

Kinjal Raykarmakar

A Dual-Stream Cross-Attention Framework for fine-grained Yoga Pose classification using Spatiotemporal visual cues and...

 Research Papers

Document Details

Submission ID

trn:oid:::27535:140533100

Submission Date

May 26, 2026, 6:16 PM GMT+5:30

Download Date

May 26, 2026, 6:22 PM GMT+5:30

File Name

24ISY19_Kinjal Raykarmakar_TurnItInSubmission.pdf

File Size

3.7 MB

59 Pages

16,169 Words

97,428 Characters

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Exclusions

- 1 Excluded Match

Match Groups

-  **58 Not Cited or Quoted 3%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **40 Missing Citation 2%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 1%  Internet sources
- 2%  Publications
- 5%  Submitted works (Student Papers)

Match Groups

- 58 Not Cited or Quoted 3%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 40 Missing Citation 2%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 1% Internet sources
- 2% Publications
- 5% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	Delhi Technological University on 2026-05-20	2%
2	Student papers	University of Leeds on 2026-05-01	<1%
3	Internet	www.mdpi.com	<1%
4	Student papers	Indian Institute of Technology Roorkee on 2026-05-22	<1%
5	Student papers	King's College on 2025-08-06	<1%
6	Internet	export.arxiv.org	<1%
7	Student papers	Imperial College of Science, Technology and Medicine on 2025-09-18	<1%
8	Student papers	Liverpool John Moores University on 2025-03-17	<1%
9	Publication	Ziyuan, Huang. "Deep Representation Learning for Video Foundation Models.", N...	<1%
10	Student papers	University of Technology, Sydney on 2024-10-31	<1%

11	Student papers	National Forensic Sciences University on 2026-04-22	<1%
12	Student papers	Gisma University of Applied Sciences GmbH on 2026-03-27	<1%
13	Student papers	Higher Education Commission Pakistan on 2026-05-23	<1%
14	Internet	doaj.org	<1%
15	Student papers	University College London on 2016-04-04	<1%
16	Student papers	Kaunas University of Technology on 2026-05-22	<1%
17	Publication	"Advances in Multimedia Information Processing – PCM 2018", Springer Science a...	<1%
18	Student papers	National Institute of Technology, Rourkela on 2026-05-11	<1%
19	Student papers	City University of Hong Kong on 2022-02-06	<1%
20	Publication	Hao Wu, Shui Yu, Zhengliang Hu, Zhongtao Liu, Jingxuan Wang, Hongna Zhu. "Dy...	<1%
21	Student papers	University of Glasgow on 2026-04-29	<1%
22	Student papers	University of Northumbria at Newcastle on 2020-01-28	<1%
23	Student papers	Arab Open University on 2026-05-14	<1%
24	Student papers	Brunel University on 2026-04-17	<1%

25	Student papers	George Washington University on 2026-03-16	<1%
26	Student papers	Loughborough University on 2022-09-05	<1%
27	Student papers	UT, Dallas on 2020-11-09	<1%
28	Student papers	University of Dundee on 2026-04-17	<1%
29	Student papers	University of Glamorgan on 2024-09-30	<1%
30	Student papers	University of Lancaster on 2025-06-13	<1%
31	Student papers	University of Sheffield on 2026-05-07	<1%
32	Internet	escholarship.org	<1%
33	Internet	mediaverse-project.eu	<1%
34	Student papers	Cranfield University on 2021-08-18	<1%
35	Publication	Naif Almughamisi, Gibrael Abosamra, Adnan Albar, Mostafa Saleh. "Hybrid Conv...	<1%
36	Student papers	University of Northumbria at Newcastle on 2023-05-16	<1%
37	Internet	mdpi-res.com	<1%
38	Student papers	City University on 2020-01-24	<1%

39	Student papers	Higher Education Commission Pakistan on 2025-02-19	<1%
40	Student papers	Kotebe University on 2023-09-18	<1%
41	Student papers	University of Edinburgh on 2019-08-21	<1%
42	Student papers	Vellore Institute of Technology on 2026-05-13	<1%
43	Publication	Ying, Tan Ying. "Fish Species Recognition Using Convolutional Neural Network", U...	<1%
44	Publication	"Computer Vision – ECCV 2016", Springer Science and Business Media LLC, 2016	<1%
45	Publication	Hyland, Benjamin Alexander. "Citrullus amarus Drought Tolerance: An Above and...	<1%
46	Publication	Włodzimierz Kasprzak, Bartłomiej Jankowski. "Light-Weight Classification of Hum...	<1%
47	Student papers	The University of Manchester on 2020-01-05	<1%

A DUAL-STREAM CROSS-ATTENTION FRAMEWORK FOR FINE-GRAINED YOGA POSE CLASSIFICATION USING SPATIOTEMPORAL VISUAL CUES AND 3D SKELETAL GEOMETRY

Kinjal Raykarmakar

ABSTRACT

The accurate automated recognition of yoga postures from a single video sequence presents itself as an intricate problem, lying in the intersection between computer vision and biomechanics, and holds enormous potential towards achieving superior fitness training, physical therapy, and man-machine interaction applications. Existing techniques are mainly based on either appearance features, susceptible to changes in clothing and background, or geometric keypoint features that lack contextual information. The main focus of this thesis is the proposition that a combination of the two modalities can outperform each of them independently.

We introduce a complete pipeline starting from the pre-processing of untrimmed online videos of the 3DYoga90 dataset and extracting the 3D world-coordinate skeletons with the use of MediaPipe. Two novel backbones are fine-tuned on an extremely strict cross-subject split as the unimodal baselines – the first one is **Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN)** on the skeleton modality, and the other is Video Swin Transformer. We propose a logit-level cross-modal fusion model, termed *LogitPerceiver*, which merges the predictions without requiring full-fledged end-to-end training due to its high computational cost. In addition, the model projects the backbone's logits into a unified space and employs a trainable query to concentrate on two input streams as well as a gated adjustment to the skeleton prediction results.

The late-fusion framework enables adaptive and context-dependent fusion of information related to the input samples from both visual and geometric modalities. Experiments carried out using the 3DYoga90 dataset reveal that the skeleton-based CTR-GCN can achieve a balanced accuracy of 86.96% compared to 81.64% by the Video Swin Transformer. The proposed fusion framework is able to score 89.64% balanced accuracy, 91.06% overall accuracy, and 88.96% macro-averaged F1-score, beating the performance of all unimodal baselines and establishing a new baseline on the dataset. This thesis provides a reproducible multimodal framework, a light fusion network compatible with frozen experts, and a strong benchmark dataset for yoga position classification.

CONTENTS

CANDIDATE'S DECLARATION.....	II
ACKNOWLEDGEMENT.....	III
CERTIFICATE BY THE SUPERVISOR.....	IV
ABSTRACT.....	V
CONTENTS.....	VI
LIST OF FIGURES.....	X
LIST OF TABLES.....	XI
LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE.....	XII
CHAPTER 1 INTRODUCTION	1
1.1 MOTIVATION AND CONTEXT	1
1.2 CHALLENGES IN YOGA POSE RECOGNITION	1
1.3 MULTIMODAL FUSION FOR ROBUST RECOGNITION	2
1.4 PROPOSED APPROACH AND CONTRIBUTIONS	2
CHAPTER 2 LITERATURE REVIEW	3
2.1 BACKGROUND AND SCOPE	3
2.2 DATASETS FOR YOGA POSE ANALYSIS.....	3
2.2.1 Static 2D Datasets: Yoga-82 and Its Variants	3
2.2.2 3D and Video Datasets: 3DYoga90 and 3D-Yoga	4
2.3 SKELETON-BASED ACTION RECOGNITION: GCN	4
2.3.1 From CNNs/RNNs to Graph Convolutions	4
2.3.2 Adaptive and Channel-wise Topology Refinement	5
2.4 VIDEO-BASED ACTION RECOGNITION: VISION TRANSFORMERS	5
2.4.1 3D Convolutional Approaches and Their Limitations	5
2.4.2 Video Transformers: The Swin Paradigm.....	5
2.5 MULTIMODAL FUSION FOR ACTION RECOGNITION.....	6
2.5.1 The Complementary Nature of Appearance and Skeleton.....	6
2.5.2 Fusion Taxonomies: Early, Intermediate and Late Fusion.....	6
2.6 RESEARCH GAP.....	7
CHAPTER 3 PROPOSED METHODOLOGY	8
3.1 DATASET	8
3.1.1 Primary Research Dataset: 3DYoga90.....	8
3.1.1.1 Multi-Modal Spatio-Temporal Sequences	8

3.1.1.2 Expanded Hierarchical Taxonomy and Expert Curation.....	8
3.1.1.3 Dataset Statistics and Characteristics.....	9
3.1.2 Justification for Modality Transition	11
3.2 SPATIO-TEMPORAL SKELETON EXTRACTION	11
3.2.1 Keypoint Extraction and Representation	11
3.2.2 Processing Video Frames: Temporal Consistency and World Coordinates	11
3.2.3 Data Structure of Extracted Skeletons	12
3.3 VIDEO DATASET FOR ISOLATED ACTION RECOGNITION.....	13
3.3.1 The 3DYoga90 Dataset and Its Temporal Annotations.....	13
3.3.2 Frame-Accurate Temporal Clipping	13
3.4 DATA AUGMENTATION STRATEGY	14
3.4.1 Augmentation of Spatiotemporal Visual Representations	14
3.4.1.1 Horizontal Flipping	14
3.4.1.2 Random Spatial Translation	15
3.4.2 Augmentation of 3D Skeletal Sequences.....	16
3.4.2.1 Coordinate Noise Injection	16
3.5 THE DUAL-MODAL DATASET	17
3.5.1 Modality-Specific Preprocessing and Representations	17
3.5.2 Constructing a Paired Multimodal Instance	17
3.6 CLASS REBALANCING STRATEGIES	18
3.6.2 The Class Imbalance Problem in Supervised Learning.....	19
3.6.3 Loss-Based Rebalancing: Inverse-Frequency Class Weights.....	19
3.6.4 Sampling-Based Rebalancing: Weighted Random Sampling.....	19
3.7 TRAIN-TEST SPLITTING: THE CROSS-SUBJECT PROTOCOL.....	19
3.7.1 The Perils of Random Instance-Level Splitting.....	19
3.7.2 Source-Based Splitting as a Proxy for Subject Independence.....	20
3.7.3 Statistical Rationale and Validation	20
3.8 EVALUATION METRICS	21
3.8.1 Overall Accuracy and Its Shortcomings	21
3.8.2 Balanced Accuracy: Equalising Class Influence	22
3.8.3 Macro-Averaged F1 Score: Harmonising Precision and Recall	22
3.8.4 Combined Interpretation and Practical Use.....	23
3.9 CTR-GCN ARCHITECTURE: SKELETON-ONLY BASELINE	23
3.9.1 Overall Architecture and Data Adaptation.....	23
3.9.2 Role as a Baseline and Foundation for Multimodal Fusion	24
3.10 VIDEO SWIN TRANSFORMER: APPEARANCE-BASED BASELINE	25
3.10.1 Transfer Learning from Kinetics-400	25
3.10.2 Role as a Vision-Only Baseline and Future Fusion	26

3.11 MULTIMODAL FUSION VIA LOGIT-LEVEL CROSS-ATTENTION	26
3.11.1 Design Rationale: Frozen Experts and Late Fusion.....	27
3.11.2 The <i>LogitPerceiver</i> : Cross-Attention over Class Predictions	27
3.11.2.1 Tokenisation of Logits.....	27
3.11.2.2 Learnable Query and Cross-Attention.....	28
3.11.2.3 Feed-Forward Refinement.....	29
3.11.2.4 Residual Delta Prediction and Gating	30
3.11.3 Significance as the Final Model.....	32

CHAPTER 4 EXPERIMENTAL SETUP AND RESULTS 33

4.1 HARDWARE AND IMPLEMENTATION DETAILS.....	33
4.1.1 Software Environment and Library Stack	33
4.1.2 GPU Configuration and Mixed-Precision Training.....	34
4.2 COMMON TRAINING PROTOCOLS	34
4.2.1 Optimisation Algorithm: AdamW.....	34
4.2.2 Learning Rate Schedule: Cosine Annealing	34
4.2.3 Early Stopping and Model Selection	35
4.3 TRAINING PROTOCOL FOR SKELETON-ONLY CTR-GCN	35
4.3.1 Data Sampling Strategy	35
4.3.2 Summary of Hyperparameters	36
4.4 TRAINING PROTOCOL FOR VIDEO SWIN TRANSFORMER.....	36
4.4.1 Transfer Learning from Kinetics-400	36
4.4.2 Summary of Training Hyperparameters	37
4.5 TRAINING PROTOCOL FOR MULTIMODAL FUSION MODEL	37
4.5.1 Meta-Learning over Frozen Backbones.....	38
4.5.2 Learning Rate Schedule with Warm-Up	38
4.5.3 Summary of Hyperparameters	38
4.6 EVALUATION AND RESULTS	39
4.6.1 Unimodal Baseline Performance	39
4.6.1.2 CTR-GCN (Skeleton-Only)	39
4.6.1.3 Video Swin Transformer (Vision-Only).....	42
4.6.1.4 Comparison of Unimodal Baselines.....	44
4.6.2 Multimodal Fusion Performance	44
4.7 DISCUSSIONS ON THE RESULTS	47
4.7.1 Analysis of Fusion Gain.....	47
4.7.2 Convergence and Generalisation	47
4.7.3 Effectiveness of Cross-Modal Fusion.....	48

CHAPTER 5 CONCLUSION 49

5.1 APPLICATIONS AND SOCIETAL IMPACT.....	49
5.2 FUTURE SCOPE	49



5.3 CONCLUDING REMARKS.....	50
CHAPTER 6 REFERENCES	52

LIST OF FIGURES

Figure 3.1: 3DYoga90 Dataset Hierarchy [4]	8
Figure 3.2: Bar chart showing the top 10 Yoga Poses.....	10
Figure 3.3: Temporal Skeleton Extraction	11
Figure 3.4: Tensor Data Structure of extracted skeletons	12
Figure 3.5: Clipping the video files	14
Figure 3.6: Horizontal Flipping of Videos	15
Figure 3.7: Random Spatial Translation of videos.....	15
Figure 3.8: Dual Multi-modal Dataset formation	18
Figure 3.9: Log-Scaled Long-Tail Distribution of Yoga Poses showing significant class imbalance	18
Figure 3.10: Sankey Diagram showing Cross Subject Dataset Split	21
Figure 3.11: CTR-GCN: Overall Architecture	24
Figure 3.12: Video Swin Transformer: Updated classification head to support Yoga Pose Classification.....	25
Figure 3.13: LogitPerceiver: Tokenisation of Logits	28
Figure 3.14: LogitPerceiver: Learnable Query and Multi-head Cross-Attention	29
Figure 3.15: LogitPerceiver: Feed-Forward Refinement	30
Figure 3.16: LogitPerceiver: Building the hidden representation from Residual	30
Figure 3.17: LogitPerceiver: Delta Projection	31
Figure 3.18: LogitPerceiver: Gating and Fused Logits.....	31
Figure 4.1: CTR-GCN Validation Accuracy curve over epochs	40
Figure 4.2: CTR-GCN Balanced Accuracy curve over epochs.....	40
Figure 4.3: CTR-GCN F1 score curve over epochs.....	41
Figure 4.4: CTR-GCN Loss curve over epochs	41
Figure 4.5: Video Swin Transformer Validation Accuracy curve over epochs	42
Figure 4.6: Video Swin Transformer Balanced Accuracy curve over epochs.....	43
Figure 4.7: Video Swin Transformer F1 score curve over epochs	43
Figure 4.8: Video Swin Transformer Loss curve over epochs	44
Figure 4.9: Cross-attention fusion model Validation Accuracy curve over epochs	45
Figure 4.10: Cross-attention fusion model Balanced Accuracy curve over epochs.....	45
Figure 4.11: Cross-attention fusion model Validation F1 Score curve over epochs	46
Figure 4.12: Cross-attention fusion model Loss curve over epochs	46

LIST OF TABLES

Table 3.1: Dataset Statistics of 3DYoga90 9

Table 4.1: Training Hyperparameters for CTR-GCN.....36

Table 4.2: Training Hyperparameters for Video Swin Transformer.....37

Table 4.3: Training Hyperparameters for Logit-Level Cross-Attention Fusion.....38

Table 4.4: Comparison of Unimodal Baseline Performances 39

Table 4.5: Comparing Multimodal fusion with the baseline models 48



LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

T	Total number of frames or temporal dimension
C	Number of color channels (e.g., $C = 3$ for RGB) or the number of classes
H, W	Spatial dimensions of an image tensor (Height and Width)
V	Number of MediaPipe joints/landmarks (Vertices)
X, Y, Z	Spatial coordinates in meters (world coordinates)
x, y, z	Normalized spatial/depth coordinates in the image plane
TP_c	True Positives for a specific class c
FN_c	False Negatives for a specific class c
FP_c	False Positives for a specific class c
η	Learning rate
l_{vis}	Logit vector produced by the vision backbone
l_{geo}	Logit vector produced by the geometry (skeleton) backbone
l_{fused}	Fused logit vector
Q	Learnable query vector used in the cross-attention module
δ	Delta logit vector for residual correction applied to the skeleton logits
α	A learnable scalar gate used to control the residual connection
AGCN	Adaptive Graph Convolutional Network
AMP	Automatic Mixed Precision
CNN	Convolutional Neural Network
CTR-GCN	Channel-wise Topology Refinement Graph Convolutional Network
CUDA	Compute Unified Device Architecture
FFN	Feed-Forward Network
GCN	Graph Convolutional Neural Network
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
NaN	Not a Number
RGB-D	Red, Green, Blue – Depth
ST-GCN	Spatial-Temporal Graph Convolutional Network
ViT	Vision Transformer



CHAPTER 1

INTRODUCTION

1.1 MOTIVATION AND CONTEXT

Yoga has become extremely popular in recent years, and its popularity has not gone unnoticed for various reasons, including the physical benefits it brings, relaxation, and even mental improvement [1], [2]. While many people practice yoga exercises in their homes today, they lack the necessary guidance and assistance to do it properly. Hence, technological intervention is needed to ensure that systems are developed capable of detecting the user's errors and providing tailored instructions and recommendations, regardless of whether a personal instructor will be present [3]. Automatic detection of yoga postures on video can prove invaluable for a variety of applications, from health-related to gaming.

Considering the complexity involved when distinguishing yoga postures from the computer vision perspective, the task becomes quite challenging. The reason for this is that unlike dynamic activities such as running or jumping, yoga poses remain stationary; hence the difference lies in very subtle details such as joints angle or limbs positions [4]. Even the similarity among some poses, such as Warrior One and Warrior Two poses makes the classification difficult to perform. Furthermore, each individual varies physically, clothing style and camera angles can also contribute to the distortion of images making them look differently. To enable recognition of up to 90 yoga poses in a database such as the 3DYoga90 database, the proposed models must possess certain characteristics, including being able to recognize subtle details while ignoring non-relevant factors like lighting and attire.

1.2 CHALLENGES IN YOGA POSE RECOGNITION

Recognizing yoga poses automatically has been found to be difficult due to several reasons. The first challenge is related to the appearance variations, in which case the same pose may have different appearance characteristics based on factors such as attire, camera viewpoint, background, and lighting conditions. For instance, a machine learning algorithm trained on yoga videos shot in a well-lighted studio may perform poorly when employed for home use as a result of the varying environmental background in such a scenario [5]. Second is the existence of poses that are very close to each other whereby the distinguishing characteristic may only be the posture of certain body joints and hands or feet positions. In order to distinguish such poses, it is important to design an algorithm that captures the fine-grained appearance features [6]. Thirdly, the algorithm is supposed to detect poses regardless of the pose executor, meaning that it cannot base recognition on personal attributes such as facial characteristics, physical traits, and gait analysis. Evaluating such algorithms can prove to be difficult as they have to be generalized to work across multiple people [7].

1.3 MULTIMODAL FUSION FOR ROBUST RECOGNITION

Multimodal information fusion, i.e., an integration of different types of data such as visual and audio streams, is critical for the comprehension process in video analysis tasks. As previously mentioned, the multimodal information fusion can be performed by three techniques: early fusion of raw inputs, mid-level feature concatenation [8], and late fusion of the output of each modality classifier. Early and mid-level approaches require joint training of each expert that is prohibitively expensive for large pre-trained models. The late approach allows each individual modality model to be trained independently and then fused using the meta-learner [9], [10]. Therefore, it does not introduce any overhead in terms of computational complexity. However, simply averaging or weighing individual scores from modalities does not give any advantage since it cannot capture any intermodal dependencies between inputs. Attention-based approaches have proved their superiority in many computer vision and NLP problems [11]. It seems reasonable to assume that the attention-based mechanism can significantly improve multimodal fusion and lead to more accurate multimodal classification of a sample.

1.4 PROPOSED APPROACH AND CONTRIBUTIONS

This thesis presents a novel multi-modal approach for yoga pose recognition. Our framework integrates the visual appearance as well as the geometry of a yoga pose through a learnable cross-attention mechanism. The purpose of this thesis is to integrate multimodal data into our pose recognition problem to make our framework perform better. Specifically, we use a cross-attention head mechanism that allows us to pay attention to the most important aspects of a pose. Our model consists of three major components, and we will explain each component in detail later.

- Data preprocessing and skeleton extraction
- Unimodal baselines
- Logit-level cross-attention fusion (*LogitPerceiver*)

Key contributions made by this thesis may be outlined as follows:

- A complete end-to-end pipeline for multimodal yoga pose recognition from raw videos to classification, specific to 3DYoga90.
- The development of highly competitive unimodal baselines based on a stringent cross-subject procedure for benchmarking future methods.
- The introduction of a novel, efficient logit-level fusion network capable of combining pre-trained experts without joint fine-tuning.
- Experimental results proving that the proposed multimodal fusion approach provides 89.64% of balanced accuracy, which exceeds that of unimodal streams and is state-of-the-art in terms of 3DYoga90.



CHAPTER 2

LITERATURE REVIEW

2.1 BACKGROUND AND SCOPE

Action recognition is an approach towards automatically recognizing different activities performed by humans using visual information. Applications of action recognition include but are not limited to surveillance systems, human-computer interactions, medicine and sports [12]. Fine-grained action classification involves recognizing actions that have similar body configurations in gross but slightly varied joint alignments. The problem of fine-grained action recognition is further complicated in applications like yoga pose recognition where it is necessary for a model to not only learn global postural semantics but also the finer kinematics involved.

2.2 DATASETS FOR YOGA POSE ANALYSIS

Diverse, well-documented datasets form an essential pre-requisite for making advancements in pose classification using deep-learning techniques [4], [13]. Initial research works were based on static two-dimensional sets of poses of yoga postures, while later developments included three-dimensional spatiotemporal information.

2.2.1 Static 2D Datasets: Yoga-82 and Its Variants

The most prevalent among static yoga datasets is Yoga-82 dataset developed by Verma et al. [13]. This consists of more than 28,400 Internet-sourced images arranged into 82 different pose categories according to a three-level hierarchy – there are six main root classes (such as standing, sitting, inverted, reclining) followed by intermediary categories that split into specific pose types. The hierarchical structure acts as a remedy for the lack of detailed joint labeling by allowing a coarse-to-fine classification process. Yoga-82 became a standard dataset for testing CNN yoga classifiers, which showed an increase in accuracy when using VGG-16, ResNet-50, and DenseNet-121 models [3], [14].

Further developments include Bera et al.'s work on the SYD (Sports, Yoga, Dance) benchmark with Yoga-82 along with a larger Yoga-107 dataset, sports, and dance postures, and SYD-Net, a CNN-based model with patch-level attention [14]. Recent research has investigated various other approaches to make yoga poses recognition computationally efficient with CNN structures like YPose and a grading system based on skeletons [3], [15].

Although useful, static 2D benchmarks possess inherent disadvantages as they lack the temporal dimension, contain no depth information, and suffer from viewpoint variability along with background distractions that cause shifts in domain distribution [4], [16]. This limitation drove the trend toward using video data and skeleton-based 3D representations.

2.2.2 3D and Video Datasets: 3DYoga90 and 3D-Yoga

Given the importance of temporal and spatial data, several video-based yoga datasets have been created. 3DYoga90 [4] increases the number of poses to 90 and offers, per pose instance, a well-cut RGB video clip as well as its skeleton sequence obtained from the world coordinate space with the help of the BlazePose approach [17]. In its construction, the dataset used a group of six people including yoga professionals and videos from YouTube obtained through multilingual search (English, Korean, Japanese, Sanskrit). It consists of 6 177 RGB video clips (of which 5 526 skeleton sequences were successfully generated) with an average length of ~12 seconds and the data split based on original video sources. The latter eliminates the bias towards individual instances, which can be problematic when using random splits at the level of individual data samples [5], [7].

In addition, 3D-Yoga by Zhu et al. presents a 3D yoga pose database with 3,792 actions and 16,668 key frames from RGB-D data of 22 people captured by two RGB-D cameras, featuring 117 pose categories. This database includes 3D skeletons reconstructed from multiple views and was evaluated using a two-stream adaptive graph convolutional neural network.

