variance across folds ($\sigma = 0.011$ for accuracy) compared to BiLSTM+PoseC3D ($\sigma = 0.042$), suggesting more consistent performance across different data partitions.
- **Seed sensitivity:** BiLSTM+PoseC3D showed variation across random seeds ranging from 60.2% to 65.3%.

3.7.2 Detailed Results: Best Performing Configuration (ST-GCN+GRU, Seed 64)

Among all configurations evaluated on the internal test set, ST-GCN+GRU with random seed 64 achieved the highest test accuracy (66.9%). Table 3.6 presents the validation and test metrics for this configuration, and Table 3.7 provides the per-class breakdown.

Metric	Validation	Test
AUC	0.907	0.852
Accuracy	—	0.669
Loss	—	0.705

Table 3.6: Best internal configuration (ST-GCN+GRU, Seed 64) results.

Class	Precision	Recall	F1-Score	Support
Arm Flapping	0.800	0.533	0.640	30
Headbanging	0.765	0.867	0.813	30
Neutral	0.719	0.767	0.742	30
Spinning	0.594	0.679	0.633	28
Accuracy	—	—	**0.669**	118
Macro F1	—	—	0.707	118

Table 3.7: Classification report for ST-GCN+GRU (Seed 64) on the internal test set.

3.7.3 Results for Other Notable Configurations

BiLSTM+PoseC3D (Fold 3): The best-performing fold of the BiLSTM+PoseC3D cross-validation achieved 65.3% accuracy and 0.821 AUC. Figure 3.7 presents the per-class breakdown for this configuration.

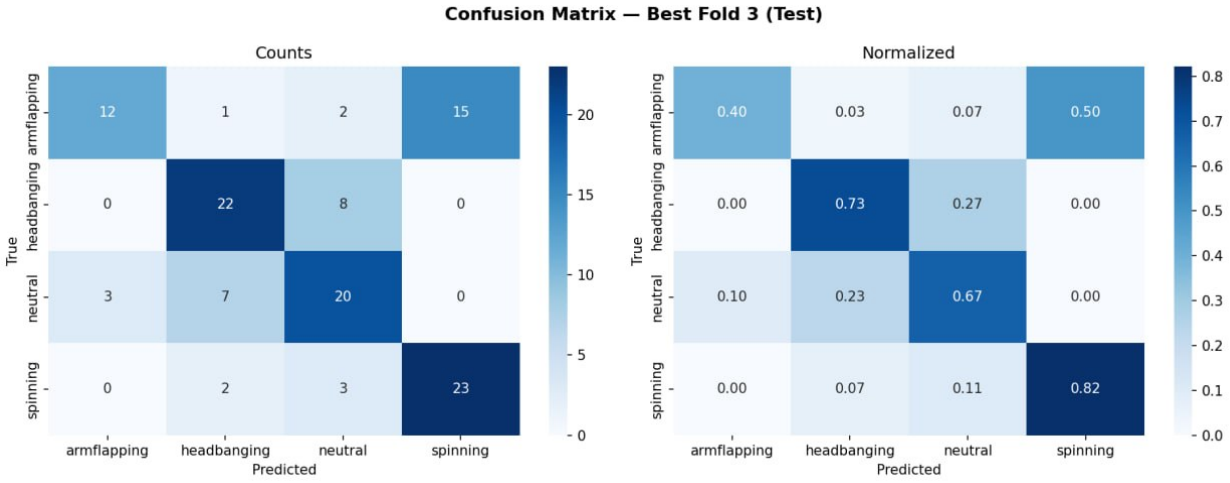

Figure 3.7: Confusion Matrix for BiLSTM+PoseC3D (Fold 3) on the internal test set.

BiLSTM+PoseC3D (Seed 100): The single-split configuration achieved 65.3% accuracy and 0.815 AUC, equal to the best BiLSTM fold. Figure 3.8 presents the per-class breakdown.

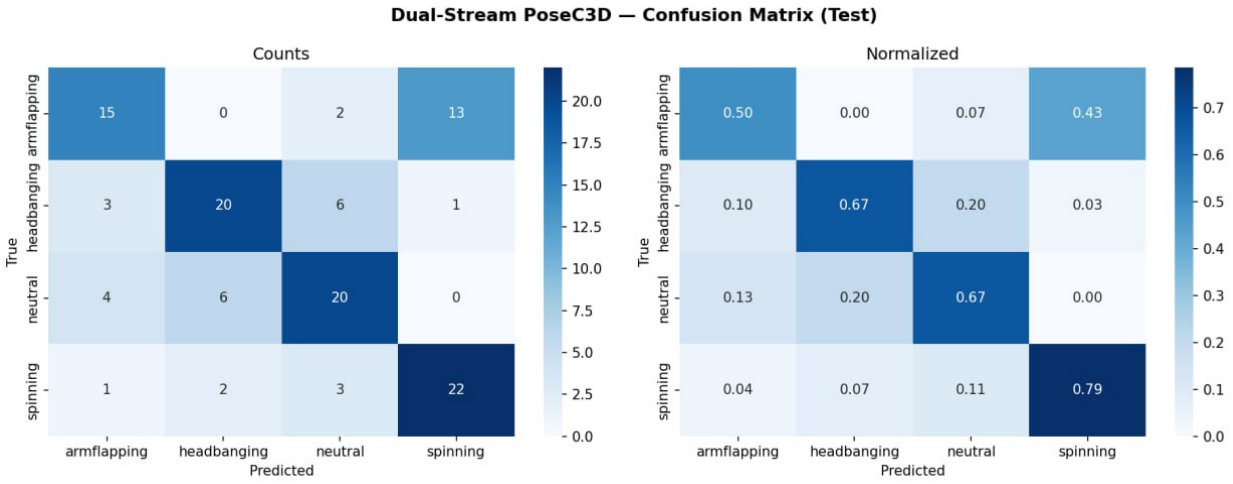

Figure 3.8: Confusion Matrix for BiLSTM+PoseC3D ((Seed 100) on the internal test set.

3.7.4 External Test Set Results

To assess generalization capability, three configurations were evaluated on a fully independent external test set of 140 clips — videos and subjects that were never part of any training, validation, or internal test split:

1. the ST-GCN+GRU cross-validation model.
2. the BiLSTM+PoseC3D cross-validation model.
3. the BiLSTM+PoseC3D single-split model with seed 100.

Model	Test Accuracy	Test AUC
ST-GCN+GRU (cross-validation)	**0.8214**	**0.9567**
BiLSTM+PoseC3D (cross-validation)	0.8143	0.9513
BiLSTM+PoseC3D (Seed 100)	0.8000	0.9342

Table 3.8: External test set summary comparison.