This thesis takes 3DYoga90 as the main empirical basis as this dataset is one of a kind, with its combination of raw RGB video and 3D skeletal landmarks [4] for a systematic investigation into the fusion of multimodal approaches.

2.3 SKELETON-BASED ACTION RECOGNITION: GCN

2.3.1 From CNNs/RNNs to Graph Convolutions

Traditional skeleton-based action recognition approaches have used joint locations as sequential or pseudo-image data and applied RNNs or 1D/2D CNNs [15], [18]. But these approaches do not take into account the underlying graph structure in the human skeleton, in which joints are connected by bones that have different physical and functional significance. Spatial-Temporal Graph Convolutional Networks (ST-GCN), proposed by Yan et al. [18], model the human body's graph structure inside each frame and apply convolutions along the temporal axis, leading to outstanding performance on action recognition datasets like NTU RGB+D [7]. The major contribution of ST-GCN is the introduction of a fixed, manually constructed adjacency matrix based on the connectivity of the skeletal system; this ensured that the network takes the anatomical constraints into account. But this fixed connectivity structure cannot represent the connectivity requirements related to actions and channels: the required connectivity between joints for detecting whether a person is walking or running can be different than for a yoga balance pose [6].

2.3.2 Adaptive and Channel-wise Topology Refinement

In contrast, the following approaches were able to overcome this limitation by proposing learnable adjacency matrices. 2s-AGCN (Shi et al. [19]) proposed a two-stream design that simultaneously learns both the graph structure and the node features, thus dynamically establishing additional relationships within the network beyond the skeletal graph. Although more flexible than ST-GCN, 2s-AGCN also uses one common adjacency matrix across all channels.

To further improve the flexibility, Chen et al. [6] introduced the Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN). By assigning a unique adjacency matrix to each output channel, they managed to increase the discriminative power of the network. Specifically, their approach involves calculating a dynamic similarity matrix between input features by using a cross-channel attention module. This allows adding it into the previously fixed skeletal adjacency matrix to obtain the final model (blocks of spatial CTR-GC interleaved with temporal convolutions), which is capable of modeling action- and channel-specific relationships in joint movements. CTR-GCN achieves better results on standard benchmarks such as NTU RGB+D and Kinetics-Skeleton [6]. Therefore, this thesis will use the CTR-GCN as a backbone of the geometry stream for 33-joint MediaPipe topology obtained from the 3DYoga90 dataset [4], [17].

2.4 VIDEO-BASED ACTION RECOGNITION: VISION TRANSFORMERS

2.4.1 3D Convolutional Approaches and Their Limitations

Prior to the emergence of transformers, video action recognition had relied mostly on 3D convolution networks like C3D [20], I3D [21], and SlowFast [22]. These models have proven to be very successful, but one problem with these 3D CNNs is the presence of localization issues, as the receptive field size increases linearly with increasing network depths [23]. Although two stream approaches where there is a distinct pathway for the spatial and temporal streams [16] can somewhat alleviate this issue, localization still persists in temporal processing.

2.4.2 Video Transformers: The Swin Paradigm

Unlike CNNs that employ convolutions, Vision Transformers (ViTs) use self-attention allowing every token to attend to all the other tokens within the same sequence [24]. Unfortunately, ViTs suffer from quadratic complexity caused by the need to perform global attention. To overcome this limitation, the Swin Transformer [25] proposed using a hierarchical design along with shifted window attention where self-attention is performed within non-overlapping spatial windows and subsequently shifted to facilitate cross-window communication, making it linear in the number of tokens.

Similarly, the Video Swin Transformer [23] extended this paradigm to modeling spatiotemporal data by creating 3D patches from the input clip and then using 3D shifted window attention. Such an approach allows modeling both short-range movement and

long-range dependencies in video while remaining linear. Pre-trained on large-scale datasets, such as Kinetics-400 [26], Video Swin achieved impressive results surpassing state-of-the-art on multiple video benchmarks [23]. Since most yoga postures require modeling the information over a 30-frame window (about 12 seconds), being able to model long-range dependencies becomes critical. In the current thesis, we use Video Swin Transformer (Tiny version) pre-trained on Kinetics-400 as our baseline and frozen expert in multimodal fusion.

2.5 MULTIMODAL FUSION FOR ACTION RECOGNITION

2.5.1 The Complementary Nature of Appearance and Skeleton

No single modality provides complete information for action recognition in unconstrained environments. RGB video captures rich appearance cues (clothing, background, texture) and motion boundaries, but is susceptible to domain shift caused by changes in illumination, viewpoint, and background clutter [16], [21]. Conversely, 3D skeletal data are invariant to such superficial variations and encode the pure kinematic configuration, yet they discard contextual information and depend on the accuracy of the upstream pose estimator [18], [27]. These properties are complementary: a robust system can leverage skeletal geometry as a stable backbone and appearance cues to resolve ambiguities when the skeleton is occluded or misestimated [8], [28].

2.5.2 Fusion Taxonomies: Early, Intermediate and Late Fusion

Multimodal fusion methods can be roughly grouped based on their position within the representation hierarchy [28]. Early fusion is a method that fuses raw inputs prior to modality-specific processing; however, it may not necessarily learn domain-specific representations. Feature-level fusion, also known as intermediate fusion, extracts feature-specific representations for each individual modality first and then concatenates or sums up or performs attention on them [8]. Decision-level fusion combines independent unimodal decisions (logits or probabilities) using average, weighted voting, or a meta-learner [9], [29]. The benefits of decision-level fusion include decoupling the training of each expert and the ability to train frozen pretrained models.

There have been various attempts at attention-based fusion. CMF-Transformer is a cross-modal transformer-based fusion model to fuse spatiotemporal video features and skeleton joint features. TransMODAL is an attention-based fusion mechanism to facilitate iterative cross-modal information sharing between the RGB and pose stream. MAF-Net uses both self-attention and skeletal attention modules under the decision-level fusion framework.

2.6 RESEARCH GAP

Literature review demonstrates the following development trends:

- **Data representation:** The community evolved from 2D static image databases such as Yoga-82 [13] and SYD [14] to 3D video datasets like 3DYoga90 [4] and 3D-Yoga that provide the appearance and skeleton streams respectively.
- **Geometric recognition:** Graph convolution networks progressed from static architecture (ST-GCN [18]) to dynamic graph-based models (2s-AGCN [19], CTR-GCN [6]) with solid geometric priors.
- **Video recognition:** Recent advancements in vision transformer architecture such as Video Swin [23] outperformed CNN architectures on various benchmarks by learning distant spatial-temporal relations.
- **Multimodal processing:** Several works have proposed multimodal solutions based on attention mechanism, which combine RGB and skeleton modality, however, most of them require joint training or fine-tuning of backbones.

However, there is still an important research question to be answered: how can we efficiently integrate well-trained skeletal and video models for fine-grained yoga pose classification without requiring costly end-to-end training again? Fortunately, this problem is precisely what the current thesis tries to solve with a proposed cross-attention strategy at the logit level using a frozen CTR-GCN and Video Swin backbone coupled with a trainable residual correction head. Not only does this method utilize the synergies of both geometry and appearance, but it also follows a subject-independent evaluation by employing a cross-subject split setting, making it state-of-the-art.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 DATASET

3.1.1 Primary Research Dataset: 3DYoga90

3.1.1.1 Multi-Modal Spatio-Temporal Sequences

In order to address the deficiencies associated with static 2D images, the methodology used in this thesis has been modified such that it can utilize the 3DYoga90 dataset [4]. 3DYoga90 is an important contribution to datasets involving human action because it provides multi-modal sets of continuous RGB videos together with their respective 3D skeletons. Since there is temporal information contained within the video, there is an opportunity to model the transitions between different postures. Consequently, there will be the possibility of not only capturing the spatial arrangement of the joints but also capturing kinematic features like joint velocities and accelerations.

3.1.1.2 Expanded Hierarchical Taxonomy and Expert Curation

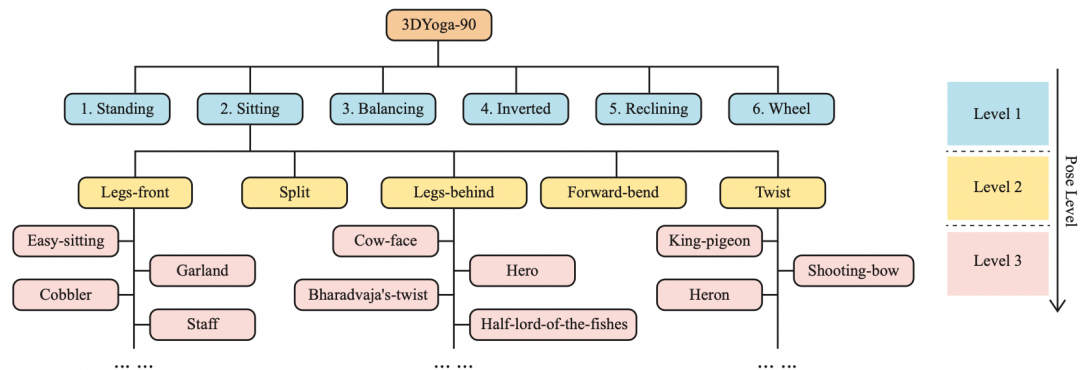


Figure 3.1: 3DYoga90 Dataset Hierarchy [4]

Building on the structure taxonomy created by Yoga-82, 3DYoga90 adopts a highly detailed hierarchical labeling scheme involving three levels of labels, while increasing the number of categories from 82 to 90 [4]. This is due to the greater variety of complex poses, which pose additional challenges to advanced spatio-temporal recognition models. Importantly, unlike web-sourced databases, 3DYoga90 was carefully compiled by professionals in the field, including yoga teachers. This ensures that there are no anatomical inaccuracies in the execution of poses since such a database is needed for scientific purposes. Figure 3.1 shows the dataset hierarchy for the 3DYoga90 dataset.

3.1.1.3 Dataset Statistics and Characteristics

Overall, the total number of curated RGB video samples in 3DYoga90 dataset amounts to 6,177 samples. The base videos have been extracted from 2,170 different channels/creators, thus guaranteeing an extensive variation of environments and users. Table 3.1: describes the statistics of different classes within the dataset.

Table 3.1: Dataset Statistics of 3DYoga90

Yoga Pose	Count	Yoga Pose	Count	Yoga Pose	Count
downward-dog	245	half-moon	48	dolphin	43
standing-forward-bend	226	garland	48	warrior-2	43
half-way-lift	212	crane-crow	48	standing-split	43
mountain	163	extended-puppy	48	cow-face	43
chair	140	side-plank	48	half-lord-of-the-fishes	43
cobra	128	upward-facing-two-foot-staff	48	flying-splits	43
cockerel	120	gate	47	shoulder-stand	43
extended-triangle	107	warrior-3	47	plow	43
extended-side-angle	101	child	47	peacock	43
corpse	95	lizard	47	upward-plank	43
staff	94	side-reclining-leg-lift	47	pigeon	43
wind-relieving	94	wild-things	47	high-lunge	42
fish	93	cat	47	bharadvaja's-twist	42
happy-baby	93	lord-of-dance	46	eight-angle	42
shoulder-pressing	92	shooting-bow	46	handstand	42
reclining-cobbler	91	king-pigeon	46	feather-peacock	42
reclining-hero	82	upward-bow	46	standing-big-toe-hold	41
frog	80	cow	46	heron	41
tree	69	bow	46	side-crane-crow	41
intense-side-stretch	59	eagle	45	bridge	41
scale	59	hero	45	three-legged-dog	40
scorpion	57	firefly	45	tortoise-pose	40
reclining-hand-to-big-toe	57	supine-twist	45	dolphin-plank	40

locust	55	camel	45	boat	40
revolved-head-to-knee	54	warrior-1	44	easy-sitting	39
monkey	53	low-lunge	44	headstand	39
low-plank	51	seated-forward-bend	44	noose	38
reverse-warrior	49	balancing-table	44	wide-legged-forward-bend	36
cobbler	49	plank	44	downward-facing-frog	36
head-to-knee	49	goddess	43	wide-angle-seated-forward-bend	33

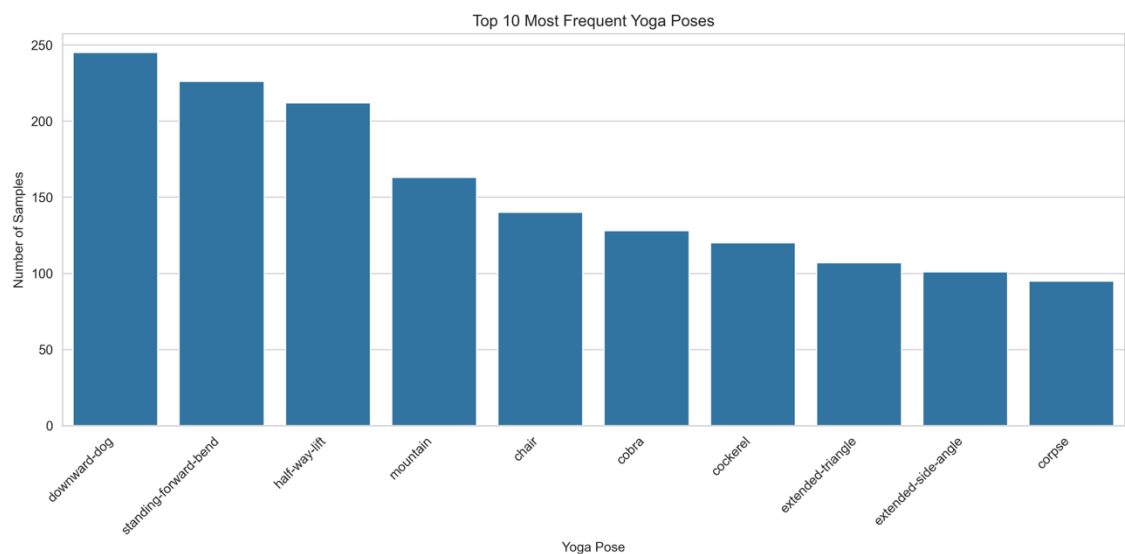


Figure 3.2: Bar chart showing the top 10 Yoga Poses

Figure 3.2 shows the top 10 yoga poses in the dataset.

- **Skeleton Dataset Attrition:** During the stage of skeleton extraction, the cases in which BlazePose did not detect poses in some of the frames led to the attrition of the dataset, and thus the final skeleton dataset consists of 5,526 skeleton data.
- **Distribution and Sequence Length:** In terms of distribution and sequence length at level 3 classification, the pose 'Downward Dog' occurs frequently, with 245 video occurrences. Throughout the dataset, the average number of occurrences per pose is 75. The average length of the extracted sequence is 12.16 seconds, having a minimum of 2.83 seconds (for pose 'Halfway Lift') and maximum of 25.84 seconds (for pose 'Balancing Table').

3.1.2 Justification for Modality Transition

The evolution from the image-based Yoga-82 dataset to the video-based 3DYoga90 dataset illustrates a paradigm shift from the extraction of static features to the analysis of the entire spatio-temporal sequence. The nature of the 3D skeleton sequences in 3DYoga90 allows us to obtain invariant topological representation of the human body, which automatically avoids the issues of viewpoint dependence and background noise encountered in 2D datasets. Through the integration of temporal information by means of video sequence, the current approach can accurately model the transitional phase of motions, providing us with a high-quality representation of human kinematics.

3.2 SPATIO-TEMPORAL SKELETON EXTRACTION

3.2.1 Keypoint Extraction and Representation

BlazePose model, a key component of the Mediapipe pipeline [17], is used to detect keypoints in body poses. BlazePose model is a fast and extremely accurate full body pose detection solution suitable for real-time use. It detects 33 keypoints, each with four properties: normalized (x, y) coordinates in the image plane, relative depth coordinate z , and a confidence score that tells the visibility of the particular point. This representation gives a concise illustration of the skeletal structure without the need to work with complete pictures.

The extracted landmarks provide a structured representation which holds a considerable amount of data about the physical structure of the body. The (x, y) location denotes the relative placement, whereas the depth aspect signifies the motion of the body parts in the direction of the optic axis of the camera. This aids the distinction between yoga poses that may seem similar but differ with respect to front/backward stretch. The visibility score helps the system cope with invisible/ambiguous keypoints [30], [31].

3.2.2 Processing Video Frames: Temporal Consistency and World Coordinates

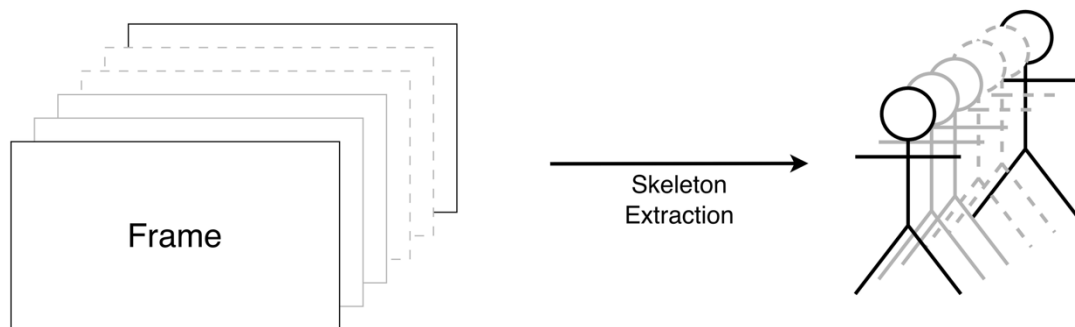


Figure 3.3: Temporal Skeleton Extraction

For each frame at index t , the raw network output comprises:

- **Image landmarks:** $(x_{i,t}, y_{i,t})$ in normalized image coordinates (range $[0, 1]$).
- **World landmarks:** $(X_{i,t}, Y_{i,t}, Z_{i,t})$ in meters, where $i \in \{0, \dots, 32\}$

As the process operates on every frame independently as shown in Figure 3.3 (with an exception of the ROI linking function), there is no built-in temporal filtering or smoothing done automatically. The design of the algorithm is aimed at preserving the fast-changing features inherent to the moving object. This feature is important for gesture recognition and motion capture.

3.2.3 Data Structure of Extracted Skeletons

In the case of video data represented by T frames, the landmarks in the world coordinate system are organized in the form of a three-dimensional array $\mathcal{S} \in \mathbb{R}^{3 \times 33 \times T}$. The axes are defined as follows:

- **Dimension 0 (coordinate):** $0 \rightarrow X, 1 \rightarrow Y, 2 \rightarrow Z$.
- **Dimension 1 (landmark index):** An integer i from 0 to 32, corresponding to the anatomical landmark.
- **Dimension 2 (time):** Frame index t ranging from 0 to $T - 1$.

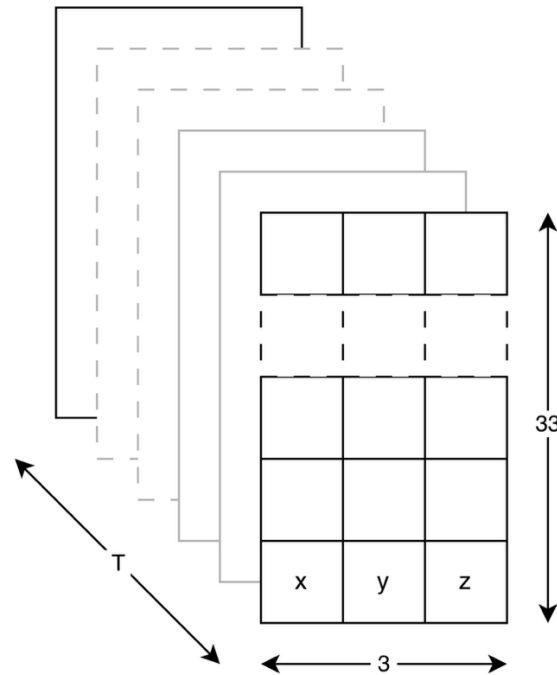


Figure 3.4: Tensor Data Structure of extracted skeletons

Each element $\mathcal{S}_{c,i,t}$ thus stores the c -th coordinate of landmark i at frame t as shown in Figure 3.4. When a landmark is not detected in a given frame (e.g., due to occlusion or low confidence), the corresponding values are set to *NaN*. The array is pre-allocated with NaN entries of the maximal possible length and then trimmed to $T_{\text{actual}} \leq T$ after processing all frames, removing any trailing frames that were never filled because the video ended earlier than estimated or contained fewer readable frames.

So, for each element $\mathcal{S}_{c,i,t}$, the value corresponds to the coordinate c of the landmark i at time frame t . For those landmarks that cannot be detected during certain time frames (for example, due to occlusion or uncertainty), their respective values will be assigned NaN values. The whole array is allocated in advance with the maximum number of NaN elements and then shortened to $T_{\text{actual}} \leq T$ depending on the number of frames processed.

This tensor representation of data allows fast vector processing for further analysis, such as calculating the velocity of joint movement through finite differences on the time axis or isolating a trajectory of a chosen landmark throughout the whole clip. The use of a world coordinate system measured in meters guarantees an unambiguous interpretation of numeric results – for instance, moving +1.0 along the Y-axis means changing altitude by one meter, irrespective of the zoom level of the camera.

3.3 VIDEO DATASET FOR ISOLATED ACTION RECOGNITION

3.3.1 The 3DYoga90 Dataset and Its Temporal Annotations

The 3DYoga90 database [4] offers an extensively annotated set of yoga postures, featuring hundreds of action instances extracted from publicly available videos on the web. In addition to a unique identifier and a URL pointing to the source video, the database offers temporal limits of these action instances in seconds. The limits represent the exact period when the specific yoga posture is executed. Unlike in frame-based annotation, when actions depend on the particular resolution and framerate of the video, timestamp-based annotations make sure that the definition of the action instance is independent of the encoding and frame rate of a particular video. This approach aligns well with benchmark practice [32], [33], as temporal annotations are provided in seconds.

3.3.2 Frame-Accurate Temporal Clipping

Once the right source file has been identified, the start time and duration are used to perform a video clip procedure. The end goal is to generate a new video file starting exactly at the given start time and with the given duration, maintaining the original frame rate and image quality. Fast-forwarding to specific points in time to obtain precise clips is not trivial in the case of compressed video streams, since most codecs use some form of inter-frame prediction, whereby seeking directly to a desired point in time may yield a P- or B-frame that relies on previous keyframes for its decoding. In order to get around this problem, the process entails fast-seeking to the closest keyframe before the intended timestamp, decoding from there, and ignoring all of the frames prior to the precise start

time. The clipped segment is encoded in a standardised format (H.264/MP4), with constant frame rate, as this is necessary for temporal determinism when processing the stream frame by frame [34]. Figure 3.5 shows clipping of the video files.

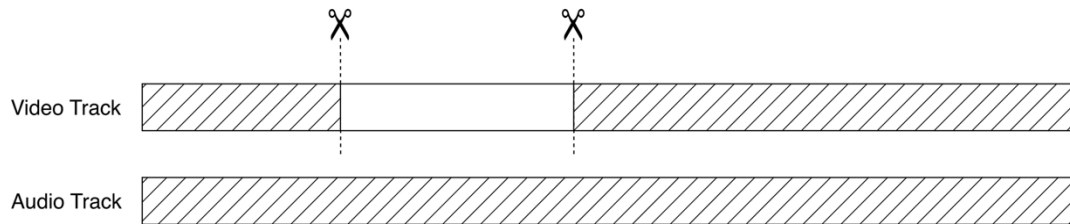


Figure 3.5: Clipping the video files

Opting for an encoder preset that promises speed and efficiency is a conscious decision to sacrifice efficiency for throughput because the target is to have the best possible clip produced in the shortest time possible since the video files are intermediate products.

3.4 DATA AUGMENTATION STRATEGY

The effectiveness of deep neural network models for action recognition is dependent on the variability and quality of the data used for training. In cases where video datasets labeled with relevant annotations are costly and cumbersome to acquire – for example, highly specific physical activities like yoga poses – data augmentation becomes an essential approach for mitigating overfitting and increasing generalization performance [35]. The augmentation framework employed in this thesis is designed for the bi-modal structure of the 3DYoga90 dataset [4] consisting of video tensors based on visual appearances as well as geometric representations of skeletons. Data augmentation procedures are carried out independently for each type of data depending on its properties, but they follow a consistent label-preserving philosophy that does not affect the semantics of the actions.

3.4.1 Augmentation of Spatiotemporal Visual Representations

The pathway for vision in this architecture consumes pre-processed video frames converted to fixed-size tensors with dimensions $T \times C \times H \times W$, where T is the number of frames, C the number of color channels, and H and W the spatial dimensions. For this input modality, there are two transformations that are performed with probability $p = 0.5$.

3.4.1.1 Horizontal Flipping

Horizontal frame flipping is probably the most commonly used photo-metric augmentation for action recognition since it takes advantage of the natural symmetry between the left and the right side of a human body while keeping the category of the performed action unaltered [36]. Yoga actions are typically defined based on the

configuration of arms, legs, and body regardless of its orientation, such that a video horizontally flipped corresponds to another instance of the same action class. For example, the right planking action becomes the left planking action but it remains an instance of the same pose class as shown in Figure 3.6. This augmentation helps in increasing the number of possible orientations in the training process which results in learning feature representations that do not depend on the orientation of the actor. This augmentation strategy is known to help improving accuracy in two-stream CNNs [16] and temporal segment networks [37], as well as in recent transformers for video analysis. The same transformation is applied to all frames of a given action sequence.

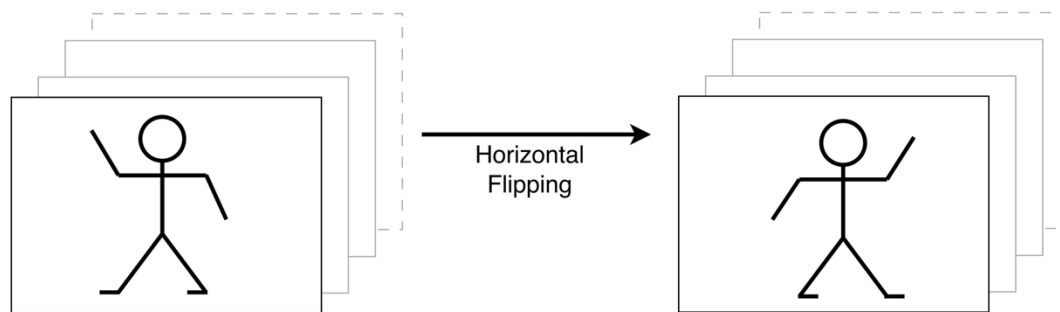


Figure 3.6: Horizontal Flipping of Videos

3.4.1.2 Random Spatial Translation

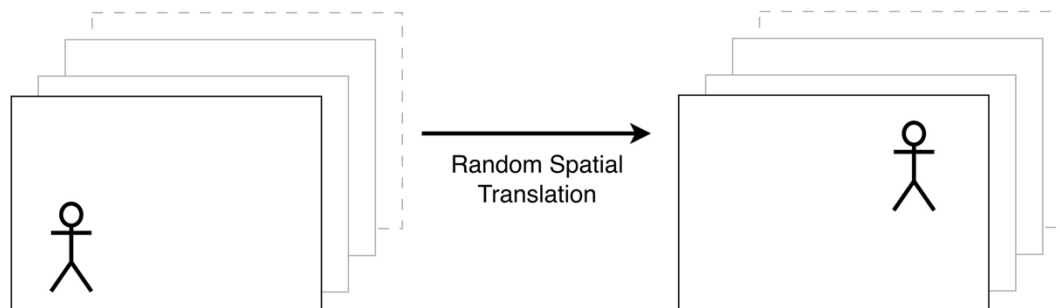


Figure 3.7: Random Spatial Translation of videos

The second type of affine transformation randomly translates the input by up to ± 16 pixels horizontally and vertically as shown in Figure 3.7. This technique simulates the spatial jitter that occurs due to the inaccurate detection of the object, small camera movements, or even small changes in the location of the subject in the scene. In the practical application of yoga exercise, it cannot always be expected that the subject will occupy the central position in the image. A robust detector should be tolerant of such shifts. Such a transformation technique is inspired by research conducted by Krizhevsky

et al. [38] in the area of image classification and its applicability in the case of videos has been proved for architectures like C3D [20] and I3D [21]. The displacement ensures that the body key points stay within the image boundary.

Using both of these operations separately results in an augmentation space which is created combinatorially. This means that a single training example can be horizontally flipped alone, translated alone, both of these, or neither, producing a four-fold increase in the augmented training data set per epoch. The use of randomization in augmentation is similar to that used by RandAugment [39], where no fine-tuning is required.

3.4.2 Augmentation of 3D Skeletal Sequences

The geometry-based pipeline operates on normalized 3D skeletal data, which has dimensions of $C \times T \times V$, where $C = 3$ (world coordinates in X, Y, Z directions) and $V = 33$ (MediaPipe joints). Here, we encounter an entirely different type of data augmentation, as this time the coordinates are continuous and refer to the actual world rather than being pixels from the screen. We use only one type of transformation – adding isotropic Gaussian noise with a standard deviation of $\sigma = 0.01$ meters.

3.4.2.1 Coordinate Noise Injection

Noise injection is a proven strategy for making skeleton-based action recognition networks more robust [6], [18]. Introducing small quantities of random noise into normalized joint coordinates prevents the network from learning exact numeric relationships and forces it to learn the actual structure of the kinematics involved. When it comes to 3D pose estimation, there is intrinsic noise in the normalized coordinates due to the inaccuracies of monocular 3D lifting [27]; therefore, injecting additional noise during training can be seen as a way of simulating the inherent variability at test time. This type of data augmentation is akin to input-level dropout and ensures that $\sigma = 0.01$ m represents about 1 cm of jitter in the world coordinate system, which is realistic considering that current state-of-the-art 3D pose estimation methods are accurate within the same range [40]. Our decision to use isotropic Gaussian noise, as opposed to joint- or axis-wise schedules, is a relatively modest regularizer since it takes into account the equal scaling factor of the metric space. Recent studies involving ST-GCN [15] and its derivatives have shown that joint-coordinate noise significantly boosts cross-subject and cross-view generalization performance.

Since the skeleton coordinates are pre-normalized by subtracting the hip center point, the noise addition does not result in any translational movements but rather mimics sensor inaccuracies. Such behavior is especially beneficial in the case of yoga, since fine changes in limb positions are used to differentiate between different postures, while the network needs to focus only on limb relative positions.

3.5 THE DUAL-MODAL DATASET

3.5.1 Modality-Specific Preprocessing and Representations

The joint data comprises the two modality repositories built separately using independent pipelines that were explained previously.

- **Appearance modality (Video Tensors):** Each trimmed action clip has been transformed into a fixed-size tensor of shape $T \times C \times H \times W$, where $T = 30$ frames, $C = 3$ RGB channels, and $H = W = 224$ spatial dimensions. In this way, the temporal resizing allows for having the same length input tensor irrespective of the initial duration of the action instance, whereas the spatial resizing aligns the tensor dimensions to the Swin3D-T video backbone [23].
- **Geometry modality (3D Skeletons):** For the same action instance, the landmark extraction process produces a variable-length sequence of 33 world-coordinate landmarks. This sequence is temporally resampled to $T = 30$ frames using a linear or edge-padding strategy, yielding a tensor of shape $3 \times 30 \times 33$ (coordinates $\times$ frames $\times$ landmarks). The coordinates are subsequently normalised to a hip-centred reference frame, removing absolute translation and leaving only the intrinsic pose configuration.

The modality-specific datasets such as *YogaTensorDataset* for videos and *YogaSkeletonDataset* for skeletons can independently support one single stream of models. Yet, the real strength of the framework lies in its combination.

3.5.2 Constructing a Paired Multimodal Instance

The DualYogaDataset represents (as shown in Figure 3.8) the logical connection between both datasets. Within, it creates both visual and geometry datasets on the same set of IDs and labels. Most importantly, it determines the set of IDs in common among the two datasets, thus discarding all sequences in which one modality is not present. This is crucial to ensure the one-to-one relationship between samples: every sample passed to the model contains both video tensor and skeleton tensor extracted from the same action clip.

For a training sample, the data loader will first load the tensor corresponding to the video frame and its associated label from the visual dataset, as well as the tensor for the skeleton from the geometric dataset. The label will only be loaded once (from the visual modality, as both use the same mapping) and will be combined along with the two inputs into a tuple of values $(\mathbf{V}, \mathbf{S}, y)$. No synchronization needs to be performed as both data modalities were already synchronized onto the same 30-frame timeline. The result is an avoidance of time warping for heterogenous datasets and ensures that, at each time step, the visual frame and skeleton pose match up to the real-world time.

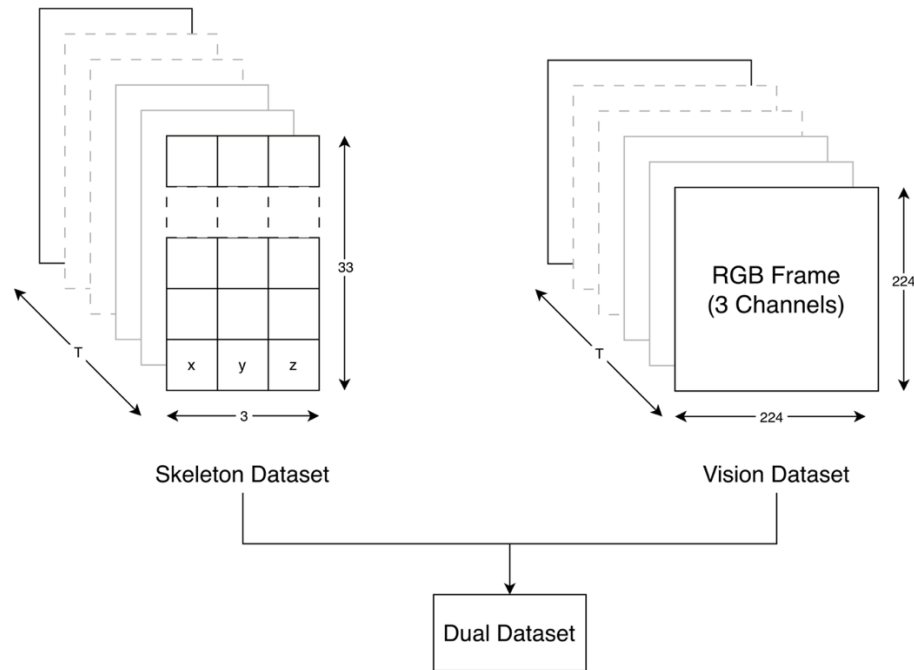


Figure 3.8: Dual Multi-modal Dataset formation

3.6 CLASS REBALANCING STRATEGIES

In real-world action recognition datasets, the distribution of the different classes is typically not balanced as shown in Figure 3.9. The 3DYoga90 dataset is no different; some yoga postures appear more frequently in the data than others, resulting in a long tail distribution which affects the training process [4]. Without intervention, this will make the model biased towards learning majority classes, sometimes even ignoring the smaller classes which could hold more significance. In order to overcome this challenge, this study employs two strategies of rebalancing; loss weighting and sampling weighting.

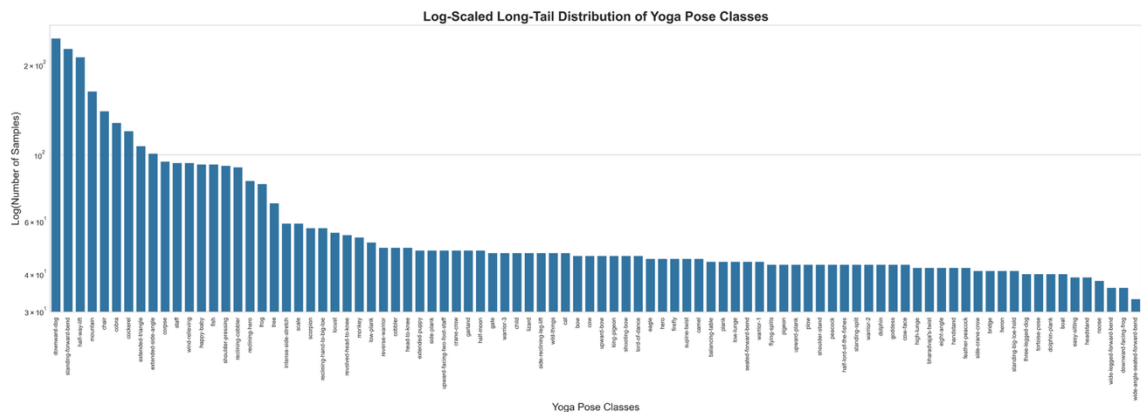


Figure 3.9: Log-Scaled Long-Tail Distribution of Yoga Poses showing significant class imbalance

3.6.2 The Class Imbalance Problem in Supervised Learning

Mitigation techniques can be classified into two broad groups: techniques at the data level where the distribution of samples in the training dataset is adjusted, and techniques at the algorithm level where the loss function is modified so as to assign higher penalties for misclassification in minority classes [41]. The two techniques included in this codebase implement one example of each classification method.

3.6.3 Loss-Based Rebalancing: Inverse-Frequency Class Weights

Inverse frequency-based weighting techniques have been around in machine learning algorithms for quite some time now. Cost-sensitive learning approaches [42] as well as modifications made for deep neural networks [43] use similar strategies of giving high weights to losses of less frequent classes in order not to overlook them while training. Such an approach makes sense whenever the minority classes have distinctive meaning and value, just like different yoga poses that require different levels of mechanical complexity without having any hierarchy. A downside of using inverse frequency weights lies in the possibility of amplifying noise and anomalies within the less represented data. However, since all instances in the 3DYoga90 database are unique poses without any noise and are properly edited, the problem becomes irrelevant [44].

3.6.4 Sampling-Based Rebalancing: Weighted Random Sampling

A weighted random sample has various benefits compared to undersampling. Firstly, weighted random sampling retains all the samples from the majority classes without losing any information but rectifies the class balance of the dataset [45]. Secondly, due to the fact that weighted random sampling occurs at the level of the data loader, the technique does not affect the model and the loss function; hence, it works well with any architecture without needing changes in design like output biases by class. It has also been empirically found that sampling techniques perform on par with or better than loss weighting in image classification [46].

3.7 TRAIN-TEST SPLITTING: THE CROSS-SUBJECT PROTOCOL

A strong evaluation protocol requires that training data and testing data be split in such a way that they cannot easily learn the traits specific to subjects and can be tested for their ability to generalize beyond them. In this case, the data split that satisfies this criterion for the 3DYoga90 dataset [4] is source-based cross-subject splitting where the basis of split is the unique source from which the videos have been captured (instructor).

3.7.1 The Perils of Random Instance-Level Splitting

If video instances are divided randomly into training and testing sets, clips with the same performer could end up in either set. Although computationally easy to perform, it creates an implicit and very dangerous form of leakage between datasets. Both CNNs and transformers are good at picking up correlations within a particular individual: facial

appearance, body form, attire, and even motion style. Thus, when evaluating the model using the random split method, it would be possible to get misleadingly high accuracy based not on the action itself but the identity of its performer [5].

This issue becomes more critical in cases like yoga because only a few trainers provide the most samples. This would mean that there will be samples from the same trainer in both sets, and the network can use postures unique to the trainer (such as a specific angle of the knee in a particular pose performed by one trainer). In effect, this evaluation measures closed-set person identification instead of open-set pose classification. The benchmark becomes irrelevant to the practical application because the system needs to generalize beyond the training set.

3.7.2 Source-Based Splitting as a Proxy for Subject Independence

In order to control for actor confounding, the dataset is split using source ID, which is a unique identifier for the source video that corresponds to a video clip. In the 3DYoga90 dataset, each source video contains one person performing various sequences of yoga poses [4]. By putting all clips belonging to the same source in either the training set or the validation set, we can avoid having the same person included in both datasets. This strategy follows the conventional methodology used in cross-subject benchmarking of 3D human action analysis, e.g., NTU RGB+D [7] and NTU RGB+D 120 [47].

The algorithm works in two steps. In the first step, all instance IDs are grouped based on their source, which gives us a mapping between the source and the list of its instances. The sources are then randomly shuffled and split into an 80% training and 20% validation set. Lastly, the instances in each source are put together in order to create the partitions. Since the split was done at the source level, both the training and validation datasets include disjoint sets of teachers, and therefore test the ability of the model to detect yoga postures regardless of body shape and style of practice.

3.7.3 Statistical Rationale and Validation

The 80/20 rule has become an accepted heuristic which allows the acquisition of enough samples for convergence of the model, while still having a sufficient number of samples for the validation set to achieve stable estimates on performance. The 80/20 rule is applied to the number of sources instead of the number of instances to ensure that there is representation of poses, though from different subjects in each partition. Source-overlap checks then follow to make sure that none of the sources are common in the two partitions. Figure 3.10 shows a Sankey diagram for the train-test split.



Figure 3.10: Sankey Diagram showing Cross Subject Dataset Split

3.8 EVALUATION METRICS

Assessing the performance of a multi-class action recognition model is a process that entails using a number of metrics which together measure the overall accuracy of the model, the sensitivity of the model to unbalanced class distributions, and the capability of the model to discriminate between various pose classes. The Evaluation metrics utility calculates five metrics based on the results generated by the model when presented with the test dataset. These metrics include accuracy, balanced accuracy, macro average F1-score, confusion matrix, and recall for each pose class.

3.8.1 Overall Accuracy and Its Shortcomings

Overall accuracy is the most straightforward and easy to interpret performance measure, which is given by the ratio of correctly classified samples to the total number of testing samples as shown in the Equation 3.1:

$$Accuracy = \frac{1}{N} \sum_{i=1}^N 1 [\hat{y}_i = y_i] \quad (3.1)$$

where N is the total number of test samples, $\hat{y}_i$ is the predicted class, and y_i is the ground truth. Although this gives a quick overall picture of how good a classifier is performing, there is a serious limitation of this measure, namely, it fails to give us any information about poor classification results for less prevalent classes [43]. When working with an unbalanced yoga dataset where one class can appear hundred of times, while another just a few (for example, firefly pose versus mountain pose), one can build a classifier which just predicts the majority class and achieves high accuracy, although being completely meaningless clinically.

3.8.2 Balanced Accuracy: Equalising Class Influence

The calculation of the balanced accuracy aims to compensate for the bias towards majority classes through averaging the recall per class and thus treating each pose equally irrespective of its occurrence frequency [48] as shown in Equation 3.2:

$$\text{Balanced Accuracy} = \frac{1}{C} \sum_{c=1}^C \frac{TP_c}{TP_c + FN_c} \quad (3.2)$$

where C is the total number of classes, TP_c is the true positives of class c , and FN_c is the false negatives. In fact, this measure represents the accuracy expected when the test set is perfectly balanced. Balanced accuracy is an especially suitable choice in the case of the 3DYoga90 dataset since the cross-subject split in Section 3.7 deliberately leads to a skewed distribution of the training set compared to the distribution of the validation set in terms of subject as well as possibly class occurrences. Comparing the metrics alongside raw accuracy can demonstrate the reason behind improved results: if they are attributed to better results for majority classes only or not [32].

3.8.3 Macro-Averaged F1 Score: Harmonising Precision and Recall

The F1 score represents the harmonic mean of precision and recall for one particular class. However, when computing macro-averaged F1 scores in a multi-class setup, the F1 scores of all classes are computed separately before averaging their values equally [49] as shown in Equation 3.3:

$$F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (3.3)$$

$$\text{Macro F1} = \frac{1}{C} \sum_{c=1}^C F1_c \quad (3.4)$$

where,

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \quad (3.5)$$

The macro-F1 score takes into account both the false positives and false negatives, which makes it a stricter measure compared to balanced accuracy. In case of pose estimation for yoga practice, having a high macro-F1 implies that the algorithm not only correctly identifies most of the occurrences of the pose (recall), but also doesn't confuse poses with each other (precision). It is very important in the process of instructing since a false positive may cause the algorithm to provide incorrect recommendations. The previous studies in skeleton-based action recognition have used the macro-F1 score as an

evaluation measure since it negatively rewards both over- and underestimating specific actions [18].

3.8.4 Combined Interpretation and Practical Use

The use of all five measures together obeys the principle that a number cannot fully represent the multidimensional nature of multi-class classification tasks [50]. While accuracy provides a simple snapshot, it should be supplemented by other measures for proper evaluation. Insensitive to the distribution of classes, balanced accuracy and F1 provide the first two numbers that determine the quality of the classifier and used as main criteria for hyperparameter tuning and model selection. For the dual-modal yoga recognition task, all three metrics are calculated for vision-based, skeleton-based, and fusion approaches alike.

In compliance with the evaluation procedure suggested in the action recognition literature [7], the current study contributes to reliable and reproducible research.

3.9 CTR-GCN ARCHITECTURE: SKELETON-ONLY BASELINE

Graph Convolutional Neural Networks (GCNs) have emerged as the default technique for skeleton-based action recognition through the use of a graph that represents the human body. Although previous work such as the ST-GCN [18] has proved the value of such a representation, it was based on predefined adjacency matrices of the bones of the human skeleton. However, it is evident that such fixed representations fail to account for action-dependent kinematics. For example, there is no connection between the hands and the feet in the physical skeleton; however, their spatial correlation determines the difference between yoga poses like Warrior I and Warrior II.

As a solution to this limitation, we select the Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN) [6] as our geometry stream baseline model. Contrary to previous adaptive networks which learn only one global topology [19], the CTR-GCN architecture learns unique adjacency matrices per channel using a cross-channel attention approach. The pairwise similarity of the joints per channel is computed based on a given feature map, and as a result, each channel focuses on particular kinematic features, such as joints of limbs in case of balance yoga poses compared to torso joints in seated poses [6]. Such refined channel information is used in addition to the existing static anatomical priors from MediaPipe.

3.9.1 Overall Architecture and Data Adaptation

The entire network architecture (Figure 3.11) is tailored for processing the sequences of 3D world coordinates obtained using 33-joint MediaPipe [51] pipeline. The initial input tensor, (C_{in}, T, V) , where each dimension corresponds to coordinate, frames, and joint respectively, is successively projected to higher-level features using six spatial-temporal layers. The global average pooling operation collapses the two dimensions into a 256-dimensional vector.

The CTR-GCN benefits from channel-wise topology optimization and functions as an extremely compact yet powerful geometry expert. Within the scope of our multimodal architecture, this particular network is pretrained separately and frozen thereafter. The logits predicted by this network serve as the viewpoint invariant and kinematically sound ground truth for the refinement of our cross-attention fusion.

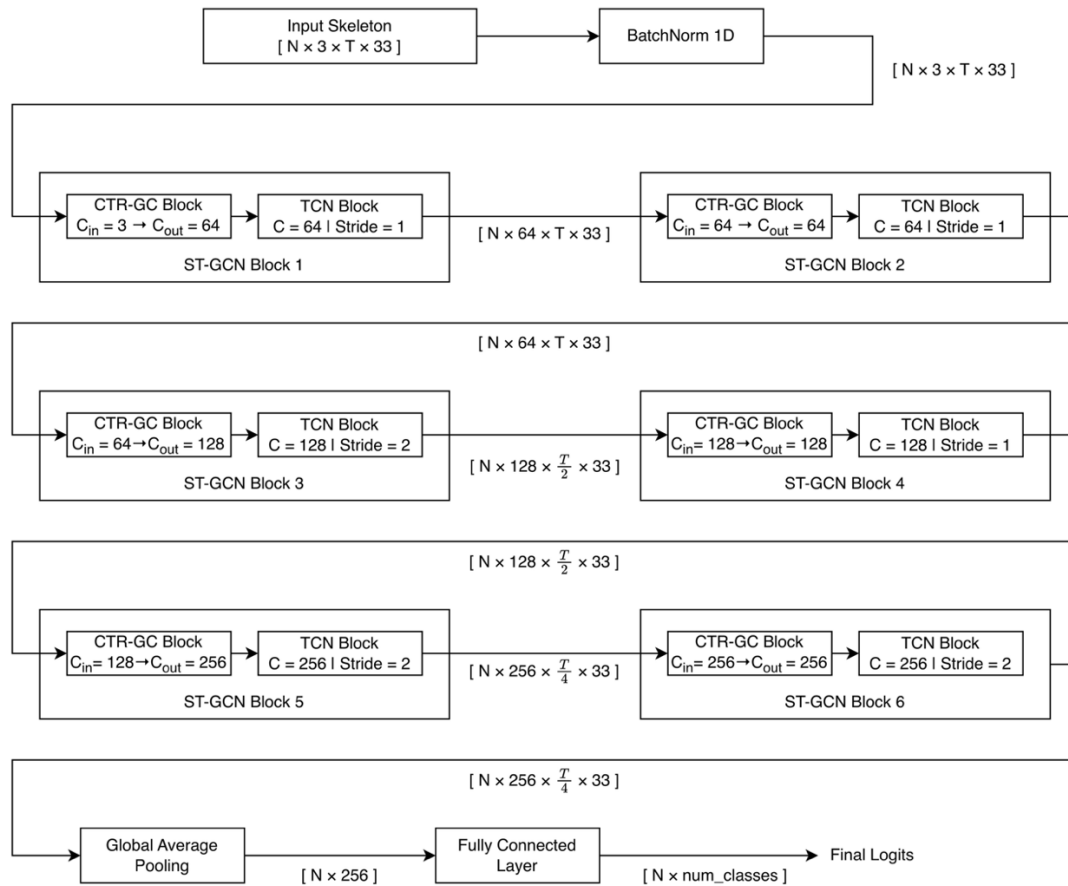


Figure 3.11: CTR-GCN: Overall Architecture

3.9.2 Role as a Baseline and Foundation for Multimodal Fusion

Within the larger multimodal pipeline, the CTR GCN operates as the geometry stream, feeding on the hip centred 3D skeletal coordinates obtained by analysing each trimmed yoga clip. The choice of the CTR GCN is informed by its superior state of the art performance on skeleton based datasets [3], while also being flexible enough to enable extraction of feature tokens. This can be done through pooling of the feature tensor produced by the final spatial temporal block, which has dimensions $256 \times T' \times V$, where T' represents a reduced temporal dimensionality. Later on, the CTR-GCN will produce its feature maps in order to project them into a common joint embedding space, and then

fuse them with visual feature vectors using a cross-attention network. To precisely measure the contribution made by multimodal fusion and cross-modal attention, it is first necessary to establish CTR GCN as a competitive unimodal baseline.