Notably, we observed that all three models performed significantly better when evaluated on the external test set compared to their performance on the internal test set. While we initially regarded this result as unexpected, we can offer several clear explanations for this discrepancy:

1. **Superior Visual Quality:** The external videos possessed a higher level of visual quality, including higher resolution, compared to those in our internal dataset. This factor made the task of skeletal detection much easier for our pipeline.
2. **Reduced Environmental Noise:** We noted that the external videos contained less noise, such as camera shake, background clutter, and occlusions. Consequently, the model could estimate skeletal poses with much higher precision.

3. **Pronounced Behavioral Expression:** The target actions in the external set were expressed more vividly and clearly, which led to the formation of highly distinctive motion patterns.

4. **Consistent Keypoint Visibility:** The subjects' body joints were visible in the video frames more frequently, resulting in fewer keypoint detection errors. As we previously mentioned, our two proposed architectures depend heavily on the quality of the upfront pose detection, and this high-quality input positively influenced their overall performance.

In light of these factors, we suggest that the lower performance on our internal test set does not imply poor generalization ability. Instead, we conclude that it simply highlights the inherent complexity and rugged, real-world nature of our internally collected dataset.

It is clear from the good performance obtained using the test dataset that the representations learned using pose information and biomechanics generalize well to new subjects and recording settings. The findings also stress the significance of the quality of data for skeleton-based behavior classification. It is evident that videos with higher resolution, better body articulation visibility, and low levels of background interference improve the performance of both models studied.

ST-GCN+GRU (Cross-Validation) External Results

Class	Precision	Recall	F1-Score	Support
Arm Flapping	0.974	0.760	0.854	50
Headbanging	0.938	0.938	0.938	16
Neutral	0.611	0.892	0.725	37
Spinning	0.935	0.784	0.853	37
Macro Avg	**0.865**	**0.843**	**0.842**	140

Table 3.9: Classification report for ST-GCN+GRU on the external test set.

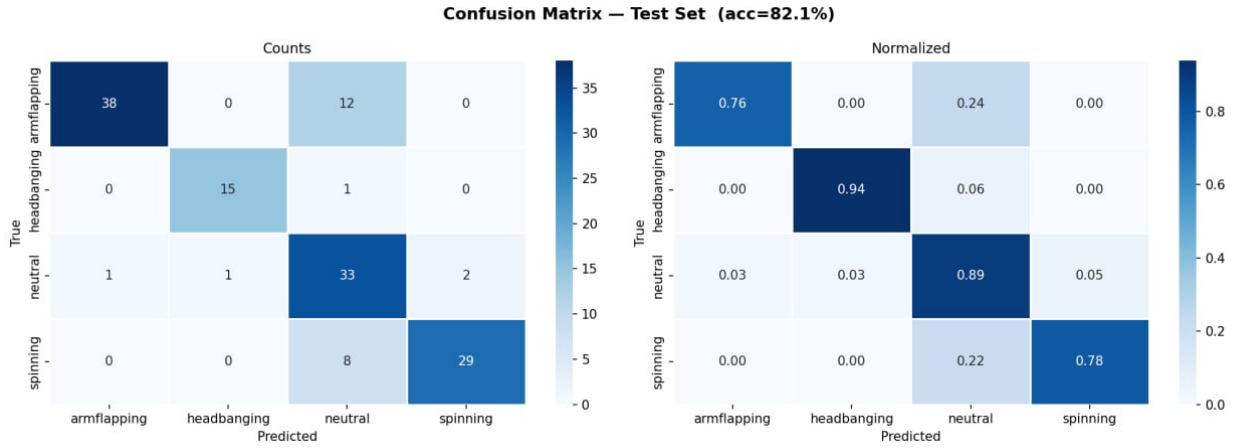

Figure 3.9: Confusion matrix for ST-GCN+GRU on the external test set.

BiLSTM+PoseC3D (Cross-Validation) External Results

Class	Precision	Recall	F1-Score	Support
Arm Flapping	0.848	0.780	0.813	50
Headbanging	0.762	1.000	0.865	16
Neutral	0.718	0.757	0.737	37
Spinning	0.912	0.838	0.873	37
Macro Avg	**0.810**	**0.844**	**0.822**	140

Table 3.10: Classification report for BiLSTM+PoseC3D (cross-validation) on the external test set.

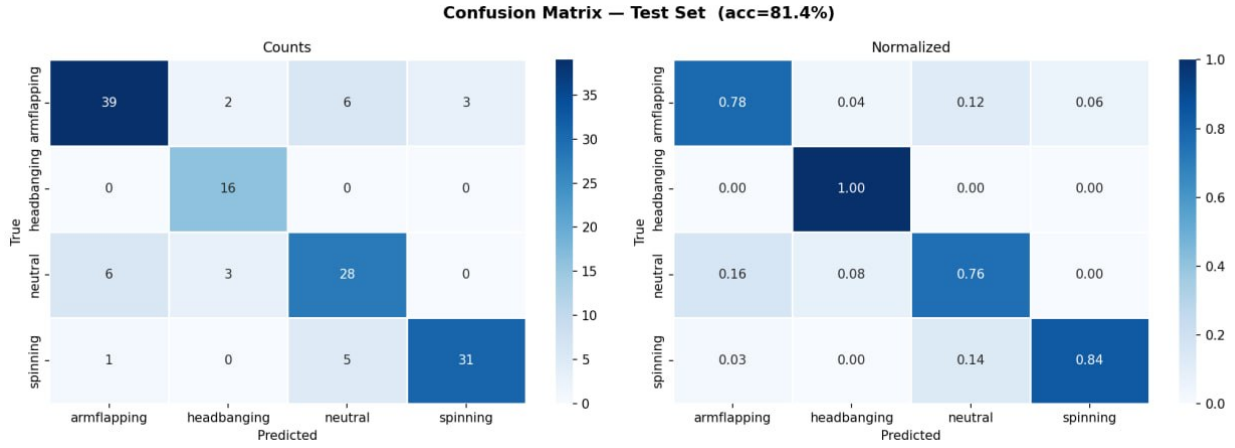

Figure 3.10: Confusion matrix for BiLSTM+PoseC3D (cross-validation) on the external test set.

BiLSTM+PoseC3D (Seed 100) External Results

Class	Precision	Recall	F1-Score	Support
Arm Flapping	0.841	0.740	0.787	50
Headbanging	0.867	0.813	0.839	16
Neutral	0.700	0.757	0.727	37
Spinning	0.829	0.919	0.872	37
Macro Avg	**0.809**	**0.807**	**0.806**	140

Table 3.11: Classification report for BiLSTM+PoseC3D (Seed 100) on the external test set.