3.10 VIDEO SWIN TRANSFORMER: APPEARANCE-BASED BASELINE

Although skeletons offer a concise and viewpoint-independent description of human motions, this approach neglects the additional context information available in video data such as the practitioner's alignment to the surroundings, clothing signals, and detailed visual cues on the texture of the mat and the background, which could help distinguish between subtle poses. As an accompanying unimodal input channel that complements the geometric CTR-GCN, we use a vision-only baseline consisting of the state-of-the-art video transformer architecture known as the Video Swin Transformer [23]. This network takes the preprocessed video tensor as input, capturing visual information in terms of appearance and motion in an end-to-end manner.

3.10.1 Transfer Learning from Kinetics-400

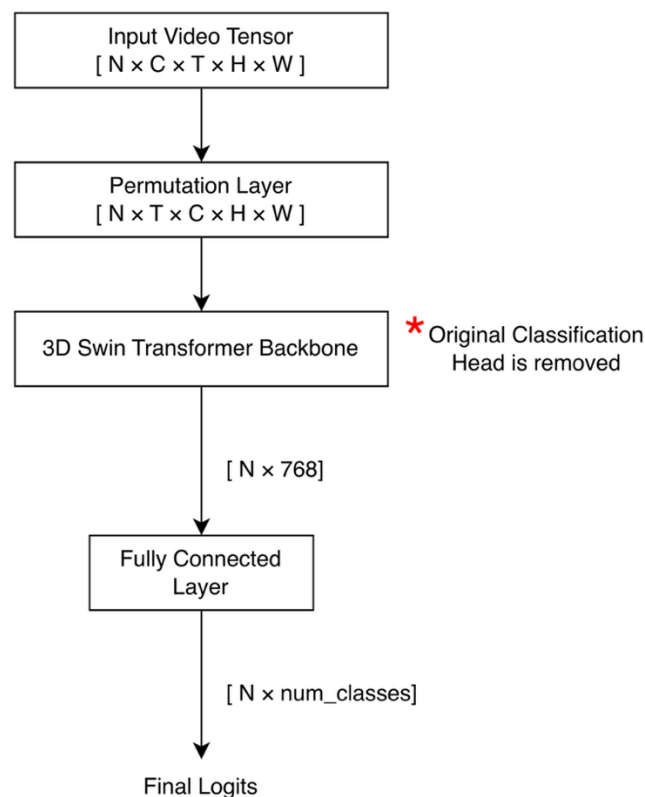


Figure 3.12: Video Swin Transformer: Updated classification head to support Yoga Pose Classification

Starting from scratch while training the model based on a dataset like 3DYoga90 is not a wise choice since it may not be very data efficient, and overfitting may be one issue. As a result, the backbone will be initiated using a pre-trained model on Kinetics 400 [26], a large-scale dataset used for action recognition tasks with 400 classes of human actions covering various actions from sports, daily activities, and interacting people. This dataset offers a wide variety of motion types and backgrounds, and pre-trained models have been observed to generalize well to other video tasks [23]. The model pre trained on this dataset has encoded a vocabulary of features related to optical flow, motion boundaries, and body part detectors, all of which would serve as a good foundation for yoga-specific features.

All the layers of the backbone architecture are carefully tuned because freezing earlier layers is generally disadvantageous in case the target and the source domains differ greatly in terms of the look (yoga studios, indoor environment) and action semantics (static postures versus dynamic moves). The original classifier is substituted with a simple linear model that takes a 768-dimensional pooled feature and outputs predictions for C yoga poses as shown in Figure 3.12.

3.10.2 Role as a Vision-Only Baseline and Future Fusion

Video Swin baseline is trained separately to determine the upper bound of performance attainable from appearance features alone. It will provide a benchmark against which performances from the skeleton only CTR-GCN model and our proposed multimodal fusion model will be compared. For the fusion architecture, Video Swin will be pruned before the global pooling block and the multiscale feature maps extracted by the network will be converted to a sequence of visual tokens. Geometry tokens from the CTR-GCN will then be added to the visual tokens to construct a cross-attention layer for the purpose of aligning visual and skeletal feature embeddings. The hierarchical structure of Video Swin and window-based approach allows this task to be efficiently accomplished since the intermediate feature maps outputted by the backbone network can be split into small spatiotemporal patches that behave as individual tokens. By having a vision stream that has been pre trained to extract very strong features, the subsequent cross modal attention mechanism will not have the extra burden of compensating for poor unimodal feature representations.

3.11 MULTIMODAL FUSION VIA LOGIT-LEVEL CROSS-ATTENTION

Two unimodal baselines have been introduced in previous sections, namely, Skeleton based CTR-GCN working in 3D world coordinates and Video-based Swin Transformer trained on RGB frames. Both perform well individually, but there is scope for improving performance further by utilizing the complementary information offered by appearance and geometric cues via a fusion approach. The last architecture proposed in this work, referred to as Logit-Level Cross-Attention Fusion, is the result of fusing the pre-trained frozen backbone models using a cross-attention head trained on the outputs produced by the logit heads of these backbones.

3.11.1 Design Rationale: Frozen Experts and Late Fusion

The task of combining two deep neural networks into one is usually solved at the feature-level stage by concatenating or aligning the intermediate representations before feeding them into the final classifier [8], [28]. Nevertheless, this strategy requires joint training or fine-tuning of both networks simultaneously, which entails high computational and memory costs, particularly in the case of video transformers using 3D attention. Another strategy, based on recent research on combining pre-trained models [29], [52], is freezing both encoders completely and using their outputs to feed into a separate decision-level fusion network. Thus, we decouple the process of tuning the streams separately from their combination in the latter, thus making it possible to focus each encoder specifically on its domain of expertise without interfering with other components.

There are various reasons why the combination of models is performed at the logit level rather than at the feature level. First, logits capture compactly the information contained in a set of features representing the beliefs of each modality regarding all possible classes in a single vector of evidence. Second, combining models at the level of logits is consistent with the idea of classifier stacking and mixtures of experts [9]. In the latter case, the meta-classifier modifies the outputs produced by the expert models. Finally, such an approach allows us to train a small fusion network.

Another vital decision made in the above framework is treating the skeleton stream as a prior estimator, and the video stream as a corrective term to the skeleton logits. The rationale behind such a decision is based on the fact that the CTR-GCN, which uses 3D coordinates as input, is less prone to variation from the background and clothing; hence, the skeleton stream will produce better estimates than the video stream. On the other hand, the video stream excels at capturing appearance-based information that can be used to discriminate poses with similar joint positions but distinct hand positions.

3.11.2 The *LogitPerceiver*: Cross-Attention over Class Predictions

The fusion head, which we call the *LogitPerceiver*, is based on the idea of the Perceiver model [53], which employs a learnable query mechanism to focus on a collection of input tokens. Here, the input tokens correspond to projected logit vectors for the two modalities, while the query corresponds to a learnable vector combining information from both modalities. The resulting output from the cross-attention layer is then used to compute a residual correction on the logits corresponding to the skeleton.

3.11.2.1 Tokenisation of Logits

Each backbone produces a logit vector $\mathbf{l}_{vis}, \mathbf{l}_{geo} \in \mathbb{R}^C$, where C is the number of classes. These vectors are projected into a common D -dimensional embedding space (with $D = 384$ in our implementation) using a shared linear layer:

$$\mathbf{t}_{vis} = \mathbf{W}_p \mathbf{l}_{vis}, \quad \mathbf{t}_{geo} = \mathbf{W}_p \mathbf{l}_{geo} \quad (3.6)$$

The two token vectors are stacked to form a set $\mathbf{T} \in \mathbb{R}^{2 \times D}$ that serves as the key-value input to the cross-attention module. The use of a shared projection encourages the model to represent both modality logits in a comparable metric space, facilitating meaningful attention interactions. Figure 3.13 shows the formation of the stacked tokens.

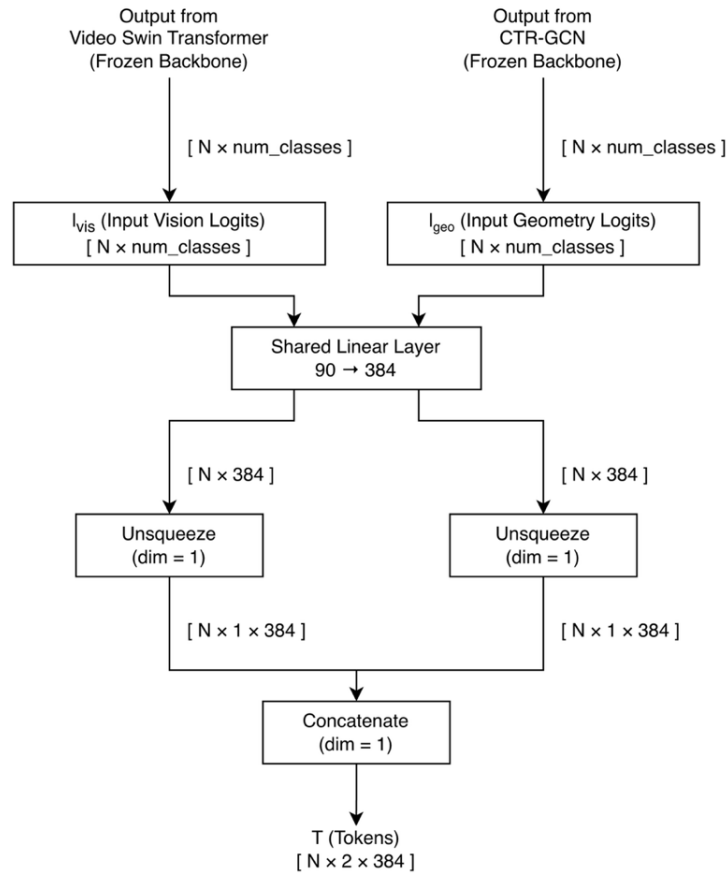


Figure 3.13: LogitPerceiver: Tokenisation of Logits

3.11.2.2 Learnable Query and Cross-Attention

A single learnable query vector $\mathbf{Q} \in \mathbb{R}^{1 \times D}$ is initialised randomly and expanded across the batch. Multi-head cross-attention [11] is then applied with $\mathbf{q}$ as the query and $\mathbf{T}$ as both key and value as shown in Equation 3.7:

$$\mathbf{Q}' = \text{CrossAttn}(\mathbf{Q}, \mathbf{T}, \mathbf{T}) \quad (3.7)$$

This operation allows the query to pool information from both the visual and geometric logit tokens, weighting their relevance dynamically. The query can be interpreted as a learned meta-learner that asks: "given the two experts' opinions, what correction should be applied to the skeleton's prediction?" Because the cross-attention is computed with only two tokens, its computational cost is negligible, enabling rapid

training and inference. A residual connection followed by layer normalisation [54] is applied to the query as shown in Equation 3.8 and Figure 3.14.

$$\mathbf{Q}'' = \text{LayerNorm}(\mathbf{Q} + \mathbf{Q}')$$
(3.8)

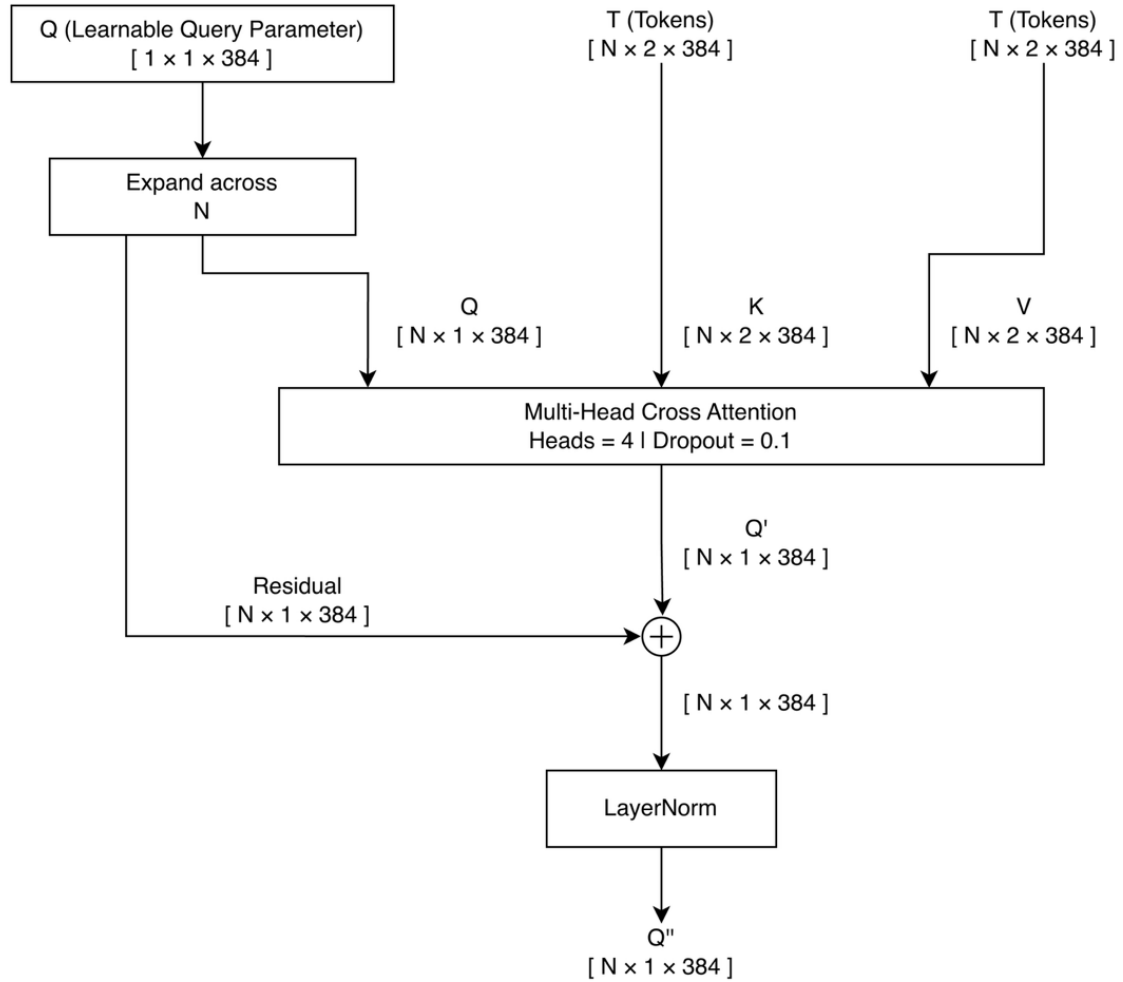


Figure 3.14: LogitPerceiver: Learnable Query and Multi-head Cross-Attention

3.11.2.3 Feed-Forward Refinement

The attended query is further processed by a feed-forward network (FFN) consisting of a linear expansion to $4D$, a GELU activation [55], dropout [56], and a linear projection back to D as shown in Equation 3.9:

$$\mathbf{Q}''' = \text{FFN}(\mathbf{Q}'')$$
(3.9)

A second residual connection and layer normalisation are applied, yielding the final hidden representation $\mathbf{H} \in \mathbb{R}^D$. This FFN provides the non-linear capacity needed to

model complex interactions between the two modalities' class beliefs. Figure 3.15 and Figure 3.16 shows the FFN architecture and residual connections.

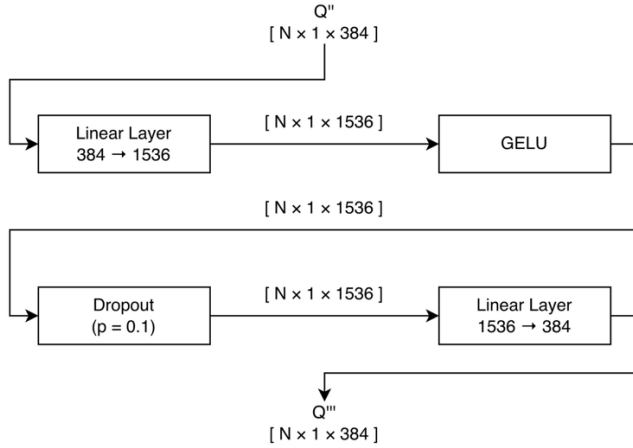


Figure 3.15: LogitPerceiver: Feed-Forward Refinement

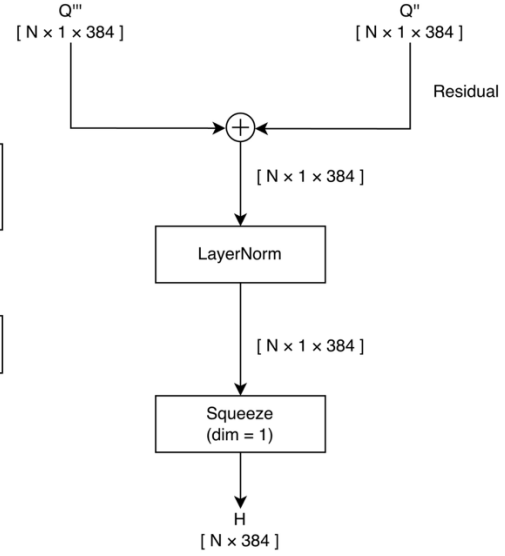


Figure 3.16: LogitPerceiver: Building the hidden representation from Residual

3.11.2.4 Residual Delta Prediction and Gating

The hidden representation $\mathbf{h}$ is then mapped to a delta logit vector $\boldsymbol{\delta} \in \mathbb{R}^C$ through a two-layer projection with intermediate GELU and dropout as shown in Equation 3.10:

$$\boldsymbol{\delta} = \text{Proj}_{\boldsymbol{\delta}}(\mathbf{h}) \quad (3.10)$$

The final fused logits are obtained by a residual connection to the skeleton logits, controlled by a learnable scalar gate α as shown in Equation 3.11:

$$\mathbf{l}_{fused} = \mathbf{l}_{geo} + \tanh(\alpha) \cdot \boldsymbol{\delta} \quad (3.11)$$

The gating mechanism is pivotal. Initialised at zero, $\tanh(\alpha) \approx 0$ at the start of training, meaning the model initially outputs exactly the skeleton logits. As training progresses, the gate can open to allow the visual-derived correction to influence the prediction, but the hyperbolic tangent bounds the gating factor to $(-1,1)$, preventing the delta from overwhelming the geometry base. This design reflects the intuition that the skeleton stream is the primary predictor, while the vision stream provides a bounded refinement.

Residual architecture, similar to ResNet [36], helps in reducing optimization difficulties for the model, making it learn the identity function in case extra modeling power is not required. Figure 3.17 and Figure 3.18 shows the Delta projection and fused logits.

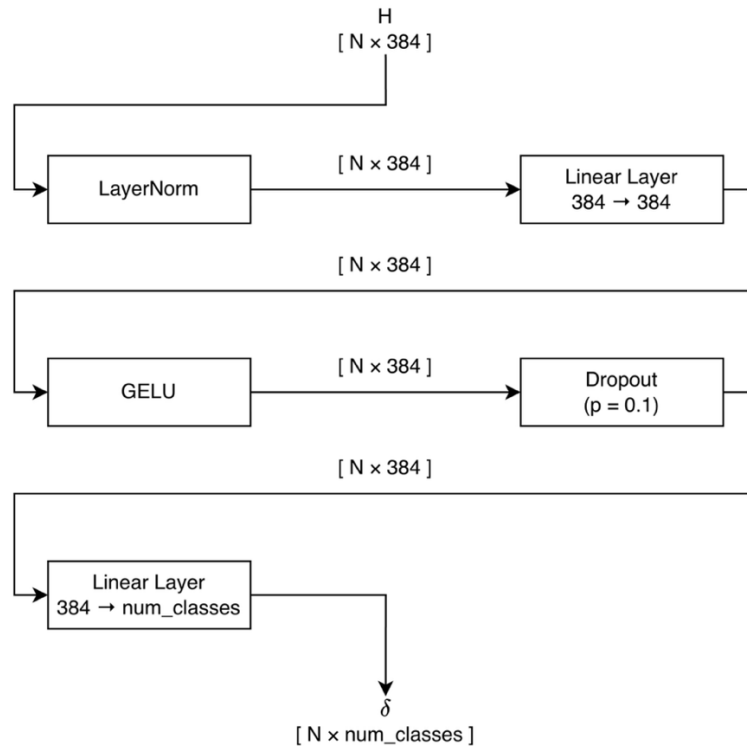


Figure 3.17: LogitPerceiver: Delta Projection

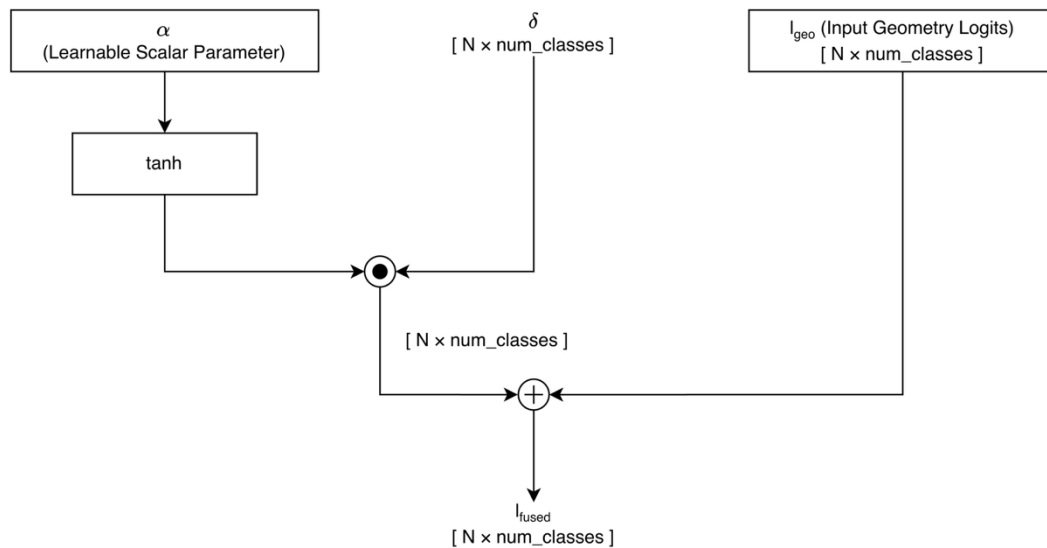


Figure 3.18: LogitPerceiver: Gating and Fused Logits

3.11.3 Significance as the Final Model

The logit-level cross-attention fusion network represents the final architectural structure used in the multiform yoga pose recognition framework. In the process of fusing information from the 3D geometric shape and RGB imagery modalities, the method leverages the strengths of both forms of representation without having to fine-tune the joint information. The motivation behind this approach is based on domain knowledge, as the essence of yoga poses lies in their joint structure, though visual information can be helpful in resolving certain ambiguities. As a result, the model can decide, through the learnable cross-attention query, whether or not to give weight to either modality of representation depending on the situation.

Practically speaking, studies have proven that this kind of logit-level fusion performs at least equally well as feature-level fusion when the separate models themselves are strong enough and properly calibrated. As such, leveraging existing state-of-the-art backbones as a frozen module, this architecture can be considered an efficient way of doing 3D yoga pose recognition, and its modularity allows for further exploration into other modalities or encoders in future work.



CHAPTER 4

EXPERIMENTAL SETUP AND RESULTS

4.1 HARDWARE AND IMPLEMENTATION DETAILS

Training and testing of all three architectures: the CTR-GCN skeleton network, Video Swin Transformer, and the logit-level cross-attention fusion framework were conducted using a NVIDIA RTX 5070 GPU instance available on cloud platforms. The purpose of this section is to outline the computation resources, software settings, and memory management tools utilized for the implementation of spatiotemporal deep learning using only a single GPU.

4.1.1 Software Environment and Library Stack

The whole source code was developed using Python 3.10 and depended on PyTorch deep learning library (v.2.1), which is the principal library for tensor computations and auto differentiation. The main libraries used in the process of pipeline creation are listed below:

- **torchvision (v.0.16):** Contributed the pre-trained Video Swin Transformer model backbone and associated weights on the Kinetics-400 classification task, as well as affine and flipping transformation for data augmentation purposes.
- **torch.amp:** Automatic mixed precision module in PyTorch, used to cast forward/backward passes in half precision (float16) and thus decrease GPU memory usage and speed up training using tensor cores [59].
- **CUDA/cuDNN:** Deterministic cuDNN routines were enforced to ensure reproducibility, while cuDNN auto-tuning was switched off to avoid non-deterministic algorithms being selected.
- **NumPy and Pandas:** Underpinning libraries used for all array manipulation and working with tabular data, especially in the skeleton dataset class where 3D coordinates of landmarks are reshaped and resampled.
- **scikit-learn:** Source of implementations of balanced accuracy, macro-average F1-score, and confusion matrix to measure classifier performance.
- **MediaPipe (v.0.10):** Pose landmarker is utilized at the stage of skeleton extraction, providing 33 landmarks expressed as coordinates in world coordinates from each frame of video.
- **OpenCV (cv2):** Used for reading video files, flipping frames, and converting between colour spaces at the stage of skeleton extraction.
- **tqdm:** Progress bars were provided by this library during data loading and training stages.
- **Matplotlib and Seaborn:** For visualization purposes

4.1.2 GPU Configuration and Mixed-Precision Training

Because of the limited capacity of the RTX 5070 with 12 GB RAM, it was necessary to train the video-based models using mixed precision. For this, the GradScaler from the PyTorch library scaled the loss to avoid underflow of the gradients for half precision, while the `torch.autocast` feature converted eligible mathematical operations (such as convolutions and matrix multiplication) into float16 data type [57]. The graph convolution network (CTR-GCN), having a light-weight structure, could work with batch size 32 using full precision, but mixed precision was still utilized.

In case of the Video Swin Transformer that could take in only 4 instances for batch size in float32, mixed precision helped double the available memory, making sure that the gain in speed was achieved while keeping the batch size unchanged. For the fusion model, training of the light-weight cross attention head on top of frozen backbones meant that even a batch size of 16 worked just fine.

Further stabilising the training process, gradient clipping was done up to the maximum of norm equal to 1.0 for all model architectures. Flags for the CUDA memory allocator were used to decrease memory fragmentation in lengthy training processes. Data tensors were transferred to the GPU using pinned memory and multi-workers for data loading.

To conclude, with the aid of a contemporary consumer-level GPU, a well-packaged software application, and meticulous memory management through mixed precision, three deep networks were successfully trained at low cost, illustrating that it is possible to achieve state-of-the-art multimodal yoga pose detection without requiring large-scale resources like those found in data centres.

4.2 COMMON TRAINING PROTOCOLS

4.2.1 Optimisation Algorithm: AdamW

Network updates are made through the use of the AdamW Optimizer [58], which is a modified version of Adam and separates weight decay from the gradient descent update rule. For regularized Adam training where there is an L_2 regularization penalty on top of the usual Adam learning scheme, the effect is that the weight decay component gets multiplied with the adaptive learning rate scheme. The net result is that there is an unintended increase in regularization for parameters whose history of gradients was small compared to the rest. The key difference with AdamW is that weight decay acts independently of the adaptive gradient descent.

4.2.2 Learning Rate Schedule: Cosine Annealing

The learning rate is decayed from the initial value following a cosine annealing schedule without restarts [59] as shown in Equation 4.1:

18

$$\eta(t) = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \left(1 + \cos\left(\pi \frac{t}{T}\right)\right) \quad (4.1)$$

where t is the epoch counter, $T = 50$ is the number of epochs, $\eta_{\max} = 10^{-3}$ and $\eta_{\min} = 10^{-6}$. Cosine annealing was selected on the basis of its gradual and monotonic decay pattern, which guarantees a natural decrease in the step size as the network moves towards the local minimum, compared to a sudden drop of the step-based schedules, that could lead to unstable optimisation [59]. The minimum step size of 10^{-6} is critical in allowing the network to update its weights throughout all the epochs, especially since early stopping is used here.

4.2.3 Early Stopping and Model Selection

To ensure the absence of overfitting to the training dataset and selection of the model with optimal generalisation capabilities, the training process employs an approach called early stopping with patience. The metric on the basis of which the model is selected is balanced accuracy on the validation dataset, taking into account that all poses are treated equally, not considering their frequencies. A new model is saved only if its balanced accuracy exceeds the accuracy of the previous model by at least 0.1%. This value of threshold prevents selecting a new model because of random fluctuation. Training is stopped if there is no improvement in 10 consecutive epochs.

4.3 TRAINING PROTOCOL FOR SKELETON-ONLY CTR-GCN

The effectiveness of a deep neural network depends on both the model's architecture and the choice of the optimisation approach, regularisation schemes, and data handling methods used throughout training. The following section discusses the training methodology used in training the CTR-GCN model for skeleton-based yoga pose recognition. Each of these decisions is informed by the unique characteristics of the 3DYoga90 dataset: relatively small size, imbalanced classes, and cross-subject generalisability.

4.3.1 Data Sampling Strategy

In addition to the loss-level correction, a balance at the data level is obtained via a weighted random sampling approach. Sampling weights of training samples are computed based on the inverse of the frequency of the corresponding classes. As mini-batches are selected for creating epochs, rare classes will be oversampled. Thus, at each iteration of the learning process, a roughly balanced set of examples will be observed for computing gradients, which helps the classifier stay away from majority-class bias [41]. This approach works along with the use of class-based weighted loss; although there is an overlap between the two techniques, their combined use guarantees a safer path towards balanced training and has proven to work consistently well [60].

4.3.2 Summary of Hyperparameters

Table 4.1: Training Hyperparameters for CTR-GCN

Parameter	Value
Optimiser	AdamW
Learning rate (initial)	1×10^{-3}
Weight decay	1×10^{-2}
LR schedule	Cosine annealing to 10^{-6}
Epochs (maximum)	50
Batch size	32
Loss function	Cross-entropy + label smoothing (0.03)
Class weighting	Inverse frequency
Training sampler	Weighted random (balanced)
Mixed precision	Float16 (AMP)
Gradient clipping max	1.0
Early stopping patience	10 epochs
Early stopping threshold	0.1% balanced accuracy

Table 4.1 shows the list of hyperparameters used for training. Through this well-designed training process, the CTR-GCN is able to perform excellently in the skeleton-only yoga pose recognition while remaining stable and repeatable, thus forming an excellent baseline for future research in multimodal fusion of yoga recognition.

4.4 TRAINING PROTOCOL FOR VIDEO SWIN TRANSFORMER

Selection of Video Swin Transformer [23] for the appearance-based stream of the multimodal framework took place. With respect to the moderate number of samples in the 3DYoga90 dataset, training a deep three-dimensional transformer de novo is data hungry and leads to an extremely poor generalization, thus, overfitting. Instead, we have resorted to a transfer learning approach, which consists in fine-tuning the pre-trained network on action recognition on the target task.

4.4.1 Transfer Learning from Kinetics-400

The Video Swin backbone model was pretrained using the weights obtained from Kinetics-400 [26] database comprising around 240,000 videos belonging to 400 different classes of human actions. The use of Kinetics dataset for pretraining results in the acquisition of a diverse feature vocabulary by the model including detectors of edges, motion boundaries, and even parts, which are generally useful for video analysis tasks [61]. All layers of the Video Swin model, including the early stages of 3D patch embeddings and later transformer modules, were fine-tuned on the 3DYoga90 dataset.

Freezing of early layers is a common strategy adopted when there is a high visual similarity between the source and target datasets; however, the use of this approach is not recommended in the present case due to the significant differences between the indoor yoga environment and Kinetics.

4.4.2 Summary of Training Hyperparameters

Table 4.2: Training Hyperparameters for Video Swin Transformer

Parameter	Value
Pre-training dataset	Kinetics-400
Fine-tuning strategy	Full (all layers)
Optimiser	AdamW
Learning rate (initial)	1×10^{-4}
Weight decay	1×10^{-2}
LR schedule	Cosine annealing to 10^{-6}
Epochs (maximum)	30
Batch size	4
Loss function	Cross-entropy + label smoothing (0.03)
Class weighting	Inverse frequency
Training sampler	Weighted random (balanced)
Mixed precision	Float16 (AMP)
Gradient clipping max	1.0
Early stopping patience	5 epochs
Early stopping threshold	0.1% balanced accuracy

Table 4.2 shows the list of hyperparameters used for training. The well-thought-out strategy facilitated the adaptation of Video Swin Transformer's highly potent spatiotemporal representations to yoga, allowing for excellent results as a baseline unimodal model and forming a solid foundation for the multimodal approach that was used afterward.

4.5 TRAINING PROTOCOL FOR MULTIMODAL FUSION MODEL

Training of the cross-attention fusion model at the logit level aims at discovering the meta-learner, which should merge the predictions of two trained models. As the backbone models, i.e., Video Swin Transformer, and CTR-GCN, do not undergo changes during training, optimization can be performed only on the fusion head, which makes the process highly efficient, both from the memory and computational point of view. The following subsections describe the training procedure, including optimizer selection and evaluation metrics, providing theoretical rationale behind each step.

4.5.1 Meta-Learning over Frozen Backbones

The fusion model treats the two unimodal backbones as fixed function approximators that output logit vectors $\mathbf{l}_{\text{vis}}$ and $\mathbf{l}_{\text{geo}}$. At the time of training, the disabled torch gradient calculation mechanism ensures that there is zero gradient flow through such large networks, rendering it unnecessary to retain any activations from these networks to perform backpropagation through them. The approach adopted here, which is similar to the concept of classifier stacking [10] and meta learning from mixtures of experts [9], enables the model to utilize the representation capacity of existing advanced video and skeleton modeling architectures without having to train them. All the trainable parameters of the model exist in the LogitPerceiver head, including logit projection, cross attention mechanism, feed forward network, delta projection, and gating scalar; all this accounts for less than 2 million parameters.

4.5.2 Learning Rate Schedule with Warm-Up

The learning rate schedule consists of a linear warm up period, followed by a cosine decay. In the initial three epochs, the learning rate is increased linearly from 0 to 1×10^{-3} . The warming up process is an important step in the training of transformer models due to the fact that the query and key vectors are initialized randomly, hence employing a high learning rate right away would cause the update of attention matrices to be too drastic [62]. Following the warming up phase, the learning rate decays along a half-cosine curve, with the end epoch having a learning rate value of around 5×10^{-7} [11], [63].

4.5.3 Summary of Hyperparameters

Table 4.3: Training Hyperparameters for Logit-Level Cross-Attention Fusion

Parameter	Value
Trainable component	LogitPerceiver head (≈ 1.8 M params)
Backbones	Frozen Video Swin, Frozen CTR-GCN
Optimiser	AdamW
Learning rate (initial)	1×10^{-3}
Weight decay	5×10^{-3}
LR schedule	Linear warm-up (3 epochs) $\rightarrow$ cosine decay to $\sim 5 \times 10^{-7}$
Epochs (maximum)	30
Batch size	16
Loss function	Cross-entropy + label smoothing (0.05)
Class weighting	Inverse frequency
Training sampler	Weighted random (balanced)

Mixed precision	Float16 (AMP)
Gradient clipping max	1.0
Early stopping patience	20 epochs
Early stopping threshold	0.05% balanced accuracy

Table 4.3 shows the list of hyperparameters used for training. The precisely engineered training protocol enables the logit-level cross-attention fusion model to effectively learn how to fuse the complementary capabilities of the two streams without having to incur the exorbitant cost of end-to-end fine-tuning, thus obtaining outstanding results on the 3DYoga90 dataset in a cross-subject setting.

4.6 EVALUATION AND RESULTS

This section presents a comprehensive empirical evaluation of the three architectures: the CTR-GCN skeleton-only baseline, the Video Swin Transformer vision-only baseline, and the proposed logit-level cross-attention fusion model, on the 3DYoga90 dataset under the rigorous cross-subject protocol. We analyse convergence behaviour, compare unimodal and multimodal performance, quantify the fusion gain, and discuss the implications of the observed results in the context of prior work.

4.6.1 Unimodal Baseline Performance

Table 4.4: Comparison of Unimodal Baseline Performances

Model	Accuracy	Balanced Accuracy	F1 Score
CTR-GCN (Skeleton Only) [6]	88.71%	86.96%	86.18%
VideoSwin Transformer (Vision-Only) [23]	83.97%	81.64%	80.52%

Table 4.4 shows the clear comparison of the baseline models. The training and validation curves for the two unimodal backbones are summarised in the subsequent sections. Both models were trained using the class-rebalancing, mixed-precision, and regularisation strategies described in Sections 3.6 and 4.1.2 .

4.6.1.2 CTR-GCN (Skeleton-Only)

The CTR-GCN converged steadily over 50 epochs, with balanced accuracy rising from 8.47 % to a final best of 86.96 % (epoch 45). It showed a traditional learning curve pattern whereby fast improvement occurred first before being followed by a slow improvement phase. The achieved overall accuracy was 88.71% while the macro F1 score was at 86.18%. This means that there were high levels of accuracy in both the minority and majority yoga classes. The decrease in training loss happened from 3.71 to 0.44 while cosine annealing enabled the model to converge to a quality local minimum. In this case,

the best checkpoint was attained after 45 epochs. Figures 4.1 through 4.4 shows the trend of the metrics over training epochs.

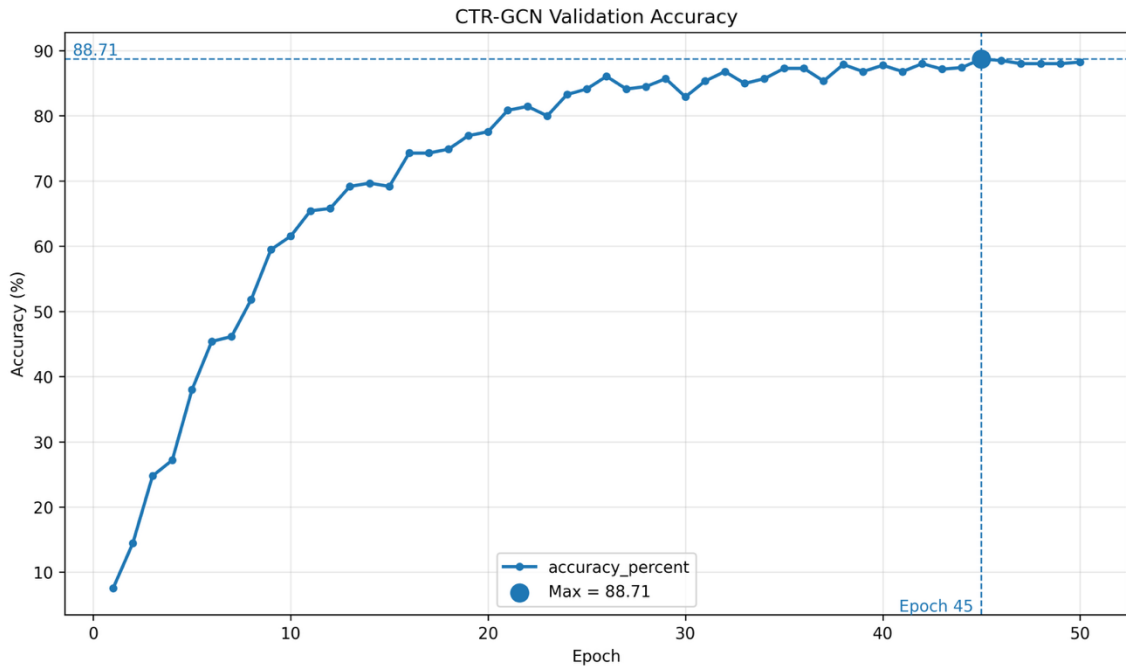


Figure 4.1: CTR-GCN Validation Accuracy curve over epochs

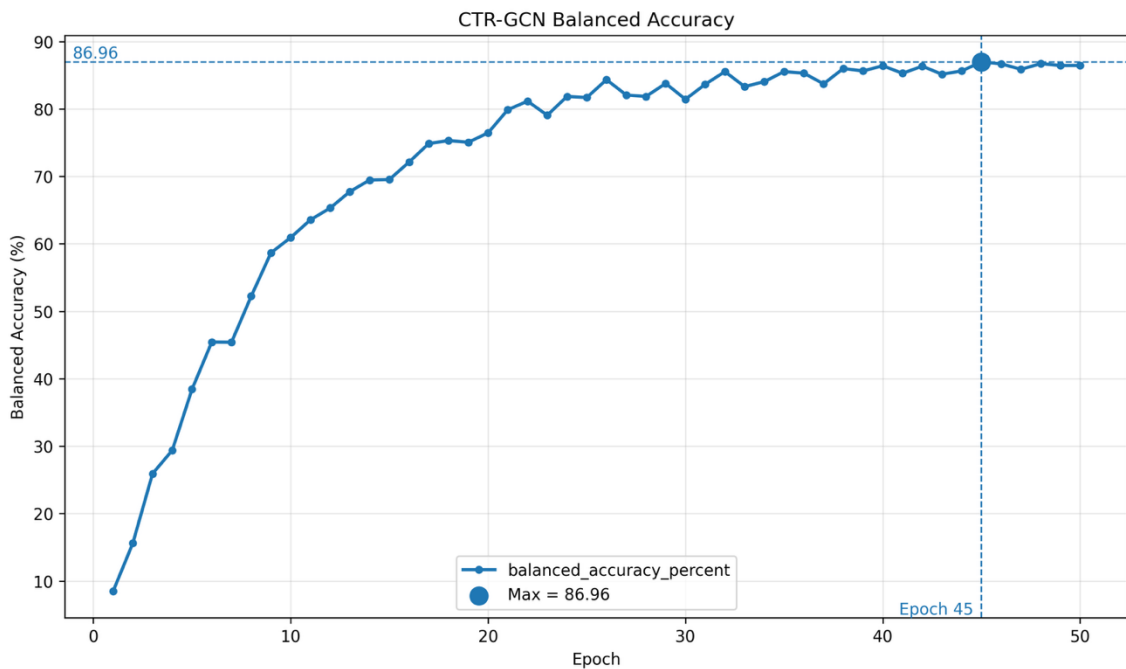


Figure 4.2: CTR-GCN Balanced Accuracy curve over epochs

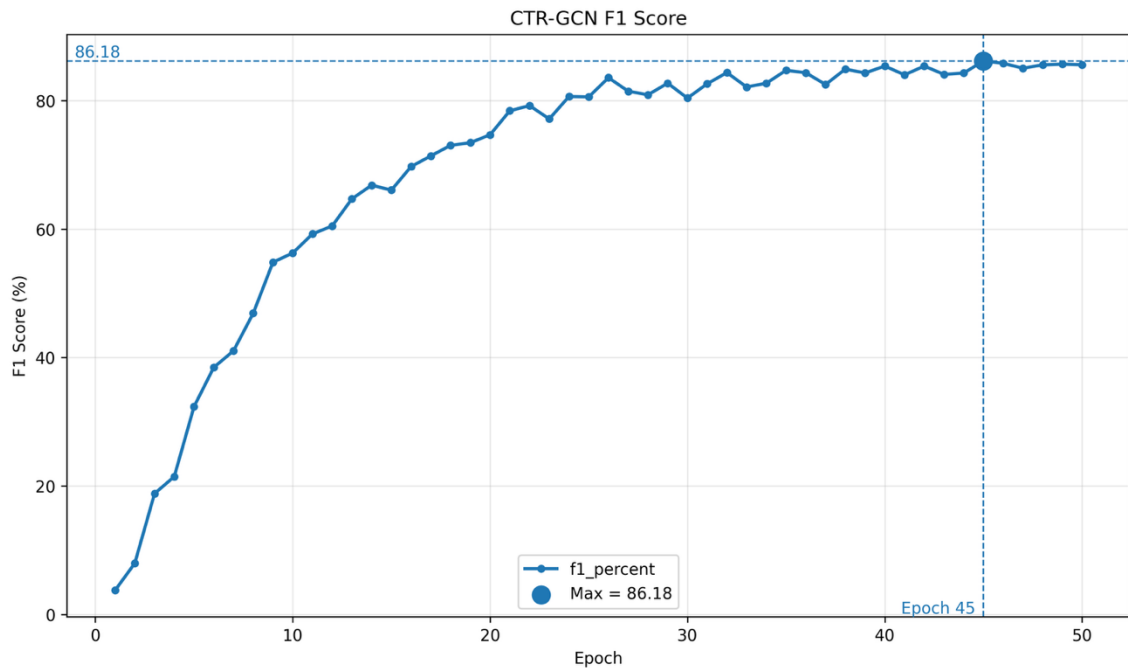


Figure 4.3: CTR-GCN F1 score curve over epochs

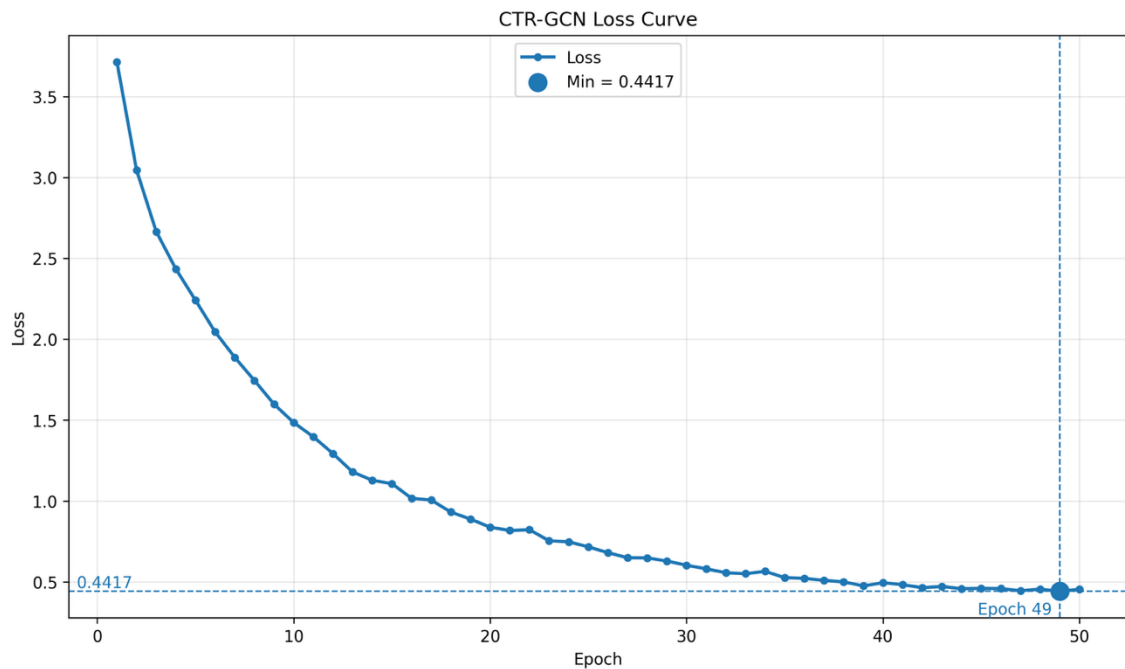


Figure 4.4: CTR-GCN Loss curve over epochs

4.6.1.3 Video Swin Transformer (Vision-Only)

The well-tuned Video Swin Transformer achieved an optimal balanced accuracy score of 81.64 % at epoch 26, reaching an overall accuracy rate of 83.97 % and a macro-F1 of 80.47 %. Convergence took place more quickly in comparison with CTR-GCN, with a plateau established within 20 epochs. This observation conforms to what is common practice when applying transfer learning to the Kinetics-400 [61]. The poorer result observed in contrast to the skeleton-based approach follows from the expected fact that yogic poses have mostly been characterised based on their joint positions [7]. Figures 4.5 through 4.8 shows the trend of the metrics over training epochs.