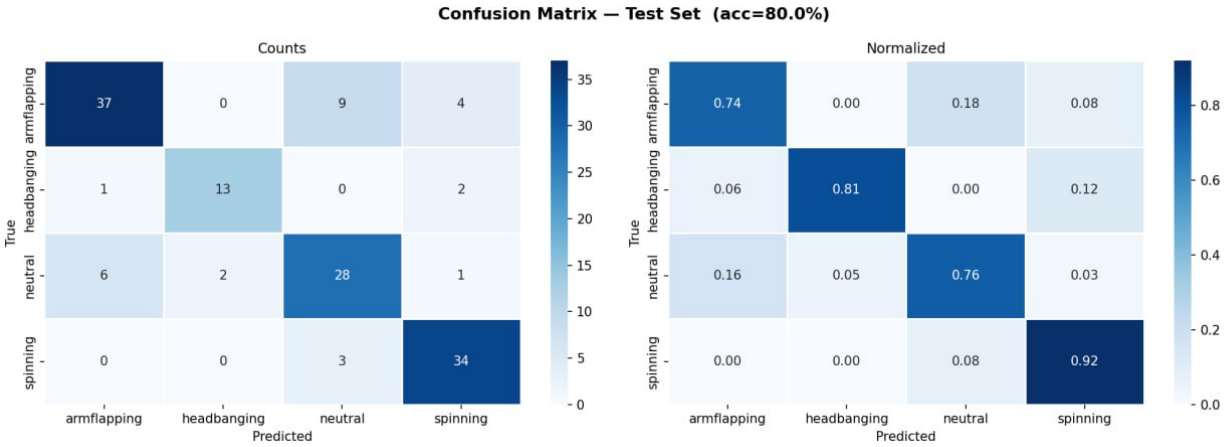

Figure 3.11: Confusion matrix for BiLSTM+PoseC3D (Seed 100) on the external test set.

Comparative Analysis of External Results

All three models demonstrate strong generalization to unseen data, with external accuracy substantially exceeding internal test performance. Key observations include:

Headbanging is the most reliably detected behavior across all models:

- ST-GCN+GRU CV: F1 = 0.938 (15/16 correct)
- BiLSTM+PoseC3D CV: F1 = 0.865 (16/16 correct — perfect recall)
- BiLSTM+PoseC3D Seed 100: F1 = 0.839 (13/16 correct)

Spinning performs very well across all models:

- ST-GCN+GRU CV: F1 = 0.853 (29/37 correct)
- BiLSTM+PoseC3D CV: F1 = 0.873 (31/37 correct)
- BiLSTM+PoseC3D Seed 100: F1 = 0.872 (34/37 correct)

Arm Flapping shows variation across models:

- ST-GCN+GRU CV: Highest precision (0.974), lower recall (0.760)
- BiLSTM+PoseC3D CV: Most balanced (precision 0.848, recall 0.780)
- BiLSTM+PoseC3D Seed 100: Lower recall (0.740)

Neutral is the most challenging class for all models:

- ST-GCN+GRU CV: F1 = 0.725 (low precision 0.611)
- BiLSTM+PoseC3D CV: F1 = 0.737 (most balanced)
- BiLSTM+PoseC3D Seed 100: F1 = 0.727

Common error pattern: All three models tend to misclassify stimming behaviors as Neutral when uncertain, rather than confusing one stimming class for another. This suggests a conservative bias where the model defaults to the non-stimulatory class.

Cross-validation vs. single-split: The cross-validation versions of BiLSTM+PoseC3D (81.4% accuracy) outperform the single-split version (80.0% accuracy), confirming that training on the full combined training+validation set (695 clips) yields better generalization than using only the 581 training clips.

Overall comparison:

- **ST-GCN+GRU CV** achieves the highest accuracy (82.1%) and AUC (0.957)
- **BiLSTM+PoseC3D CV** is a close second (81.4% accuracy, 0.951 AUC)
- **BiLSTM+PoseC3D Seed 100** achieves 80.0% accuracy, 0.934 AUC

The performance gap between the two cross-validation models is negligible (0.7 percentage points in accuracy, 0.005 in AUC), suggesting that both architectures achieve comparable efficacy when trained under optimal conditions.

3.7.5 Summary of Findings

The experimental results across multiple architectures and configurations yield the following key findings:

1. **Best internal performance** was achieved by ST-GCN+GRU with seed 64, attaining 66.9% accuracy on the internal test set.
2. **Best external performance** was achieved by ST-GCN+GRU cross-validation, attaining 82.1% accuracy and 0.957 AUC. BiLSTM+PoseC3D cross-validation achieved 81.4% accuracy and 0.951 AUC — a nearly equivalent result.
3. **Cross-validation consistently outperformed single-split training** for BiLSTM+PoseC3D, with an improvement of 1.4 percentage points on external accuracy.

4. **ST-GCN+GRU demonstrated more stable cross-validation performance** ($\sigma = 0.011$ for accuracy) compared to BiLSTM+PoseC3D ($\sigma = 0.042$) on the internal test set, indicating more consistent performance across different data partitions.

5. **BiLSTM+PoseC3D showed moderate seed sensitivity**, with performance varying by approximately 5 percentage points across different random seeds (60.2% to 65.3%).

6. **Headbanging** was the most reliably detected behavior across all models, with BiLSTM+PoseC3D CV achieving perfect recall (1.000) and ST-GCN+GRU CV achieving the highest F1 (0.938).

7. **Neutral** was the most challenging class for all models, primarily because many stimming behaviors were misclassified as Neutral, suggesting a conservative bias where the model defaults to the non-stimulatory class when uncertain.

8. **External generalization** consistently exceeded internal performance, with improvements of approximately 11–18 percentage points, confirming that the learned biomechanical and pose-based features transfer well to cleaner evaluation conditions.

Taken together, these results establish that both the BiLSTM+PoseC3D and ST-GCN+GRU architectures are highly effective for stimming behavior detection. The two architectures achieve nearly equivalent external performance when trained with cross-validation (82.1% vs. 81.4%), with ST-GCN+GRU showing a slight edge in consistency across internal folds and peak external accuracy.

3.7.6 Underfitting Diagnosis: Model Capacity vs. Data Scarcity

During training of all aforementioned models, we observed a consistent pattern: **training accuracy and validation accuracy converged to similar levels, neither exceeding approximately 70%**, without the characteristic gap indicative of overfitting. This behavior suggests that the model fails to learn the underlying data distribution adequately, even after increasing model capacity.