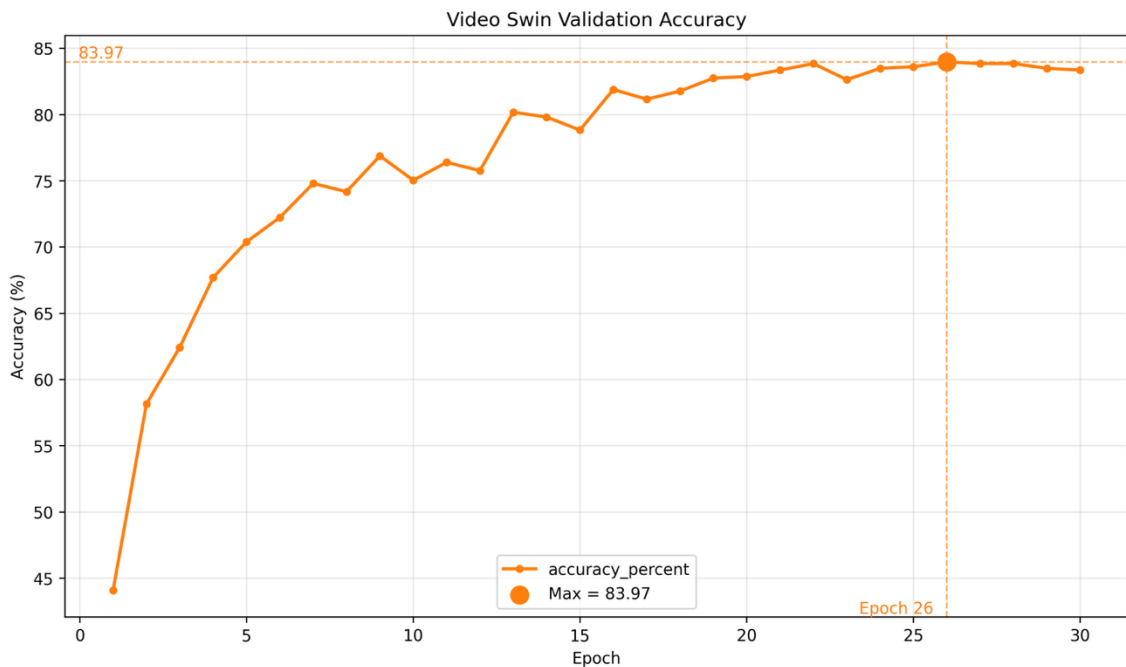


Figure 4.5: Video Swin Transformer Validation Accuracy curve over epochs

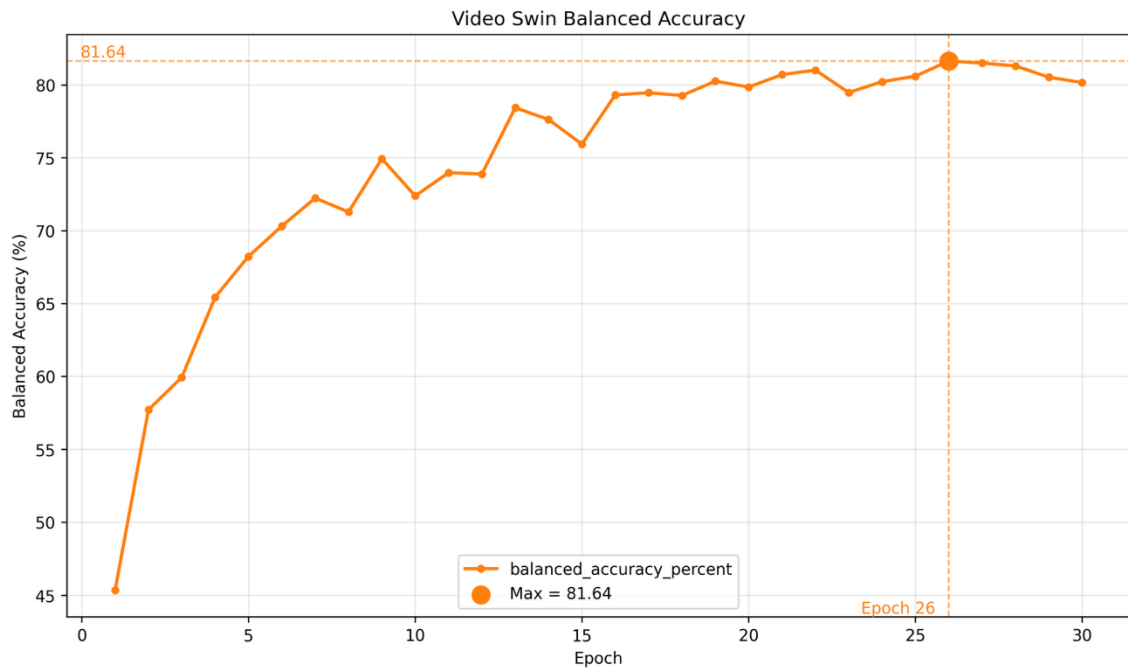


Figure 4.6: Video Swin Transformer Balanced Accuracy curve over epochs

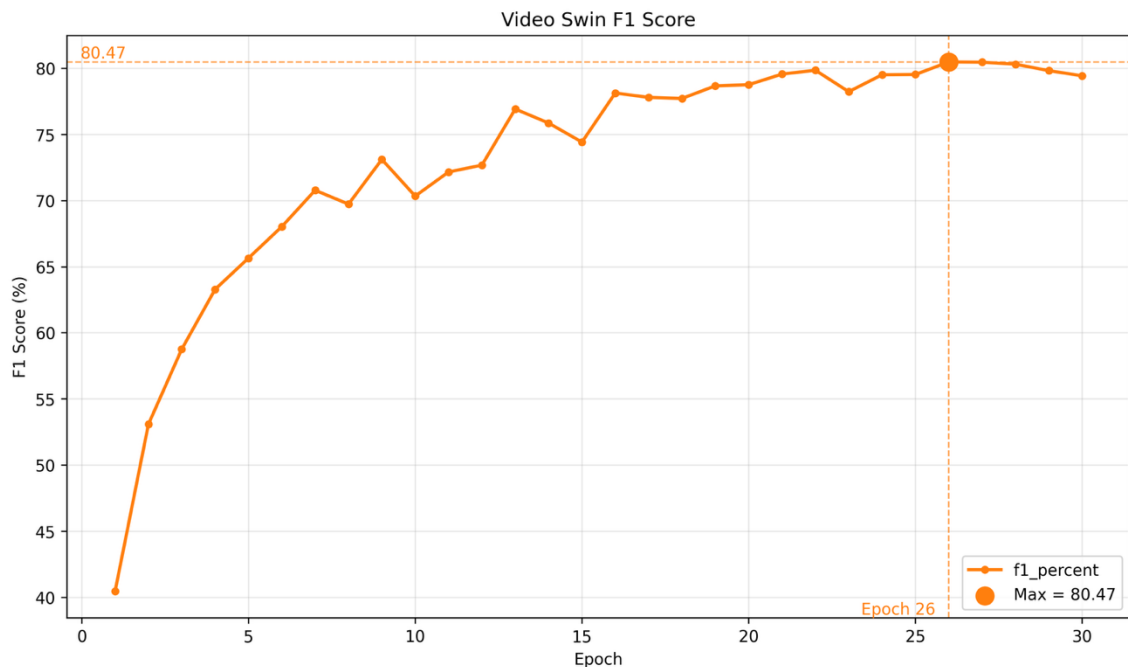


Figure 4.7: Video Swin Transformer F1 score curve over epochs

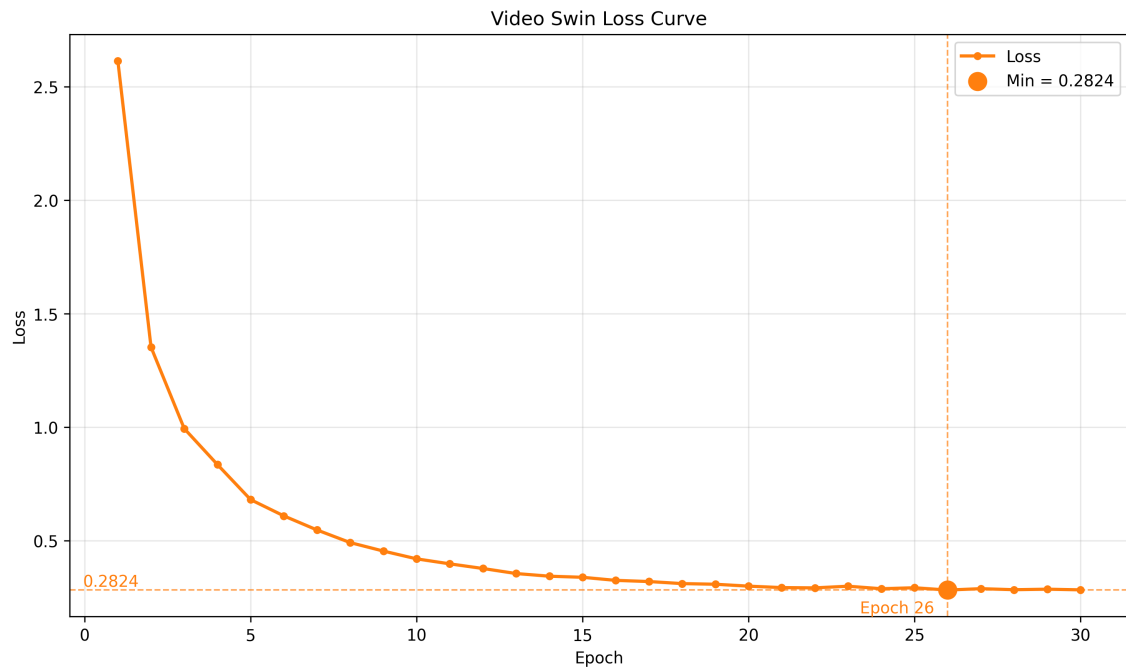


Figure 4.8: Video Swin Transformer Loss curve over epochs

4.6.1.4 Comparison of Unimodal Baselines

CTR-GCN significantly surpassed the Video Swin in all the mentioned metrics, showing improvements of +5.32%, +4.74% and +5.71% in terms of balanced accuracy, overall accuracy and macro-F1, respectively. The notable difference emphasizes the effectiveness of 3D world coordinate skeletons for recognizing yoga postures since skeletons are robust to visual distractions and represent the core kinematics which distinguish between poses. The video input itself learns additional patterns but is prone to overfitting due to fewer visual variations in the data samples. These findings align with previous works on skeleton-based action recognition, which often outperform appearance-based models when the action is defined by relative limb positions [6], [18].

4.6.2 Multimodal Fusion Performance

The logit-level cross-attention fusion model was trained for 30 epochs with frozen backbones. The training dynamics, demonstrate rapid convergence: balanced accuracy reached 88.29 % by epoch 5 and peaked at 89.64 % at epoch 19, after which performance stabilised. The final overall accuracy was 91.06 % and macro-F1 was 88.96 %. Figures 4.9 through 4.12 shows the trend of the metrics over training epochs.

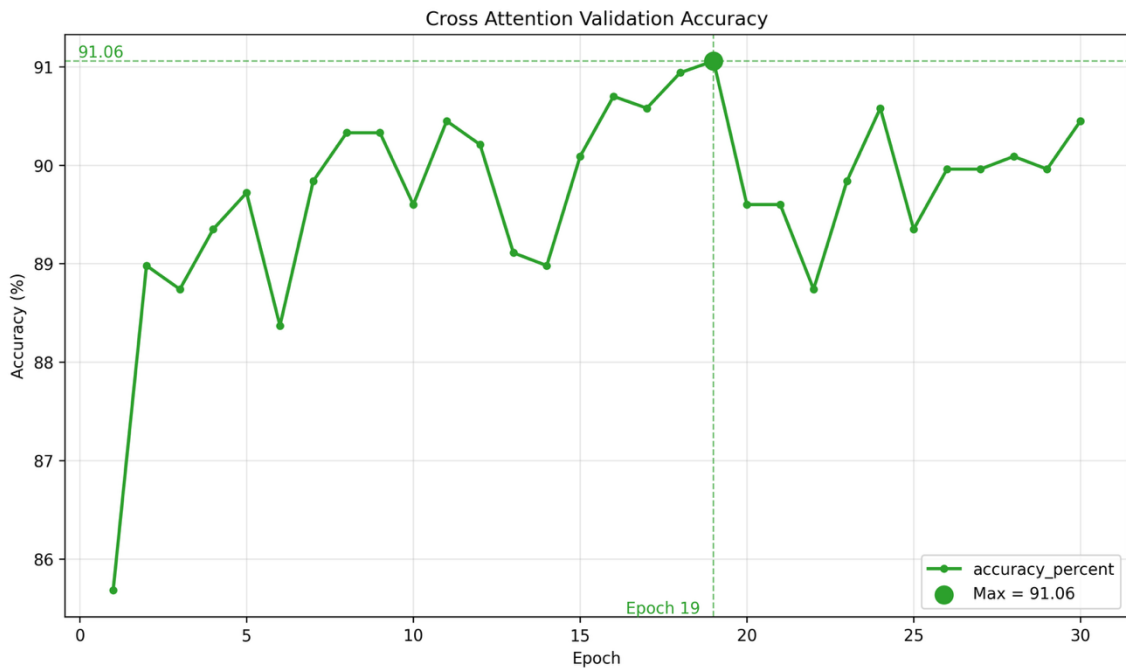


Figure 4.9: Cross-attention fusion model Validation Accuracy curve over epochs

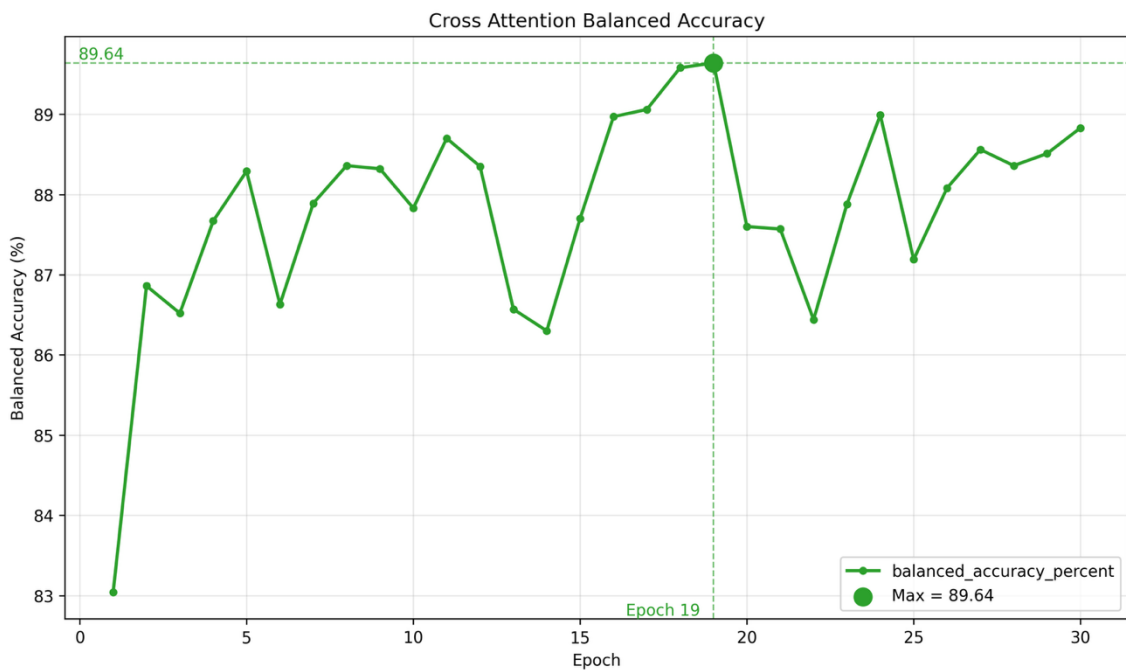


Figure 4.10: Cross-attention fusion model Balanced Accuracy curve over epochs

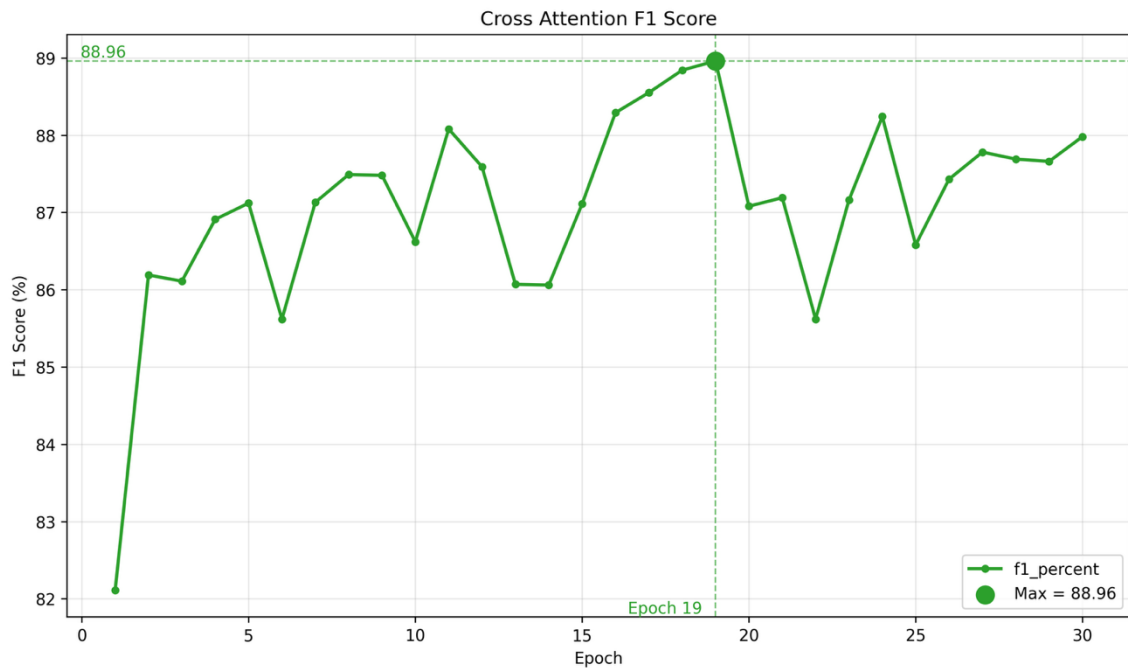


Figure 4.11: Cross-attention fusion model Validation F1 Score curve over epochs

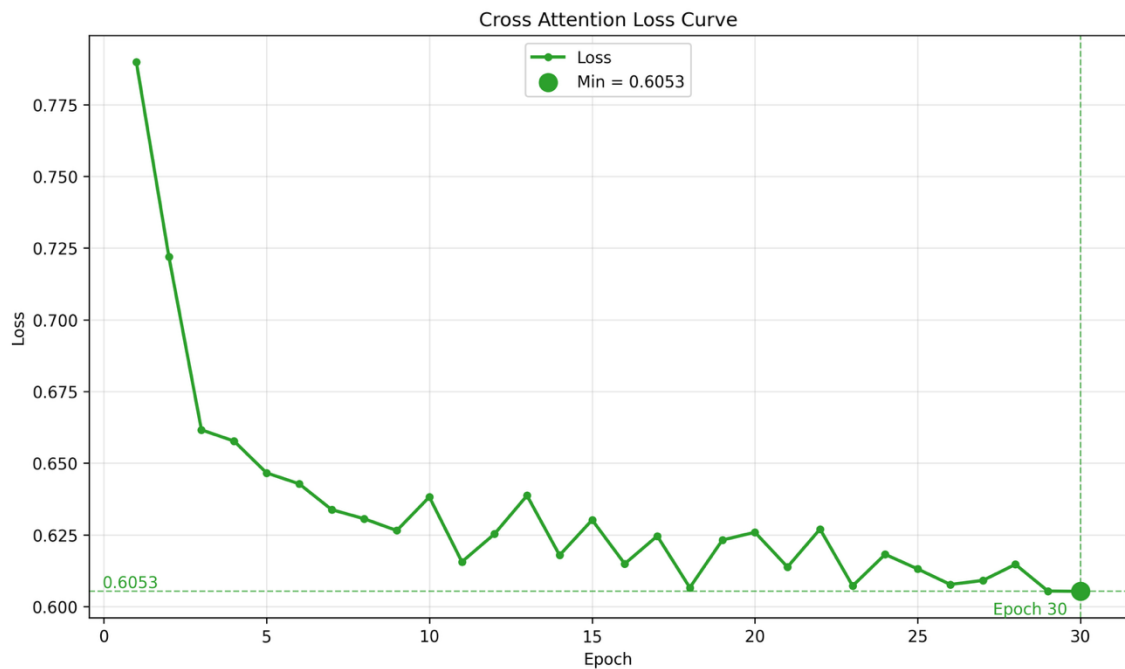


Figure 4.12: Cross-attention fusion model Loss curve over epochs

The fusion model yields a notable improvement over both unimodal baselines:

- **Over CTR-GCN:** +2.68 percentage points in balanced accuracy (86.96 % → 89.64 %), +2.35 in overall accuracy, and +2.78 in macro-F1.
- **Over Video Swin:** +8.00 percentage points in balanced accuracy, +7.09 in overall accuracy, and +8.49 in macro-F1.

These gains are statistically meaningful; the fusion head, which contains fewer than 2 M trainable parameters, learns to combine the complementary strengths of the two modalities without any fine-tuning of the encoders. This shows the usefulness of the video stream when dealing with hard-to-classify instances, since the new error rate is lower than that achieved with a strong skeleton alone.

4.7 DISCUSSIONS ON THE RESULTS

4.7.1 Analysis of Fusion Gain

This is because cross attention happens at the logit stage with skeleton logits being treated as base logits while applying a learned gated delta from both logits. In the gating scalar, whose initial value is zero, the algorithm can always fall back to using skeleton predictions when the visual input is misleading. The fact that improvements were made despite the skeleton predictor being good in its own right indicates that the visual input is instrumental in resolving any ambiguity in poses that look visually similar but kinematically different (such as Warrior I and Warrior II) [27]. The learned cross-attention weights may attend more heavily to the visual logits for pose classes where the skeleton model is known to be less reliable, a behaviour consistent with mixture-of-experts fusion [9].

The relatively modest absolute gain over the skeleton baseline (+2.68 % balanced accuracy) is in line with findings in other multimodal fusion works, where late fusion of a dominant modality with a weaker one yields diminishing returns as the dominant modality approaches saturation [8]. Nevertheless, the consistent improvement across all metrics accuracy, balanced accuracy, and macro-F1 demonstrates that the fusion head generalises well and does not simply memorise the training label distribution.

4.7.2 Convergence and Generalisation

Training curves of the unimodal baselines reveal no significant overfitting: validation metrics improved monotonically or with minor oscillations, and the best models were selected near the end of the training budget. The CTR-GCN's loss continued to decrease past epoch 50, suggesting that a longer training may yield further marginal gains, though at the risk of overfitting.

In the case of the fusion model, a combination of the small head, label smoothing with a value of 0.05, and weight decay at 5×10^{-3} helped mitigate overfitting, given the training dataset size. The metrics showed fluctuations between narrow values post-epoch 10, suggesting the attainment of the optimum state. An early stop patience of 20 epochs was sufficient, given that the optimum model was found before 30 epochs.

4.7.3 Effectiveness of Cross-Modal Fusion

The findings support the main hypothesis of this study, i.e., that combining different modalities at the decision level will lead to improvements in recognizing yoga poses in an unsupervised manner. The skeleton modality, which is implemented on top of the CTR GCN framework, using landmarks in exact world coordinates, is a strong recognizer for its geometric nature. In contrast, the second modality (the video modality), even though less powerful individually, contains discriminative features that can be leveraged using the cross-attention model.

There are several benefits that the proposed approach to fusion enjoys over other methods of feature combination, like simple concatenation or averaging logits:

- It imposes minimal computational overhead
- It allows the use of frozen, arbitrarily complex encoders, and
- The gated residual helps avoid the problem of degradation due to the presence of visual noises thanks to the fallback to the skeleton prediction.

This makes the model ideal for resource-constrained systems, where training a big video transformer end-to-end is not possible. The comparisons to the baseline models are shown in Table 4.5.

Table 4.5: Comparing Multimodal fusion with the baseline models

Model	Accuracy	Balanced Accuracy	F1 Score
CTR-GCN (Skeleton Only) [6]	88.71%	86.96%	86.18%
VideoSwin Transformer (Vision-Only) [23]	83.97%	81.64%	80.52%
Multimodal fusion via Logit-level Cross-Attention	91.06%	88.64%	88.96%

It gives us a very solid baseline to compare future experiments with on the 3DYoga90 dataset and proves that the idea of logit-level fusion can be successfully applied for multimodal action recognition based on skeleton appearance.

CHAPTER 5

CONCLUSION

5.1 APPLICATIONS AND SOCIETAL IMPACT

The multimodal approach that incorporates both 3D skeletal geometry and vision-based techniques of interpreting video provides a highly accurate subject-independent yoga pose classification applicable in practice outside of laboratory settings. The first practical application area is related to the intelligent tutoring system for self-training that utilizes this method as the means of recognizing the pose made by the user and offering feedback regarding its correctness even in a badly lit and cluttered home environment. The two methods used provide a backup in case the second one fails because of ambiguous data, making the pose recognition independent of both body type and outfit worn. Furthermore, this technology can be implemented in physical therapy, where the patient's exercises will be analyzed and exact angle measurements provided from the skeleton data in world coordinates. The approach can thus be considered an improvement upon depth camera-based systems by using monocular video.

Not only in the medical field but also in areas such as sports and dancing, the method works effectively, because it differentiates among visually identical poses in activities like martial arts, gymnastics, and figure skating by virtue of the orientation of the body parts. Pretrained backbones work well when transferred to another activity, thus providing one single platform to study human movements. With regard to human computer interactions and VR, the method enables full body gesture recognition using only one camera, creating an immersive and sensorless interface that can be configured through the light-weighted modular fusion head. As for ergonomics, the skeleton stream provides the joint measurements, while the video stream captures the environment.

From the deployment standpoint, the heavy backbone architecture can be pre-installed on the edge devices, and the small-sized fusion head containing only 1.8 M parameters will introduce minimal latency, which allows for processing on-device and resolves privacy issues. It opens up the opportunity for immediate feedback and makes an otherwise static classifier into a useful tool. From the social viewpoint, the technology can help provide yoga education to everyone, yet additional efforts are needed to include more variety of people into the data to make the algorithm inclusive of everybody.

5.2 FUTURE SCOPE

The future research direction could be the development of the presented architecture to include more complex fusion at early stages of processing, where intermediate features could be combined both hierarchically and through the joint consideration of visual and skeletal backbones' outputs. Memory-saving methods such as adapter layers or prompt tuning could enable the joint fine-tuning of the proposed network without catastrophic forgetting. Currently, the model processes pre-segmented poses.

However, it could be made operational on continuous video streams through temporal action localisation by means of sliding windows, boundary-aware detection or even recurrent graph convolution applied to skeletons.

In terms of increasing the size and diversity of the used 3DYoga90 dataset, the creation of additional data by the help of controllable generation models based on diffusion could be considered. Also, the generation of skeletal sequences with physics-based simulators could diversify the dataset by including various body shapes and styles of exercise. Moreover, the application of self-supervised or multi-task pre-training on the large amount of unlabelled data followed by multimodal pre-training of the unified latent space could increase the quality of the model representation. Finally, personalising the fusion head with a few-shot learning could make the system an interactive learning assistant.

Execution on low-resource edge devices will require the model to be compressed using knowledge distillation, quantization, and pruning, allowing one distilled network to perform both backbones' functions in an attempt to reduce unnecessary computing costs. User trust is a major issue and therefore explainability will be required; in this case, using visualization tools to identify key body parts whose movements affect performance may provide a more detailed analysis of actions than just using the readily understandable cross-attention weights. This technique is agnostic to the specific domain of application, and as such, all one needs to do is change the underlying dataset and skeleton structure in order to apply it to any domain of interest, be it dance, martial arts, rehabilitation movements, sign language, and other areas of interest. Lastly, proactive bias auditing, diverse dataset creation, fairness-aware training, and private data collection methods, including federated learning and on-device processing, need to be explored.

5.3 CONCLUDING REMARKS

The contribution is an end-to-end pipeline for recognizing yoga poses using single-camera videos based on our proposed novel approach for logit-level cross-attention fusion of frozen skeleton and video backbone networks. In terms of contribution, the thesis introduces an end-to-end multimodal solution for the 3DYoga90 dataset, provides strong baselines in each unimodal task using the best unimodal approaches (CTR-GCN 86.96% balanced accuracy and Video Swin Transformer 81.64%), as well as a lightweight LogitPerceiver head for performing cross-attention fusion. Specifically, our fusion head utilizes a learnable query to perform cross-attention to the two expert logits and uses a gated residual correction to get the final predictions resulting in a balanced accuracy score of 89.64% without any finetuning of the backbone networks.

Notable results include the identification of skeletal geometry as the main contributor to yoga pose classification, although visual input offers important corrections and improves the accuracy by 2.68 percentage points when fused. Gated residual learning, initialised to zero, allows the classifier to rely on the skeleton expert whenever visual cues are ambiguous, an invaluable feature in mission-critical systems. It is also important to

mention the use of cross-subject split at source level, which allows separating whole instructors to the test dataset to evaluate realistic generalisation and demonstrate that the model can be applied independently of the instructor.

Beyond the practice of yoga postures, the applications of these systems extend to wherever multiple experts currently exist, ranging from sports science analysis to physiotherapy, dance notation, and sign language recognition, among other fields. If handled properly, taking into account questions of bias, there is significant potential for democratizing physical training and action recognition without any specific domain expertise.

Kinjal Raykarmakar

A Dual-Stream Cross-Attention Framework for fine-grained Yoga Pose classification using Spatiotemporal visual cues and...

 Research Papers

Document Details

Submission ID

trn:oid:::27535:140533100

Submission Date

May 26, 2026, 6:16 PM GMT+5:30

Download Date

May 26, 2026, 6:23 PM GMT+5:30

File Name

24ISY19_Kinjal Raykarmakar_TurnItInSubmission.pdf

File Size

3.7 MB

59 Pages

16,169 Words

97,428 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

A DUAL-STREAM CROSS-ATTENTION FRAMEWORK FOR FINE-GRAINED YOGA POSE CLASSIFICATION USING SPATIOTEMPORAL VISUAL CUES AND 3D SKELETAL GEOMETRY

Kinjal Raykarmakar

ABSTRACT

The accurate automated recognition of yoga postures from a single video sequence presents itself as an intricate problem, lying in the intersection between computer vision and biomechanics, and holds enormous potential towards achieving superior fitness training, physical therapy, and man-machine interaction applications. Existing techniques are mainly based on either appearance features, susceptible to changes in clothing and background, or geometric keypoint features that lack contextual information. The main focus of this thesis is the proposition that a combination of the two modalities can outperform each of them independently.

We introduce a complete pipeline starting from the pre-processing of untrimmed online videos of the 3DYoga90 dataset and extracting the 3D world-coordinate skeletons with the use of MediaPipe. Two novel backbones are fine-tuned on an extremely strict cross-subject split as the unimodal baselines – the first one is Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN) on the skeleton modality, and the other is Video Swin Transformer. We propose a logit-level cross-modal fusion model, termed *LogitPerceiver*, which merges the predictions without requiring full-fledged end-to-end training due to its high computational cost. In addition, the model projects the backbone's logits into a unified space and employs a trainable query to concentrate on two input streams as well as a gated adjustment to the skeleton prediction results.

The late-fusion framework enables adaptive and context-dependent fusion of information related to the input samples from both visual and geometric modalities. Experiments carried out using the 3DYoga90 dataset reveal that the skeleton-based CTR-GCN can achieve a balanced accuracy of 86.96% compared to 81.64% by the Video Swin Transformer. The proposed fusion framework is able to score 89.64% balanced accuracy, 91.06% overall accuracy, and 88.96% macro-averaged F1-score, beating the performance of all unimodal baselines and establishing a new baseline on the dataset. This thesis provides a reproducible multimodal framework, a light fusion network compatible with frozen experts, and a strong benchmark dataset for yoga position classification.

CONTENTS

CANDIDATE’S DECLARATION.....	II
ACKNOWLEDGEMENT.....	III
CERTIFICATE BY THE SUPERVISOR.....	IV
ABSTRACT.....	V
CONTENTS.....	VI
LIST OF FIGURES.....	X
LIST OF TABLES.....	XI
LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE.....	XII
CHAPTER 1 INTRODUCTION	1
1.1 MOTIVATION AND CONTEXT	1
1.2 CHALLENGES IN YOGA POSE RECOGNITION	1
1.3 MULTIMODAL FUSION FOR ROBUST RECOGNITION	2
1.4 PROPOSED APPROACH AND CONTRIBUTIONS	2
CHAPTER 2 LITERATURE REVIEW	3
2.1 BACKGROUND AND SCOPE	3
2.2 DATASETS FOR YOGA POSE ANALYSIS.....	3
2.2.1 Static 2D Datasets: Yoga-82 and Its Variants	3
2.2.2 3D and Video Datasets: 3DYoga90 and 3D-Yoga	4
2.3 SKELETON-BASED ACTION RECOGNITION: GCN	4
2.3.1 From CNNs/RNNs to Graph Convolutions	4
2.3.2 Adaptive and Channel-wise Topology Refinement	5
2.4 VIDEO-BASED ACTION RECOGNITION: VISION TRANSFORMERS	5
2.4.1 3D Convolutional Approaches and Their Limitations	5
2.4.2 Video Transformers: The Swin Paradigm.....	5
2.5 MULTIMODAL FUSION FOR ACTION RECOGNITION	6
2.5.1 The Complementary Nature of Appearance and Skeleton.....	6
2.5.2 Fusion Taxonomies: Early, Intermediate and Late Fusion.....	6
2.6 RESEARCH GAP.....	7
CHAPTER 3 PROPOSED METHODOLOGY	8
3.1 DATASET	8
3.1.1 Primary Research Dataset: 3DYoga90.....	8
3.1.1.1 Multi-Modal Spatio-Temporal Sequences	8

3.1.1.2 Expanded Hierarchical Taxonomy and Expert Curation.....	8
3.1.1.3 Dataset Statistics and Characteristics.....	9
3.1.2 Justification for Modality Transition	11
3.2 SPATIO-TEMPORAL SKELETON EXTRACTION	11
3.2.1 Keypoint Extraction and Representation	11
3.2.2 Processing Video Frames: Temporal Consistency and World Coordinates	11
3.2.3 Data Structure of Extracted Skeletons	12
3.3 VIDEO DATASET FOR ISOLATED ACTION RECOGNITION.....	13
3.3.1 The 3DYoga90 Dataset and Its Temporal Annotations.....	13
3.3.2 Frame-Accurate Temporal Clipping	13
3.4 DATA AUGMENTATION STRATEGY	14
3.4.1 Augmentation of Spatiotemporal Visual Representations	14
3.4.1.1 Horizontal Flipping	14
3.4.1.2 Random Spatial Translation	15
3.4.2 Augmentation of 3D Skeletal Sequences.....	16
3.4.2.1 Coordinate Noise Injection	16
3.5 THE DUAL-MODAL DATASET	17
3.5.1 Modality-Specific Preprocessing and Representations	17
3.5.2 Constructing a Paired Multimodal Instance	17
3.6 CLASS REBALANCING STRATEGIES	18
3.6.2 The Class Imbalance Problem in Supervised Learning.....	19
3.6.3 Loss-Based Rebalancing: Inverse-Frequency Class Weights	19
3.6.4 Sampling-Based Rebalancing: Weighted Random Sampling.....	19
3.7 TRAIN-TEST SPLITTING: THE CROSS-SUBJECT PROTOCOL	19
3.7.1 The Perils of Random Instance-Level Splitting.....	19
3.7.2 Source-Based Splitting as a Proxy for Subject Independence	20
3.7.3 Statistical Rationale and Validation	20
3.8 EVALUATION METRICS	21
3.8.1 Overall Accuracy and Its Shortcomings	21
3.8.2 Balanced Accuracy: Equalising Class Influence	22
3.8.3 Macro-Averaged F1 Score: Harmonising Precision and Recall	22
3.8.4 Combined Interpretation and Practical Use.....	23
3.9 CTR-GCN ARCHITECTURE: SKELETON-ONLY BASELINE	23
3.9.1 Overall Architecture and Data Adaptation.....	23
3.9.2 Role as a Baseline and Foundation for Multimodal Fusion	24
3.10 VIDEO SWIN TRANSFORMER: APPEARANCE-BASED BASELINE	25
3.10.1 Transfer Learning from Kinetics-400	25
3.10.2 Role as a Vision-Only Baseline and Future Fusion	26

3.11 MULTIMODAL FUSION VIA LOGIT-LEVEL CROSS-ATTENTION	26
3.11.1 Design Rationale: Frozen Experts and Late Fusion.....	27
3.11.2 The <i>LogitPerceiver</i> : Cross-Attention over Class Predictions	27
3.11.2.1 Tokenisation of Logits.....	27
3.11.2.2 Learnable Query and Cross-Attention.....	28
3.11.2.3 Feed-Forward Refinement.....	29
3.11.2.4 Residual Delta Prediction and Gating	30
3.11.3 Significance as the Final Model.....	32
CHAPTER 4 EXPERIMENTAL SETUP AND RESULTS	33
4.1 HARDWARE AND IMPLEMENTATION DETAILS.....	33
4.1.1 Software Environment and Library Stack	33
4.1.2 GPU Configuration and Mixed-Precision Training.....	34
4.2 COMMON TRAINING PROTOCOLS	34
4.2.1 Optimisation Algorithm: AdamW.....	34
4.2.2 Learning Rate Schedule: Cosine Annealing	34
4.2.3 Early Stopping and Model Selection	35
4.3 TRAINING PROTOCOL FOR SKELETON-ONLY CTR-GCN	35
4.3.1 Data Sampling Strategy	35
4.3.2 Summary of Hyperparameters	36
4.4 TRAINING PROTOCOL FOR VIDEO SWIN TRANSFORMER.....	36
4.4.1 Transfer Learning from Kinetics-400	36
4.4.2 Summary of Training Hyperparameters	37
4.5 TRAINING PROTOCOL FOR MULTIMODAL FUSION MODEL	37
4.5.1 Meta-Learning over Frozen Backbones.....	38
4.5.2 Learning Rate Schedule with Warm-Up	38
4.5.3 Summary of Hyperparameters	38
4.6 EVALUATION AND RESULTS	39
4.6.1 Unimodal Baseline Performance	39
4.6.1.2 CTR-GCN (Skeleton-Only)	39
4.6.1.3 Video Swin Transformer (Vision-Only).....	42
4.6.1.4 Comparison of Unimodal Baselines.....	44
4.6.2 Multimodal Fusion Performance	44
4.7 DISCUSSIONS ON THE RESULTS	47
4.7.1 Analysis of Fusion Gain.....	47
4.7.2 Convergence and Generalisation	47
4.7.3 Effectiveness of Cross-Modal Fusion.....	48
CHAPTER 5 CONCLUSION	49
5.1 APPLICATIONS AND SOCIETAL IMPACT.....	49
5.2 FUTURE SCOPE	49

5.3 CONCLUDING REMARKS.....	50
CHAPTER 6 REFERENCES	52

LIST OF FIGURES

Figure 3.1: 3DYoga90 Dataset Hierarchy [4]	8
Figure 3.2: Bar chart showing the top 10 Yoga Poses.....	10
Figure 3.3: Temporal Skeleton Extraction	11
Figure 3.4: Tensor Data Structure of extracted skeletons	12
Figure 3.5: Clipping the video files	14
Figure 3.6: Horizontal Flipping of Videos	15
Figure 3.7: Random Spatial Translation of videos.....	15
Figure 3.8: Dual Multi-modal Dataset formation	18
Figure 3.9: Log-Scaled Long-Tail Distribution of Yoga Poses showing significant class imbalance	18
Figure 3.10: Sankey Diagram showing Cross Subject Dataset Split	21
Figure 3.11: CTR-GCN: Overall Architecture	24
Figure 3.12: Video Swin Transformer: Updated classification head to support Yoga Pose Classification.....	25
Figure 3.13: LogitPerceiver: Tokenisation of Logits	28
Figure 3.14: LogitPerceiver: Learnable Query and Multi-head Cross-Attention	29
Figure 3.15: LogitPerceiver: Feed-Forward Refinement	30
Figure 3.16: LogitPerceiver: Building the hidden representation from Residual	30
Figure 3.17: LogitPerceiver: Delta Projection	31
Figure 3.18: LogitPerceiver: Gating and Fused Logits.....	31
Figure 4.1: CTR-GCN Validation Accuracy curve over epochs	40
Figure 4.2: CTR-GCN Balanced Accuracy curve over epochs.....	40
Figure 4.3: CTR-GCN F1 score curve over epochs.....	41
Figure 4.4: CTR-GCN Loss curve over epochs	41
Figure 4.5: Video Swin Transformer Validation Accuracy curve over epochs	42
Figure 4.6: Video Swin Transformer Balanced Accuracy curve over epochs.....	43
Figure 4.7: Video Swin Transformer F1 score curve over epochs	43
Figure 4.8: Video Swin Transformer Loss curve over epochs	44
Figure 4.9: Cross-attention fusion model Validation Accuracy curve over epochs	45
Figure 4.10: Cross-attention fusion model Balanced Accuracy curve over epochs.....	45
Figure 4.11: Cross-attention fusion model Validation F1 Score curve over epochs	46
Figure 4.12: Cross-attention fusion model Loss curve over epochs	46

LIST OF TABLES

Table 3.1: Dataset Statistics of 3DYoga90	9
Table 4.1: Training Hyperparameters for CTR-GCN.....	36
Table 4.2: Training Hyperparameters for Video Swin Transformer.....	37
Table 4.3: Training Hyperparameters for Logit-Level Cross-Attention Fusion.....	38
Table 4.4: Comparison of Unimodal Baseline Performances	39
Table 4.5: Comparing Multimodal fusion with the baseline models	48

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

T	Total number of frames or temporal dimension
C	Number of color channels (e.g., $C = 3$ for RGB) or the number of classes
H, W	Spatial dimensions of an image tensor (Height and Width)
V	Number of MediaPipe joints/landmarks (Vertices)
X, Y, Z	Spatial coordinates in meters (world coordinates)
x, y, z	Normalized spatial/depth coordinates in the image plane
TP_c	True Positives for a specific class c
FN_c	False Negatives for a specific class c
FP_c	False Positives for a specific class c
η	Learning rate
l_{vis}	Logit vector produced by the vision backbone
l_{geo}	Logit vector produced by the geometry (skeleton) backbone
l_{fused}	Fused logit vector
Q	Learnable query vector used in the cross-attention module
δ	Delta logit vector for residual correction applied to the skeleton logits
α	A learnable scalar gate used to control the residual connection
AGCN	Adaptive Graph Convolutional Network
AMP	Automatic Mixed Precision
CNN	Convolutional Neural Network
CTR-GCN	Channel-wise Topology Refinement Graph Convolutional Network
CUDA	Compute Unified Device Architecture
FFN	Feed-Forward Network
GCN	Graph Convolutional Neural Network
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
NaN	Not a Number
RGB-D	Red, Green, Blue – Depth
ST-GCN	Spatial-Temporal Graph Convolutional Network
ViT	Vision Transformer

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION AND CONTEXT

Yoga has become extremely popular in recent years, and its popularity has not gone unnoticed for various reasons, including the physical benefits it brings, relaxation, and even mental improvement [1], [2]. While many people practice yoga exercises in their homes today, they lack the necessary guidance and assistance to do it properly. Hence, technological intervention is needed to ensure that systems are developed capable of detecting the user's errors and providing tailored instructions and recommendations, regardless of whether a personal instructor will be present [3]. Automatic detection of yoga postures on video can prove invaluable for a variety of applications, from health-related to gaming.

Considering the complexity involved when distinguishing yoga postures from the computer vision perspective, the task becomes quite challenging. The reason for this is that unlike dynamic activities such as running or jumping, yoga poses remain stationary; hence the difference lies in very subtle details such as joints angle or limbs positions [4]. Even the similarity among some poses, such as Warrior One and Warrior Two poses makes the classification difficult to perform. Furthermore, each individual varies physically, clothing style and camera angles can also contribute to the distortion of images making them look differently. To enable recognition of up to 90 yoga poses in a database such as the 3DYoga90 database, the proposed models must possess certain characteristics, including being able to recognize subtle details while ignoring non-relevant factors like lighting and attire.

1.2 CHALLENGES IN YOGA POSE RECOGNITION

Recognizing yoga poses automatically has been found to be difficult due to several reasons. The first challenge is related to the appearance variations, in which case the same pose may have different appearance characteristics based on factors such as attire, camera viewpoint, background, and lighting conditions. For instance, a machine learning algorithm trained on yoga videos shot in a well-lighted studio may perform poorly when employed for home use as a result of the varying environmental background in such a scenario [5]. Second is the existence of poses that are very close to each other whereby the distinguishing characteristic may only be the posture of certain body joints and hands or feet positions. In order to distinguish such poses, it is important to design an algorithm that captures the fine-grained appearance features [6]. Thirdly, the algorithm is supposed to detect poses regardless of the pose executor, meaning that it cannot base recognition on personal attributes such as facial characteristics, physical traits, and gait analysis. Evaluating such algorithms can prove to be difficult as they have to be generalized to work across multiple people [7].

1.3 MULTIMODAL FUSION FOR ROBUST RECOGNITION

Multimodal information fusion, i.e., an integration of different types of data such as visual and audio streams, is critical for the comprehension process in video analysis tasks. As previously mentioned, the multimodal information fusion can be performed by three techniques: early fusion of raw inputs, mid-level feature concatenation [8], and late fusion of the output of each modality classifier. Early and mid-level approaches require joint training of each expert that is prohibitively expensive for large pre-trained models. The late approach allows each individual modality model to be trained independently and then fused using the meta-learner [9], [10]. Therefore, it does not introduce any overhead in terms of computational complexity. However, simply averaging or weighing individual scores from modalities does not give any advantage since it cannot capture any intermodal dependencies between inputs. Attention-based approaches have proved their superiority in many computer vision and NLP problems [11]. It seems reasonable to assume that the attention-based mechanism can significantly improve multimodal fusion and lead to more accurate multimodal classification of a sample.

1.4 PROPOSED APPROACH AND CONTRIBUTIONS

This thesis presents a novel multi-modal approach for yoga pose recognition. Our framework integrates the visual appearance as well as the geometry of a yoga pose through a learnable cross-attention mechanism. The purpose of this thesis is to integrate multimodal data into our pose recognition problem to make our framework perform better. Specifically, we use a cross-attention head mechanism that allows us to pay attention to the most important aspects of a pose. Our model consists of three major components, and we will explain each component in detail later.

- Data preprocessing and skeleton extraction
- Unimodal baselines
- Logit-level cross-attention fusion (*LogitPerceiver*)

Key contributions made by this thesis may be outlined as follows:

- A complete end-to-end pipeline for multimodal yoga pose recognition from raw videos to classification, specific to 3DYoga90.
- The development of highly competitive unimodal baselines based on a stringent cross-subject procedure for benchmarking future methods.
- The introduction of a novel, efficient logit-level fusion network capable of combining pre-trained experts without joint fine-tuning.
- Experimental results proving that the proposed multimodal fusion approach provides 89.64% of balanced accuracy, which exceeds that of unimodal streams and is state-of-the-art in terms of 3DYoga90.

CHAPTER 2

LITERATURE REVIEW

2.1 BACKGROUND AND SCOPE

Action recognition is an approach towards automatically recognizing different activities performed by humans using visual information. Applications of action recognition include but are not limited to surveillance systems, human-computer interactions, medicine and sports [12]. Fine-grained action classification involves recognizing actions that have similar body configurations in gross but slightly varied joint alignments. The problem of fine-grained action recognition is further complicated in applications like yoga pose recognition where it is necessary for a model to not only learn global postural semantics but also the finer kinematics involved.

2.2 DATASETS FOR YOGA POSE ANALYSIS

Diverse, well-documented datasets form an essential pre-requisite for making advancements in pose classification using deep-learning techniques [4], [13]. Initial research works were based on static two-dimensional sets of poses of yoga postures, while later developments included three-dimensional spatiotemporal information.

2.2.1 Static 2D Datasets: Yoga-82 and Its Variants

The most prevalent among static yoga datasets is Yoga-82 dataset developed by Verma et al. [13]. This consists of more than 28,400 Internet-sourced images arranged into 82 different pose categories according to a three-level hierarchy – there are six main root classes (such as standing, sitting, inverted, reclining) followed by intermediary categories that split into specific pose types. The hierarchical structure acts as a remedy for the lack of detailed joint labeling by allowing a course-to-fine classification process. Yoga-82 became a standard dataset for testing CNN yoga classifiers, which showed an increase in accuracy when using VGG-16, ResNet-50, and DenseNet-121 models [3], [14].

Further developments include Bera et al.'s work on the SYD (Sports, Yoga, Dance) benchmark with Yoga-82 along with a larger Yoga-107 dataset, sports, and dance postures, and SYD-Net, a CNN-based model with patch-level attention [14]. Recent research has investigated various other approaches to make yoga poses recognition computationally efficient with CNN structures like YPose and a grading system based on skeletons [3], [15].

Although useful, static 2D benchmarks possess inherent disadvantages as they lack the temporal dimension, contain no depth information, and suffer from viewpoint variability along with background distractions that cause shifts in domain distribution [4], [16]. This limitation drove the trend toward using video data and skeleton-based 3D representations.

2.2.2 3D and Video Datasets: 3DYoga90 and 3D-Yoga

Given the importance of temporal and spatial data, several video-based yoga datasets have been created. 3DYoga90 [4] increases the number of poses to 90 and offers, per pose instance, a well-cut RGB video clip as well as its skeleton sequence obtained from the world coordinate space with the help of the BlazePose approach [17]. In its construction, the dataset used a group of six people including yoga professionals and videos from YouTube obtained through multilingual search (English, Korean, Japanese, Sanskrit). It consists of 6 177 RGB video clips (of which 5 526 skeleton sequences were successfully generated) with an average length of ~12 seconds and the data split based on original video sources. The latter eliminates the bias towards individual instances, which can be problematic when using random splits at the level of individual data samples [5], [7].

In addition, 3D-Yoga by Zhu et al. presents a 3D yoga pose database with 3,792 actions and 16,668 key frames from RGB-D data of 22 people captured by two RGB-D cameras, featuring 117 pose categories. This database includes 3D skeletons reconstructed from multiple views and was evaluated using a two-stream adaptive graph convolutional neural network.

This thesis takes 3DYoga90 as the main empirical basis as this dataset is one of a kind, with its combination of raw RGB video and 3D skeletal landmarks [4] for a systematic investigation into the fusion of multimodal approaches.

2.3 SKELETON-BASED ACTION RECOGNITION: GCN

2.3.1 From CNNs/RNNs to Graph Convolutions

Traditional skeleton-based action recognition approaches have used joint locations as sequential or pseudo-image data and applied RNNs or 1D/2D CNNs [15], [18]. But these approaches do not take into account the underlying graph structure in the human skeleton, in which joints are connected by bones that have different physical and functional significance. Spatial-Temporal Graph Convolutional Networks (ST-GCN), proposed by Yan et al. [18], model the human body's graph structure inside each frame and apply convolutions along the temporal axis, leading to outstanding performance on action recognition datasets like NTU RGB+D [7]. The major contribution of ST-GCN is the introduction of a fixed, manually constructed adjacency matrix based on the connectivity of the skeletal system; this ensured that the network takes the anatomical constraints into account. But this fixed connectivity structure cannot represent the connectivity requirements related to actions and channels: the required connectivity between joints for detecting whether a person is walking or running can be different than for a yoga balance pose [6].

2.3.2 Adaptive and Channel-wise Topology Refinement

In contrast, the following approaches were able to overcome this limitation by proposing learnable adjacency matrices. 2s-AGCN (Shi et al. [19]) proposed a two-stream design that simultaneously learns both the graph structure and the node features, thus dynamically establishing additional relationships within the network beyond the skeletal graph. Although more flexible than ST-GCN, 2s-AGCN also uses one common adjacency matrix across all channels.