Problem Analysis: Under normal conditions, overfitting would manifest as high training accuracy (e.g., >95%) coupled with low validation accuracy. In our case, both metrics remained low and close to each other, which is a hallmark of **underfitting**.

Hypotheses tested:

1. **Model is too small:** We increased model capacity through multiple interventions:
    - Doubling the number of hidden units in BiLSTM layers (from 128 to 256)
    - Adding extra fully connected layers

- Increasing PoseC3D channels from 384 to 512

Outcome: No significant improvement in validation accuracy was observed, suggesting that model capacity is not the primary limiting factor.

2. **Data is insufficient:** After ruling out model capacity, we concluded that **the filtered training set size (581 clips)** may be inadequate for training a deep model of this complexity. Each class contains only 140–150 clips — a modest number compared to the requirements for learning rich motion representations using 3D CNNs or LSTMs.

3. **Irreducible task complexity:** An additional factor that cannot be ruled out is the inherent ambiguity of the classification task itself. The 35 biomechanical features, while carefully engineered, may not fully capture the discriminative information required to separate all four classes under all recording conditions. In particular, the boundary between Neutral movement and low-intensity stimming behaviors is inherently ambiguous — certain non-stimulatory motions share kinematic properties with mild arm flapping or subtle headbanging, making misclassification difficult to avoid regardless of model capacity or dataset size. This suggests the presence of an **irreducible error component** intrinsic to the problem, rather than a correctable limitation of the current architecture.

Implications: These findings suggest that the primary challenge in this domain is not architectural design, but rather a combination of **data scarcity** and **inherent task ambiguity**. While larger, higher-quality annotated datasets would be expected to improve performance, a performance ceiling may persist due to the intrinsic overlap between behavioral categories — particularly between Neutral and low-intensity stimming. Addressing this limitation fully may require either richer feature representations beyond the current 35-dimensional biomechanical descriptor, or the acquisition of a substantially larger corpus of high-quality, clinically annotated videos that better capture the full spectrum of behavioral variability across subjects, recording conditions, and stimming intensities.

3.8 Comparison with State-of-the-Art Methods

Method	Architecture	Dataset	Accuracy
Rajagopalan (2013/14)	Histogram features	SSBD	86.60%
Singh (2025)	CNN–LSTM	SSBD	92.62%
Jabbar (2026)	CNN–GRU	SSBD (4-cl.)	92.94%
Alkahtani (2023)	LSTM + VGG-19	SSBD	95.00%
Asmetha & Senthil. (2025)	Transformer Network	SSBD	95.01%
Aldhyani & Al-Nef. (2025)	Multi-stream CNN + Attn.	SSBD	96.55%
Yang (2025)	Temp. coherency DNN+SVM	SSBD	98.30%
Video Swin Transformer	Transformer + Language	SSBD + ESBD	94.43%
RGBPose-SlowFast	Dual-pathway 3D CNN	SSBD + HGD	91.00%
Ours (Internal Test)	**ST-GCN+GRU (Seed 64)**	**Cross-Domain**	**66.90%**
Ours (External Test)	**ST-GCN+GRU (CV)**	**Cross-Domain**	**82.10%**

Table 3.12: Comparison of our results with state-of-the-art deep learning methods for ASD behavior detection.

The results presented in Table 3.12 require careful contextual interpretation. First, a significant portion of the compared methods operate on a binary classification setting (e.g., Armflapping vs. Neutral), while others evaluate on three-class or four-class problems; only Jabbar (2026) and our proposed system explicitly address a four-class scenario, making direct accuracy comparisons inherently non-equivalent, as finer-grained classification tasks are substantially more challenging. Second, several referenced works — notably those reporting high accuracy on the SSBD dataset — do not explicitly describe their data splitting strategy, and some indicate random shuffling prior to train/test partitioning. This practice risks data leakage, as temporally adjacent frames or clips from the same subject may appear in both training and test sets, artificially inflating performance metrics. In contrast, our evaluation protocol enforces a strict subject-independent, cross-domain split, ensuring no overlap between training and test subjects and validating generalization to entirely unseen individuals. Consequently, the 82.10% accuracy achieved by our ST-GCN+GRU model on the external test set should be interpreted not as a lower bound relative to prior work, but as a more rigorous and reproducible estimate of real-world performance.

3.9 Challenges Encountered

We encountered several noteworthy difficulties throughout this study, starting with the challenges of data collection and labeling, and extending to methodological and architectural issues. In this section, we summarize the difficulties we experienced and the key insights we gained from them, as they directly informed how we redrafted our methodology.

3.9.1 Data Acquisition Barriers

One of the most serious problems we encountered in the course of this research project was the shortage of annotated video datasets available for public use on the subject of repetitive behaviors in individuals with ASD. After a detailed review of the existing literature, we managed to find a few mentions of relevant video datasets, which, unfortunately, were not readily accessible.

For instance, we contacted a scientific team in India that had published an article specifically mentioning a stimming behavior dataset. When we requested access, the scientists informed us that, due to institutional reasons, their group could not share their data externally. We recognize that this is a widespread challenge in this research sphere, as medical video databases contain highly sensitive information that often features minors with diagnosed medical conditions.

As such, we developed a combined dataset using three distinct resources. Specifically, we utilized the Self-Stimulatory Behavior Dataset (SSBD), which was expected to contain 75 videos, though we found that only 56 were accessible via the online repository. We also incorporated the Extended Self-Stimulatory Behavior Dataset (ESSBD), which contributed 70 videos. To supplement these primary sources, we collected an additional 52 videos by searching social media platforms such as YouTube, TikTok, and Facebook using the query "autism stimming."

3.9.2 Annotation Challenges

The original annotations from SSBD and ESSBD exhibited two principal limitations. First, the data were not uniformly formatted across both databases; thus, we required a considerable amount of preprocessing to normalize the dataset. Second, and even more importantly, our analysis revealed that the original data labels were inaccurate—the segmentation was not tight around the exact onset and offset of the behaviors, which would have introduced significant label noise into our training data. In addition, we noted that neither original database included data for neutral behavior.

To resolve these issues, we re-labeled all 178 videos using a custom XML format we designed to record the precise temporal boundaries and metadata of each behavioral event. For each annotated segment, we recorded its start time, end time, behavioral category (Arm Flapping, Headbanging, Spinning, or Neutral), and a subjective intensity rating. To streamline this workflow, we developed a custom Python-based GUI annotation tool that allowed us to process

these videos efficiently and precisely.

We found that the inclusion of a dedicated Neutral class proved essential to our methodology. By explicitly annotating periods of typical, non-stimulatory motion, we provided the model with a contrasting signal to reduce false positives—a design decision that significantly improved our downstream classifier's performance.

3.9.3 Data Leakage Pitfall (solution 1)

After preprocessing, we used YOLOv11 to detect and track individuals within the videos. By manually selecting the bounding box of the target child in the initial frame, we were able to crop the video sequence exclusively to the subject of interest. Following this spatial isolation, we performed temporal segmentation based on our annotations, slicing the child's behaviors into segments of 3 seconds in duration with a 1-second temporal overlap between consecutive windows.

This pipeline resulted in four distinct behavioral classes, containing approximately 220 clips per class. To evaluate our models rigorously, we manually conducted a source-video-level split, ensuring that clips derived from the same source video were never shared between the training, validation, and testing partitions. We organized the dataset splits as follows:

- **Training set:** 2,144 clips (Arm Flapping: 612, Headbanging: 604, Neutral: 340, Spinning: 588)
- **Validation set:** 125 clips (Arm Flapping: 31, Headbanging: 31, Neutral: 33, Spinning: 30)
- **Test set:** 132 clips (Arm Flapping: 33, Headbanging: 33, Neutral: 32, Spinning: 34)

Following the dataset organization, we adopted a temporal model architecture utilizing a multi-backbone feature extractor composed of three convolutional neural networks: EfficientNetV2B0, ResNet50V2, and DenseNet121. These were followed by three Bidirectional LSTM (BiLSTM) layers and a temporal attention mechanism. Due to hardware limitations—specifically our 6 GB VRAM constraint—we performed the feature extraction stage sequentially, saving the concatenated frame-level features as a final per-clip tensor of shape $(20, 4352)$ directly to the disk prior to training the temporal modeling head. Our initial evaluation of this model yielded highly encouraging results, achieving an impressive test accuracy of 96%, which we initially perceived as a major success. However, we soon discovered a critical flaw within our training pipeline: despite our rigorous manual partitioning at the source-video level, a subsequent automated step in the pipeline accidentally re-split the pre-extracted clip features completely at random. Because a single source video yields numerous overlapping clips with nearly identical visual content, this random re-splitting inadvertently distributed clips from the same source video simultaneously into both our training and test sets, resulting in severe data leakage.

However, once we corrected our dataset splitting protocol to ensure that all clips originating from a specific source video were isolated within a single partition, our model's test accuracy plummeted drastically to approximately 40%. This experience provided us with a critical lesson in data leakage, demonstrating that we must perform temporal windowing and clip generation strictly post-splitting rather than pre-splitting.

3.9.4 Resolution Degradation and the Skeletal Pivot

In addition to data leakage, we identified a secondary limitation related to our initial cropping and resizing procedure. We observed that cropping the tracked bounding box region from standard-definition video and subsequently resizing it to the required CNN input dimensions resulted in poor image resolution and severe visual blurring. This degradation was especially detrimental to the CNN backbones, as we recognized that they depend heavily on fine-grained textural details and sharp edge information to extract discriminative features.

To resolve this problem, we abandoned the cropping approach altogether. We re-executed our video preprocessing pipeline, maintaining the full resolution of the uncropped frames to preserve original image quality. From these full-resolution videos, we extracted our motion clips—trimmed to 3 seconds in duration with a 1-second temporal overlap—and we passed them directly through the YOLOv8x-pose model for skeletal keypoint estimation.

To ensure that the extracted skeletons remained consistent throughout each sequence, we filtered out clips based on two strict criteria:

1. YOLOv8x-pose failed to detect any individual in one or more frames, which would result in missing keypoints.

2. More than one individual was detected in any single frame, introducing ambiguity as to which skeleton belonged to our target subject.

Although this filtering step decreased the overall size of our dataset, we found that it significantly enhanced the consistency and reliability of our experimental results.

- **Training set:** 581 clips (Arm Flapping: 146, Headbanging: 140, Neutral: 153, Spinning: 142)
- **Validation set:** 114 clips (Arm Flapping: 30, Headbanging: 26, Neutral: 28, Spinning: 30)
- **Test set:** 118 clips (Arm Flapping: 30, Headbanging: 30, Neutral: 30, Spinning: 28)

3.9.5 Summary of Lessons Learned

Several lessons can be learned from these difficulties in applying our findings to future work in this area:

1. **Class balance versus splitting:** Performing temporal windowing after dataset splitting ensures no data leakage, but it makes class balance challenging because different source videos yield different numbers of clips.

2. **Data crop destroys data quality.** Using data at its original resolution allows for better features generation and higher efficiency overall.

3. **Cross-validation beats one-split validation.** Training on combined training + validation set beats one-split training hands down.

4. **External validation is crucial.** It is possible to achieve an unrealistic accuracy when evaluating internal test set.

3.10 Inference Pipeline for Real-World Deployment

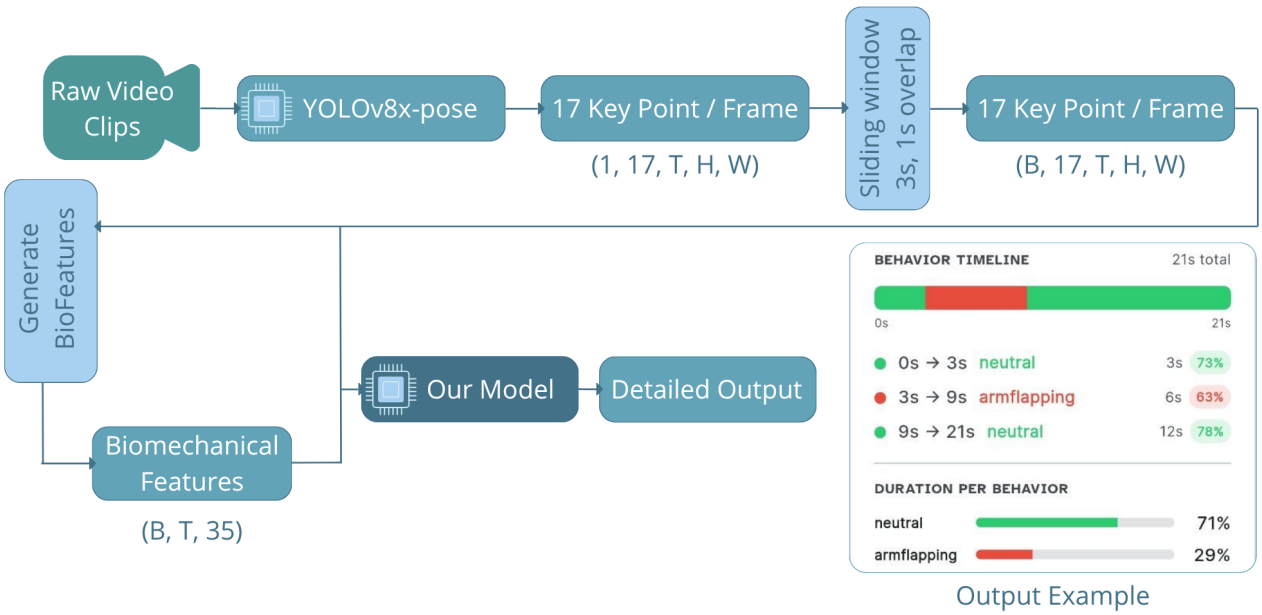

Figure 3.12: Pipeline for Real-World Deployment.

To demonstrate the practical utility of our proposed system, we developed a complete inference pipeline that processes raw videos and produces a temporally segmented behavior timeline. Unlike the training pipeline which operates on fixed 3-second clips, our inference pipeline processes videos of arbitrary length and outputs continuous behavior predictions with temporal boundaries.

Inference Pipeline Steps:

1. **Input Video:** The system accepts an arbitrary-length video file (e.g., MP4, AVI) containing a single individual.

2. **Skeleton Extraction:** Each frame is passed through the YOLOv8x-pose model to extract 17 COCO keypoints for the detected person.

3. **Sliding Window Segmentation:** A sliding window of 3 seconds with a 1-second overlap is applied to the keypoint sequence. Each window is processed independently by our trained model.

4. **Biomechanical Feature Computation:** For each window, we compute the 35 biomechanical features described in **Section 3.4.2**. These features are normalized using the same StandardScaler fitted on the training data.

5. **Behavior Classification:** For each 3-second window, the model outputs a probability distribution over the four classes: Arm Flapping, Headbanging, Spinning, and Neutral. The class with the highest probability is assigned to that window, along with its confidence score.

6. **Temporal Aggregation:** Consecutive windows with the same predicted class are merged into a single segment. The start and end times are adjusted based on the original timestamps, and the confidence score for each segment is computed as the average confidence of its constituent windows.

7. **Output Generation:** The system produces a formatted timeline report showing each behavior segment, its duration, and the model's confidence.

3.11 Conclusion

In this chapter, we have presented a detailed account of the process involved in our development of the automatic detection system for stimming behaviors exhibited by ASD patients. Initially, what seemed like a relatively simple exercise in applying a deep learning approach turned out to be a more complicated process entailing iterative processes of identifying problems, analyzing failures, switching methods, and experimenting with various solutions.

Our first step was to assemble a dataset from three different sources including SSBD, ESSBD, and social media videos, which were annotated with exact timestamps together with a Neutral class label. Our very first model using CNN features and BiLSTM temporal modeling gave us a surprisingly high score of test accuracy, 96%. However, as soon as we realized that the reason behind such high accuracy was simply because we used the wrong data split due to data leakage, our results were completely destroyed and showed only about 40% test accuracy. In addition to data leakage, there is another limitation of the algorithm: the cropping and resizing process was applied to input frames which significantly lowered the resolution of images and made it impossible to identify some minor movements characteristic of the stimming motions. Both of these factors made us completely reconsider our approach.

In particular, the shift in our methodology includes three significant changes. First, cropping is avoided, and we use full resolution video as an input. Second, instead of utilizing features

obtained from pixels directly, we use YOLOv8x-pose algorithm that detects 17 COCO key points in each frame. Third, we define a series of biomechanical features, which include joint angles, angular velocity, center of gravity, shoulder rotation angles, and various frequency parameters corresponding to arm flapping, head banging, spinning, and neutral motions.

A dual stream architecture is proposed where heat maps of 17 key points form a heatmap volume and are processed by 3D convolution neural networks similar to PoseC3D. At the same time, our custom biomechanical features are sent into a bidirectional LSTM classifier. With thorough experimentation, several setups were investigated such as cross-validation, variations of seed values, and different architectural setups (BiLSTM vs. ST-GCN+GRU). These were our main findings:

- Best accuracy on internal testing data was 66.9% (ST-GCN+GRU, Seed 64).
- Cross-validation brought stability and helped reduce the variability of the ST-GCN+GRU model ($\sigma = 0.011$) compared to BiLSTM+PoseC3D ($\sigma = 0.042$).
- External testing demonstrated significantly better performance: the best model obtained 82.1% accuracy and 0.957 AUC on an entirely separate dataset.
- Headbanging appeared to be detected the most accurately (F1 up to 0.938), whereas Neutral was the hardest class for our models because of their tendency towards being conservative about non-stimulation predictions in ambiguous cases.

The difference between internal and external performance metrics of about 15–20 percentage points indicates that our internal testing dataset, composed of various social media videos of rather poor quality, provided even more difficult conditions than external data. However, this result shows that our biomechanical and pose features are indeed strong, and they perform well on a different dataset under favorable testing conditions. We also identified a number of challenges that influenced our methodology including data acquisition challenges, data annotation issues, the issue of data leakage, and the problem of resolution reduction.

In summary, this chapter demonstrates that raw pixel-level features are insufficient for robust stimming detection, both due to resolution constraints and the inherent variability in recording conditions. The transition to skeletal pose estimation and biomechanical feature engineering, implemented within a dual-stream architecture, substantially addresses these limitations, achieving up to 82.1% accuracy on an independent external test set.

GENERAL CONCLUSION

This thesis set out to investigate whether automated, non-invasive detection of stereotypical self-stimulatory behaviours in individuals with Autism Spectrum Disorder is achievable from unconstrained video recordings using deep learning. The answer yielded by the work is a qualified yes—qualified because the results reveal as much about the structural limitations of the problem as they do about the capabilities of the proposed system.

Summary of contributions. The core contribution of this work is a two-stream deep learning pipeline that processes video clips through two complementary representations: a volumetric heatmap stream capturing the spatial configuration of 17 skeletal keypoints across time, and a biomechanical feature stream encoding 35 hand-crafted kinematic descriptors per frame. Two instantiations of this dual-stream design were implemented and evaluated: BiLSTM+PoseC3D, which combines a 3D residual convolutional network with a Bidirectional LSTM, and ST-GCN+GRU, which replaces the 3D CNN with a Spatial-Temporal Graph Convolutional Network.

A dataset of 813 filtered clips was constructed from three sources (SSBD, ESSBD, and manually collected social media videos), re-annotated at frame-level precision with a custom GUI tool, and organised to cover four behavioural classes: Arm Flapping, Headbanging, Spinning, and Neutral. The inclusion of a dedicated Neutral class—absent from existing benchmarks—was a deliberate design decision to provide the classifier with the contrasting non-stimulatory signal necessary to suppress false positives.

On the internal test set of 118 clips, the best configuration—ST-GCN+GRU with seed 64—achieved 66.9% accuracy and 0.852 AUC. On a fully independent external test set of 140 clips collected from subjects and recordings never seen during training, cross-validated ST-GCN+GRU reached **82.1% accuracy and 0.957 AUC**, and cross-validated BiLSTM+PoseC3D achieved 81.4% accuracy and 0.951 AUC. The near-equivalence of these two architectures under optimal training conditions (a gap of less than one percentage point) suggests that both the graph-based and the 3D convolutional representations of skeletal motion encode comparable discriminative information for this task.

Key lessons learned. Perhaps the most instructive outcome of this work is the data-leakage episode encountered during solution 1. A clip-generation pipeline that operated before the source-video-level split inflated test accuracy to a spurious 96%, which collapsed to approximately 40% once the leakage was corrected. This experience reinforces a methodological im-

perative that is frequently underemphasised in the literature: temporal windowing and clip generation must be performed *strictly after* dataset partitioning, with all clips from a given source video confined to a single split.

The 15–20 percentage-point improvement in performance on the external test set relative to the internal set—rather than reflecting overfitting—reflects the inherently greater difficulty of the social-media-sourced internal videos, which exhibit higher rates of occlusion, lower resolution, and more variable recording conditions. This finding underscores both the robustness of skeletal and biomechanical features under favourable conditions and the sensitivity of skeleton-based pipelines to upstream pose estimation quality.

The underfitting pattern observed throughout all training runs—where training and validation accuracy converged at around 65–70% without diverging—points to two concurrent limiting factors: the filtered training set of 581 clips is modest relative to the representational demands of 3D CNN and LSTM models, and the boundary between Neutral movement and low-intensity stimming is intrinsically ambiguous at the kinematic level, constituting an irreducible error component that no architectural modification alone can resolve.

Limitations. Several limitations constrain the scope of the conclusions drawn here. First, the datasets used—SSBD, ESSBD, and the manually collected supplement—were sourced exclusively from social media platforms, introducing a selection bias that may not represent the full behavioural and demographic diversity of individuals with ASD. Second, the external test set, while drawn from subjects unseen during training, originates from the same social media distribution and therefore does not constitute a true clinical validation. Third, the system classifies four discrete behaviour categories and does not address severity estimation, multi-subject scenes, or real-time deployment constraints. Finally, no comparison against human clinician performance or gold-standard diagnostic scores (ADOS-2, CARS) was conducted, which remains a prerequisite for any clinical translation.

Future directions. Several avenues emerge naturally from the limitations identified above. The most immediate priority is dataset expansion: acquiring or constructing a clinically annotated video corpus with ADOS-2-derived severity labels and strict ethical safeguards would substantially improve both the training signal and the clinical relevance of the evaluation. Architectural extensions worth exploring include transformer-based temporal attention over the skeletal graph, multimodal fusion of motor and acoustic features, and test-time augmentation strategies to improve robustness under low-quality recording conditions. From a methodological standpoint, integrating Explainable AI (XAI) mechanisms— such as Grad-CAM adapted to 3D convolutional streams or attention-weight visualisation over the skeletal graph—would make the system's predictions interpretable to clinicians, which is a necessary step toward any regulatory pathway. Finally, a prospective clinical study comparing the system's output against ADOS-2 ratings on a controlled cohort would provide the external validity that the current work is unable to establish.

Ethical implications and responsible deployment. While the technical results of this the-

sis are encouraging, the deployment of any AI-based system for ASD screening carries profound ethical responsibilities that must be addressed before any clinical translation can be considered. Several dimensions deserve careful attention.

First, the risk of false positives and their downstream consequences must not be underestimated. A system that incorrectly flags a neurotypical child as exhibiting ASD-associated behaviours could trigger unnecessary clinical referrals, parental anxiety, and—in under-resourced healthcare settings—misallocation of scarce specialist time. Conversely, false negatives may provide false reassurance and delay intervention. For these reasons, the system proposed in this thesis should be explicitly positioned as a screening aid, not a diagnostic tool, and its output should always be reviewed by a qualified clinician before any decision is made.

Second, algorithmic bias represents a structural concern. The datasets used in this work—SSBD, ESSBD, and manually collected social media videos—are not demographically representative of the global ASD population. They skew toward certain age groups, recording conditions, and cultural contexts. A model trained on such data may perform unevenly across children of different ethnicities, body types, or cultural movement norms, potentially disadvantaging already underserved populations. Any future deployment must include systematic bias auditing across demographic subgroups.

Third, the use of skeletal pose representations, while privacy-preserving by design, does not eliminate all privacy risks. Gait patterns and movement signatures can, in principle, serve as biometric identifiers. Any system storing or transmitting skeletal data—even in anonymised form—must implement appropriate data minimisation, encryption, and retention policies.

Finally, the question of clinical accountability must be resolved before deployment. When an AI-assisted screening tool contributes to a clinical decision that leads to harm—whether through a missed diagnosis or an incorrect flag—it must be clear which party bears responsibility: the developer, the clinician, or the institution. The absence of a recognised regulatory framework for AI-based neurodevelopmental screening tools in most jurisdictions means that this question remains open, and researchers have a duty to advocate for its resolution rather than proceeding as though it does not exist.

Closing remarks. Despite these limitations, the results of this thesis demonstrate that pose-based, biomechanically informed deep learning is a viable and principled approach to automated stimming detection. The skeletal representation enforces privacy by design, removes confounding environmental factors, and produces features that generalise across recording conditions to an extent that pixel-level representations cannot. The dual-stream architecture leveraging both spatial heatmap volumes and temporal biomechanical sequences consistently outperformed single-stream baselines, confirming the complementary value of the two representational modalities. These findings contribute a concrete, reproducible methodological foundation for future work in video-based ASD screening.

REFERENCES

[1] American Psychiatric Association. (2022). *Diagnostic and statistical manual of mental disorders* (5th ed., text rev.). American Psychiatric Association Publishing. https://doi.org/10.1176/appi.books.9780890425787

[2] Estes, A., Munson, J., Rogers, S. J., Greenson, J., Winter, J., & Dawson, G. (2015). Long-term outcomes of early intervention in 6-year-old children with autism spectrum disorder. *Journal of the American Academy of Child & Adolescent Psychiatry*, 54(7), 580–587. https://doi.org/10.1016/j.jaac.2015.04.005

[3] Choi, J., Lee, S. Y., Kim, B., & Park, H. (2025). Automated AI-based identification of autism spectrum disorder from home videos. *npj Digital Medicine*, 8(1), 1–12. https://doi.org/10.1038/s41746-025-01993-5

[4] Tan, D. W., Featherston, R., Zwi, K., & Russell, J. (2025). A systematic review and meta-analysis of autism screening and diagnosis in children using video-assisted telehealth technology. *Digital Health*, 11, 20552076251386705. https://doi.org/10.1177/20552076251386705

[5] World Health Organization. (2019). *International statistical classification of diseases and related health problems* (11th ed.). https://icd.who.int/

[6] Charman, T., & Gotham, K. (2013). Measurement issues: Screening and diagnostic instruments for autism spectrum disorders—lessons from research and practice. *Child and Adolescent Mental Health*, 18(1), 52–63. https://doi.org/10.1111/j.1475-3588.2012.00664.x

[7] Fournier, K. A., Hass, C. J., Naik, S. K., Lodha, N., & Cauraugh, J. H. (2010). Motor coordination in autism spectrum disorders: A synthesis and meta-analysis. *Journal of Autism and Developmental Disorders*, 40(10), 1227–1240. https://doi.org/10.1007/s10803-010-0981-3

[8] Jones, W., & Klin, A. (2013). Attention to eyes is present but in decline in 2–6-month-old infants later diagnosed with autism. *Nature*, 504(7480), 427–431. https://doi.org/10.1038/nature12715

[9] Manfredi, M., Billeci, L., Muratori, F., & Narzisi, A. (2024). Automatic recognition of facial expressions in autism spectrum disorder: A systematic review and meta-analysis. *IEEE Transactions on Affective Computing*, 15(2), 623–641.

[10] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[11] Buduma, N., Buduma, N., & Papa, J. (2022). *Fundamentals of deep learning* (2nd ed.). O'Reilly Media.

[12] Shanthini, S., Sindhana Devi, S., & Suriya Grace, G. (2024). A comprehensive review of learning rules and architecture of perceptron in artificial neural networks (ANNs). Taylor & Francis.

[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[14] DTStack. (2025). *Deep learning model optimization: Convolutional neural network architecture analysis* [Technical report].

[15] LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 255–258). MIT Press.

[16] Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1237–1242).

[17] Ghazanfar Latif, Jaafar Alghazo, Majid Ali Khan, Ghassen Ben Brahim, Khaled Fawagreh, Nazeeruddin Mohammad. Deep convolutional neural network (CNN) model optimization techniques—Review for medical imaging[J]. AIMS Mathematics, 2024, 9(8): 20539-20571. doi: 10.3934/math.2024998

[18] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[19] Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (pp. 92–101). Springer. https://doi.org/10.1007/978-3-642-15825-4_10

[20] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)* (Vol. 25, pp. 1097–1105).

[21] Rajagopalan, S. S., Dhall, A., & Goecke, R. (2013). Self-stimulatory behaviours in the wild for autism diagnosis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)* (pp. 755–761).

[22] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211. https://doi.org/10.1207/s15516709cog1402_1

[23] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. https://doi.org/10.1109/72.279181

[24] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

[25] Cho, M., Kim, C., Jung, K., & Jung, H. (2022). Water level prediction model applying a long short-term memory (LSTM)–gated recurrent unit (GRU) method for flood prediction. *Water*, 14(14), 2221.

[26] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

[27] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).

[28] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)* (Vol. 27, pp. 568–576).

[29] Stark, Z., et al. (2021). Automated objective measurement of motor stereotypies in children with autism spectrum disorder. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29, 1234–1243.

[30] Min, J., et al. (2024). Privacy-preserving action recognition via motion-only representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).*

[31] Madaan, V., et al. (2023). *YOLOv8: A comprehensive review of the state-of-the-art in object detection and human pose estimation* [arXiv preprint].

[32] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788).

[33] Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLOv8* [Computer software]. https://github.com/ultralytics/ultralytics

[34] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014* (pp. 740–755). Springer. https://doi.org/10.1007/978-3-319-10602-1_48

[35] Jabbar, M. K., et al. (2026). Deep learning based approach for behavior classification in diagnoses of autism spectrum disorder using naturalistic videos. *Frontiers in Computational Neuroscience*.

[36] Yang, Y., et al. (2025). Early diagnosis of autism: A review of video-based motion analysis and deep learning techniques. *IEEE Access*.

[37] Prakash, V. G., et al. (2025). RGBPose-SlowFast network for autism spectrum disorder behavior recognition.

[38] Aldhyani, T. H. H., & Al-Nefaie, A. H. (2025). DASD — Diagnosing autism spectrum disorder based on stereotypical hand-flapping movements using multi-stream neural networks and attention mechanisms. *Frontiers in Physiology*.

[39] Lawan, A. A., et al. (2025). A systematic literature review on the application of explainable AI approaches in the assessment of autism spectrum disorder. *OBM Neurobiology*.

ANNEXES

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليـــم العالـــي والبحـث العلمـــــي
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique

كلية العلوم والتكنولوجيا
قسم الرياضيات والإعلام الآلي

Faculté des Sciences et de la Technologie

Département Des Mathématiques & de l'Informatique

جامعـــة غردايـــة

الرقم .....2026/ ق.ر.ا /ك.ع.ت/ج.غ

شهادة الترخيص بالإيداع

أنا الأستاذ(ة):

Mme. BRAHIM Nacera

بصفتي مشرف (ة) والمسؤول(ة) عن تصحيح مذكرة تخرج ماستر الموسومة بـــــــ

A Deep Learning based Autism Stimming Detection from video : Skeletal Motion Analysis

من انجاز الطالب(ة):

YOUB Bassaid

والطالب(ة):

LATRECHE Brahim

الكلية: العلوم والتكنولوجيا.
القسم: الرياضيات والاعلام الالي.
الشعبة: اعلام الي.
التخصص: الأنظمة الذكية لاستخراج المعارف.
تاريخ التقييم/المناقشة: **24/06/2026**
أشهد ان الطالب (الطلبة) قد قام (قاموا) بالتعديلات والتصحيحات المطلوبة من طرف لجنة المناقشة وان المطابقة بين النسخة الورقية والالكترونية استوفت جميع شروطها.

امضاء المسؤول عن التصحيح

مصادقة رئيس القسم

رئيس قسم الرياضيات و الإعلام الآلي
الحاج موسى بلمين

جامعة غرداية
قسم الرياضيات و الاعلام الآلي
كلية العلوم والتكنولوجيا