To further improve the flexibility, Chen et al. [6] introduced the Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN). By assigning a unique adjacency matrix to each output channel, they managed to increase the discriminative power of the network. Specifically, their approach involves calculating a dynamic similarity matrix between input features by using a cross-channel attention module. This allows adding it into the previously fixed skeletal adjacency matrix to obtain the final model (blocks of spatial CTR-GC interleaved with temporal convolutions), which is capable of modeling action- and channel-specific relationships in joint movements. CTR-GCN achieves better results on standard benchmarks such as NTU RGB+D and Kinetics-Skeleton [6]. Therefore, this thesis will use the CTR-GCN as a backbone of the geometry stream for 33-joint MediaPipe topology obtained from the 3DYoga90 dataset [4], [17].

2.4 VIDEO-BASED ACTION RECOGNITION: VISION TRANSFORMERS

2.4.1 3D Convolutional Approaches and Their Limitations

Prior to the emergence of transformers, video action recognition had relied mostly on 3D convolution networks like C3D [20], I3D [21], and SlowFast [22]. These models have proven to be very successful, but one problem with these 3D CNNs is the presence of localization issues, as the receptive field size increases linearly with increasing network depths [23]. Although two stream approaches where there is a distinct pathway for the spatial and temporal streams [16] can somewhat alleviate this issue, localization still persists in temporal processing.

2.4.2 Video Transformers: The Swin Paradigm

Unlike CNNs that employ convolutions, Vision Transformers (ViTs) use self-attention allowing every token to attend to all the other tokens within the same sequence [24]. Unfortunately, ViTs suffer from quadratic complexity caused by the need to perform global attention. To overcome this limitation, the Swin Transformer [25] proposed using a hierarchical design along with shifted window attention where self-attention is performed within non-overlapping spatial windows and subsequently shifted to facilitate cross-window communication, making it linear in the number of tokens.

Similarly, the Video Swin Transformer [23] extended this paradigm to modeling spatiotemporal data by creating 3D patches from the input clip and then using 3D shifted window attention. Such an approach allows modeling both short-range movement and

long-range dependencies in video while remaining linear. Pre-trained on large-scale datasets, such as Kinetics-400 [26], Video Swin achieved impressive results surpassing state-of-the-art on multiple video benchmarks [23]. Since most yoga postures require modeling the information over a 30-frame window (about 12 seconds), being able to model long-range dependencies becomes critical. In the current thesis, we use Video Swin Transformer (Tiny version) pre-trained on Kinetics-400 as our baseline and frozen expert in multimodal fusion.

2.5 MULTIMODAL FUSION FOR ACTION RECOGNITION

2.5.1 The Complementary Nature of Appearance and Skeleton

No single modality provides complete information for action recognition in unconstrained environments. RGB video captures rich appearance cues (clothing, background, texture) and motion boundaries, but is susceptible to domain shift caused by changes in illumination, viewpoint, and background clutter [16], [21]. Conversely, 3D skeletal data are invariant to such superficial variations and encode the pure kinematic configuration, yet they discard contextual information and depend on the accuracy of the upstream pose estimator [18], [27]. These properties are complementary: a robust system can leverage skeletal geometry as a stable backbone and appearance cues to resolve ambiguities when the skeleton is occluded or misestimated [8], [28].

2.5.2 Fusion Taxonomies: Early, Intermediate and Late Fusion

Multimodal fusion methods can be roughly grouped based on their position within the representation hierarchy [28]. Early fusion is a method that fuses raw inputs prior to modality-specific processing; however, it may not necessarily learn domain-specific representations. Feature-level fusion, also known as intermediate fusion, extracts feature-specific representations for each individual modality first and then concatenates or sums up or performs attention on them [8]. Decision-level fusion combines independent unimodal decisions (logits or probabilities) using average, weighted voting, or a meta-learner [9], [29]. The benefits of decision-level fusion include decoupling the training of each expert and the ability to train frozen pretrained models.

There have been various attempts at attention-based fusion. CMF-Transformer is a cross-modal transformer-based fusion model to fuse spatiotemporal video features and skeleton joint features. TransMODAL is an attention-based fusion mechanism to facilitate iterative cross-modal information sharing between the RGB and pose stream. MAF-Net uses both self-attention and skeletal attention modules under the decision-level fusion framework.

2.6 RESEARCH GAP

Literature review demonstrates the following development trends:

- **Data representation:** The community evolved from 2D static image databases such as Yoga-82 [13] and SYD [14] to 3D video datasets like 3DYoga90 [4] and 3D-Yoga that provide the appearance and skeleton streams respectively.
- **Geometric recognition:** Graph convolution networks progressed from static architecture (ST-GCN [18]) to dynamic graph-based models (2s-AGCN [19], CTR-GCN [6]) with solid geometric priors.
- **Video recognition:** Recent advancements in vision transformer architecture such as Video Swin [23] outperformed CNN architectures on various benchmarks by learning distant spatial-temporal relations.
- **Multimodal processing:** Several works have proposed multimodal solutions based on attention mechanism, which combine RGB and skeleton modality, however, most of them require joint training or fine-tuning of backbones.

However, there is still an important research question to be answered: how can we efficiently integrate well-trained skeletal and video models for fine-grained yoga pose classification without requiring costly end-to-end training again? Fortunately, this problem is precisely what the current thesis tries to solve with a proposed cross-attention strategy at the logit level using a frozen CTR-GCN and Video Swin backbone coupled with a trainable residual correction head. Not only does this method utilize the synergies of both geometry and appearance, but it also follows a subject-independent evaluation by employing a cross-subject split setting, making it state-of-the-art.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 DATASET

3.1.1 Primary Research Dataset: 3DYoga90

3.1.1.1 Multi-Modal Spatio-Temporal Sequences

In order to address the deficiencies associated with static 2D images, the methodology used in this thesis has been modified such that it can utilize the 3DYoga90 dataset [4]. 3DYoga90 is an important contribution to datasets involving human action because it provides multi-modal sets of continuous RGB videos together with their respective 3D skeletons. Since there is temporal information contained within the video, there is an opportunity to model the transitions between different postures. Consequently, there will be the possibility of not only capturing the spatial arrangement of the joints but also capturing kinematic features like joint velocities and accelerations.

3.1.1.2 Expanded Hierarchical Taxonomy and Expert Curation

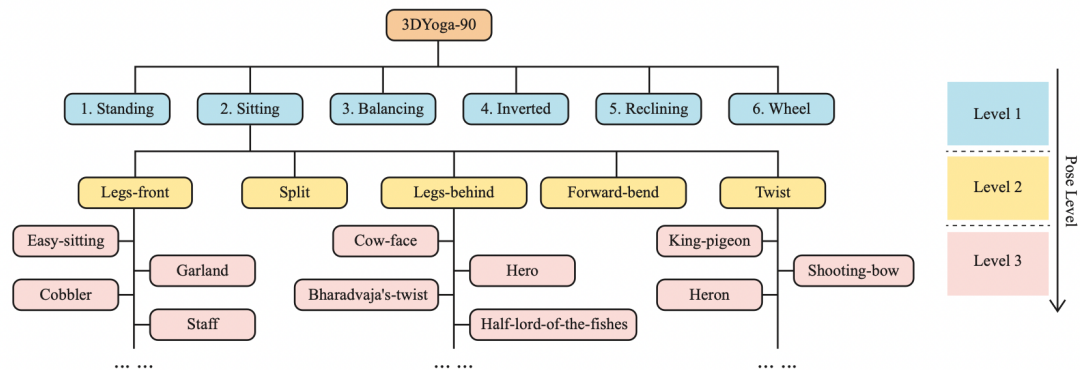


Figure 3.1: 3DYoga90 Dataset Hierarchy [4]

Building on the structure taxonomy created by Yoga-82, 3DYoga90 adopts a highly detailed hierarchical labeling scheme involving three levels of labels, while increasing the number of categories from 82 to 90 [4]. This is due to the greater variety of complex poses, which pose additional challenges to advanced spatio-temporal recognition models. Importantly, unlike web-sourced databases, 3DYoga90 was carefully compiled by professionals in the field, including yoga teachers. This ensures that there are no anatomical inaccuracies in the execution of poses since such a database is needed for scientific purposes. Figure 3.1 shows the dataset hierarchy for the 3DYoga90 dataset.

3.1.1.3 Dataset Statistics and Characteristics

Overall, the total number of curated RGB video samples in 3DYoga90 dataset amounts to 6,177 samples. The base videos have been extracted from 2,170 different channels/creators, thus guaranteeing an extensive variation of environments and users. Table 3.1: describes the statistics of different classes within the dataset.

Table 3.1: Dataset Statistics of 3DYoga90

Yoga Pose	Count	Yoga Pose	Count	Yoga Pose	Count
downward-dog	245	half-moon	48	dolphin	43
standing-forward-bend	226	garland	48	warrior-2	43
half-way-lift	212	crane-crow	48	standing-split	43
mountain	163	extended-puppy	48	cow-face	43
chair	140	side-plank	48	half-lord-of-the-fishes	43
cobra	128	upward-facing-two-foot-staff	48	flying-splits	43
cockerel	120	gate	47	shoulder-stand	43
extended-triangle	107	warrior-3	47	plow	43
extended-side-angle	101	child	47	peacock	43
corpse	95	lizard	47	upward-plank	43
staff	94	side-reclining-leg-lift	47	pigeon	43
wind-relieving	94	wild-things	47	high-lunge	42
fish	93	cat	47	bharadvaja's-twist	42
happy-baby	93	lord-of-dance	46	eight-angle	42
shoulder-pressing	92	shooting-bow	46	handstand	42
reclining-cobbler	91	king-pigeon	46	feather-peacock	42
reclining-hero	82	upward-bow	46	standing-big-toe-hold	41
frog	80	cow	46	heron	41
tree	69	bow	46	side-crane-crow	41
intense-side-stretch	59	eagle	45	bridge	41
scale	59	hero	45	three-legged-dog	40
scorpion	57	firefly	45	tortoise-pose	40
reclining-hand-to-big-toe	57	supine-twist	45	dolphin-plank	40

locust	55	camel	45	boat	40
revolved-head-to-knee	54	warrior-1	44	easy-sitting	39
monkey	53	low-lunge	44	headstand	39
low-plank	51	seated-forward-bend	44	noose	38
reverse-warrior	49	balancing-table	44	wide-legged-forward-bend	36
cobbler	49	plank	44	downward-facing-frog	36
head-to-knee	49	goddess	43	wide-angle-seated-forward-bend	33

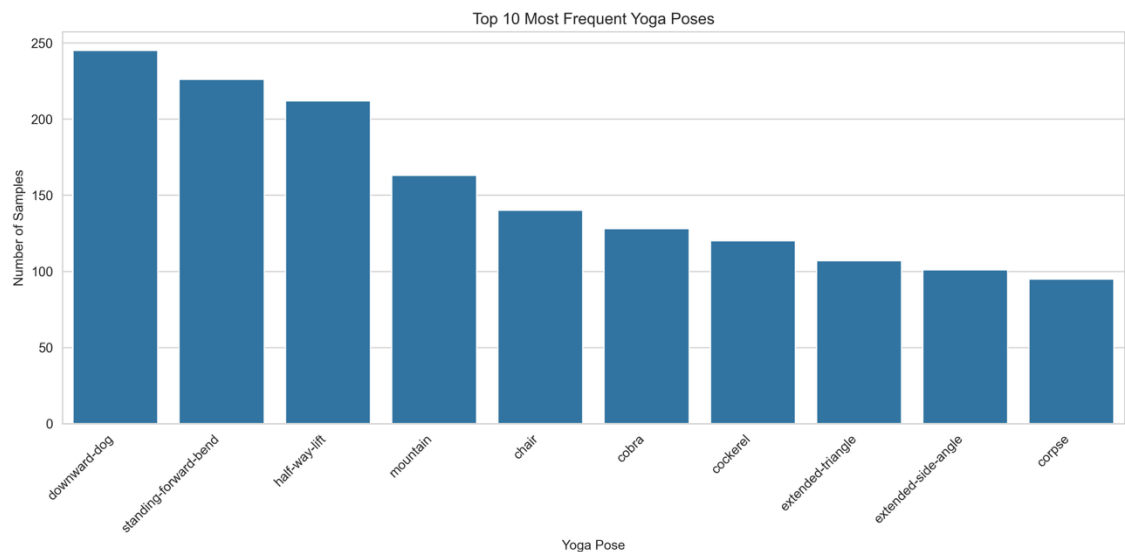


Figure 3.2: Bar chart showing the top 10 Yoga Poses

Figure 3.2 shows the top 10 yoga poses in the dataset.

- **Skeleton Dataset Attrition:** During the stage of skeleton extraction, the cases in which BlazePose did not detect poses in some of the frames led to the attrition of the dataset, and thus the final skeleton dataset consists of 5,526 skeleton data.
- **Distribution and Sequence Length:** In terms of distribution and sequence length at level 3 classification, the pose 'Downward Dog' occurs frequently, with 245 video occurrences. Throughout the dataset, the average number of occurrences per pose is 75. The average length of the extracted sequence is 12.16 seconds, having a minimum of 2.83 seconds (for pose 'Halfway Lift') and maximum of 25.84 seconds (for pose 'Balancing Table').

3.1.2 Justification for Modality Transition

The evolution from the image-based Yoga-82 dataset to the video-based 3DYoga90 dataset illustrates a paradigm shift from the extraction of static features to the analysis of the entire spatio-temporal sequence. The nature of the 3D skeleton sequences in 3DYoga90 allows us to obtain invariant topological representation of the human body, which automatically avoids the issues of viewpoint dependence and background noise encountered in 2D datasets. Through the integration of temporal information by means of video sequence, the current approach can accurately model the transitional phase of motions, providing us with a high-quality representation of human kinematics.

3.2 SPATIO-TEMPORAL SKELETON EXTRACTION

3.2.1 Keypoint Extraction and Representation

BlazePose model, a key component of the Mediapipe pipeline [17], is used to detect keypoints in body poses. BlazePose model is a fast and extremely accurate full body pose detection solution suitable for real-time use. It detects 33 keypoints, each with four properties: normalized (x, y) coordinates in the image plane, relative depth coordinate z , and a confidence score that tells the visibility of the particular point. This representation gives a concise illustration of the skeletal structure without the need to work with complete pictures.

The extracted landmarks provide a structured representation which holds a considerable amount of data about the physical structure of the body. The (x, y) location denotes the relative placement, whereas the depth aspect signifies the motion of the body parts in the direction of the optic axis of the camera. This aids the distinction between yoga poses that may seem similar but differ with respect to front/backward stretch. The visibility score helps the system cope with invisible/ambiguous keypoints [30], [31].

3.2.2 Processing Video Frames: Temporal Consistency and World Coordinates

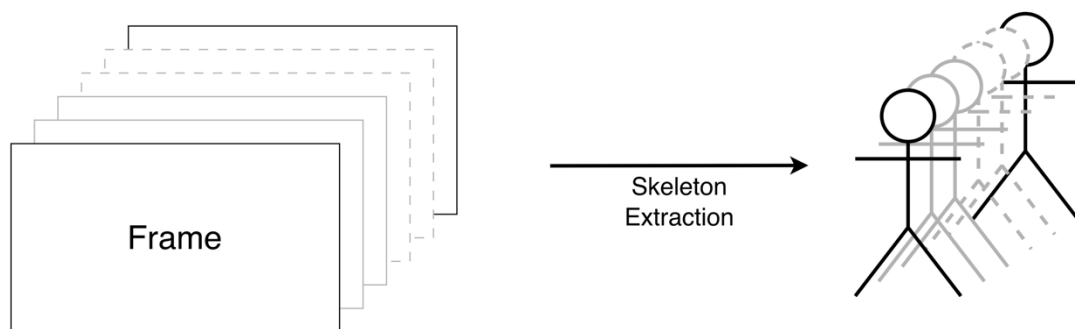


Figure 3.3: Temporal Skeleton Extraction

For each frame at index t , the raw network output comprises:

- **Image landmarks:** $(x_{i,t}, y_{i,t})$ in normalized image coordinates (range $[0, 1]$).
- **World landmarks:** $(X_{i,t}, Y_{i,t}, Z_{i,t})$ in meters, where $i \in \{0, \dots, 32\}$

As the process operates on every frame independently as shown in Figure 3.3 (with an exception of the ROI linking function), there is no built-in temporal filtering or smoothing done automatically. The design of the algorithm is aimed at preserving the fast-changing features inherent to the moving object. This feature is important for gesture recognition and motion capture.

3.2.3 Data Structure of Extracted Skeletons

In the case of video data represented by T frames, the landmarks in the world coordinate system are organized in the form of a three-dimensional array $\mathcal{S} \in \mathbb{R}^{3 \times 33 \times T}$. The axes are defined as follows:

- **Dimension 0 (coordinate):** $0 \rightarrow X, 1 \rightarrow Y, 2 \rightarrow Z$.
- **Dimension 1 (landmark index):** An integer i from 0 to 32, corresponding to the anatomical landmark.
- **Dimension 2 (time):** Frame index t ranging from 0 to $T - 1$.

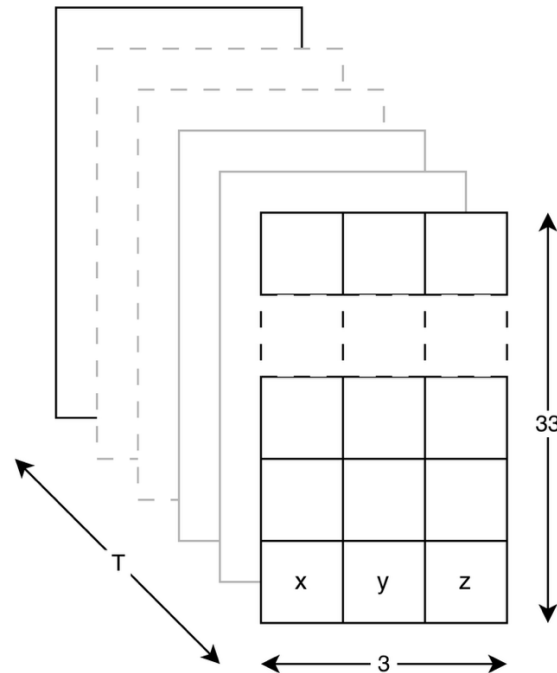


Figure 3.4: Tensor Data Structure of extracted skeletons

Each element $\mathcal{S}_{c,i,t}$ thus stores the c -th coordinate of landmark i at frame t as shown in Figure 3.4. When a landmark is not detected in a given frame (e.g., due to occlusion or low confidence), the corresponding values are set to *NaN*. The array is pre-allocated with NaN entries of the maximal possible length and then trimmed to $T_{\text{actual}} \leq T$ after processing all frames, removing any trailing frames that were never filled because the video ended earlier than estimated or contained fewer readable frames.

So, for each element $\mathcal{S}_{c,i,t}$, the value corresponds to the coordinate c of the landmark i at time frame t . For those landmarks that cannot be detected during certain time frames (for example, due to occlusion or uncertainty), their respective values will be assigned NaN values. The whole array is allocated in advance with the maximum number of NaN elements and then shortened to $T_{\text{actual}} \leq T$ depending on the number of frames processed.

This tensor representation of data allows fast vector processing for further analysis, such as calculating the velocity of joint movement through finite differences on the time axis or isolating a trajectory of a chosen landmark throughout the whole clip. The use of a world coordinate system measured in meters guarantees an unambiguous interpretation of numeric results – for instance, moving +1.0 along the Y-axis means changing altitude by one meter, irrespective of the zoom level of the camera.

3.3 VIDEO DATASET FOR ISOLATED ACTION RECOGNITION

3.3.1 The 3DYoga90 Dataset and Its Temporal Annotations

The 3DYoga90 database [4] offers an extensively annotated set of yoga postures, featuring hundreds of action instances extracted from publicly available videos on the web. In addition to a unique identifier and a URL pointing to the source video, the database offers temporal limits of these action instances in seconds. The limits represent the exact period when the specific yoga posture is executed. Unlike in frame-based annotation, when actions depend on the particular resolution and framerate of the video, timestamp-based annotations make sure that the definition of the action instance is independent of the encoding and frame rate of a particular video. This approach aligns well with benchmark practice [32], [33], as temporal annotations are provided in seconds.

3.3.2 Frame-Accurate Temporal Clipping

Once the right source file has been identified, the start time and duration are used to perform a video clip procedure. The end goal is to generate a new video file starting exactly at the given start time and with the given duration, maintaining the original frame rate and image quality. Fast-forwarding to specific points in time to obtain precise clips is not trivial in the case of compressed video streams, since most codecs use some form of inter-frame prediction, whereby seeking directly to a desired point in time may yield a P- or B-frame that relies on previous keyframes for its decoding. In order to get around this problem, the process entails fast-seeking to the closest keyframe before the intended timestamp, decoding from there, and ignoring all of the frames prior to the precise start

time. The clipped segment is encoded in a standardised format (H.264/MP4), with constant frame rate, as this is necessary for temporal determinism when processing the stream frame by frame [34]. Figure 3.5 shows clipping of the video files.

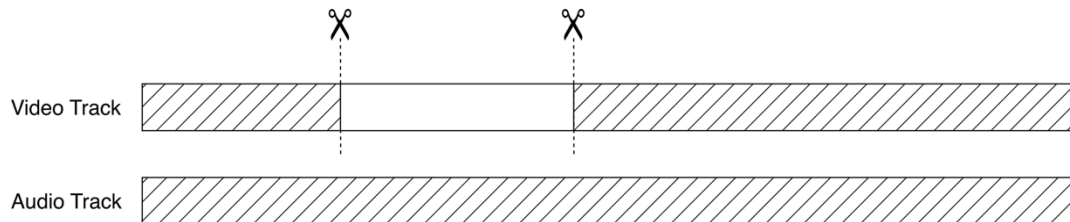


Figure 3.5: Clipping the video files

Opting for an encoder preset that promises speed and efficiency is a conscious decision to sacrifice efficiency for throughput because the target is to have the best possible clip produced in the shortest time possible since the video files are intermediate products.

3.4 DATA AUGMENTATION STRATEGY

The effectiveness of deep neural network models for action recognition is dependent on the variability and quality of the data used for training. In cases where video datasets labeled with relevant annotations are costly and cumbersome to acquire – for example, highly specific physical activities like yoga poses – data augmentation becomes an essential approach for mitigating overfitting and increasing generalization performance [35]. The augmentation framework employed in this thesis is designed for the bi-modal structure of the 3DYoga90 dataset [4] consisting of video tensors based on visual appearances as well as geometric representations of skeletons. Data augmentation procedures are carried out independently for each type of data depending on its properties, but they follow a consistent label-preserving philosophy that does not affect the semantics of the actions.

3.4.1 Augmentation of Spatiotemporal Visual Representations

The pathway for vision in this architecture consumes pre-processed video frames converted to fixed-size tensors with dimensions $T \times C \times H \times W$, where T is the number of frames, C the number of color channels, and H and W the spatial dimensions. For this input modality, there are two transformations that are performed with probability $p = 0.5$.

3.4.1.1 Horizontal Flipping

Horizontal frame flipping is probably the most commonly used photo-metric augmentation for action recognition since it takes advantage of the natural symmetry between the left and the right side of a human body while keeping the category of the performed action unaltered [36]. Yoga actions are typically defined based on the

configuration of arms, legs, and body regardless of its orientation, such that a video horizontally flipped corresponds to another instance of the same action class. For example, the right planking action becomes the left planking action but it remains an instance of the same pose class as shown in Figure 3.6. This augmentation helps in increasing the number of possible orientations in the training process which results in learning feature representations that do not depend on the orientation of the actor. This augmentation strategy is known to help improving accuracy in two-stream CNNs [16] and temporal segment networks [37], as well as in recent transformers for video analysis. The same transformation is applied to all frames of a given action sequence.

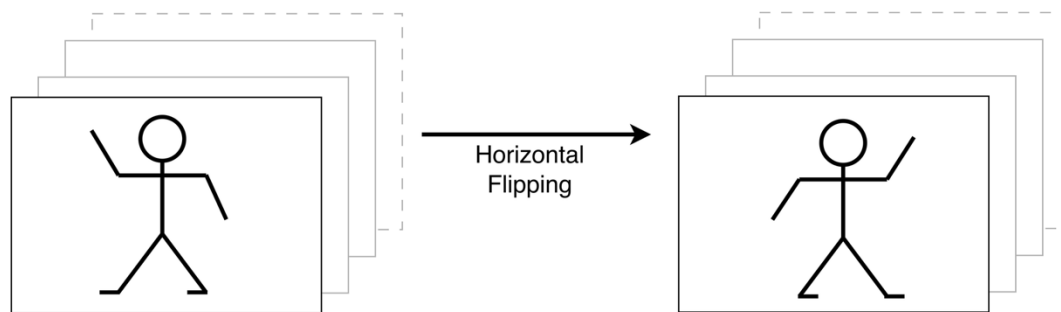


Figure 3.6: Horizontal Flipping of Videos

3.4.1.2 Random Spatial Translation

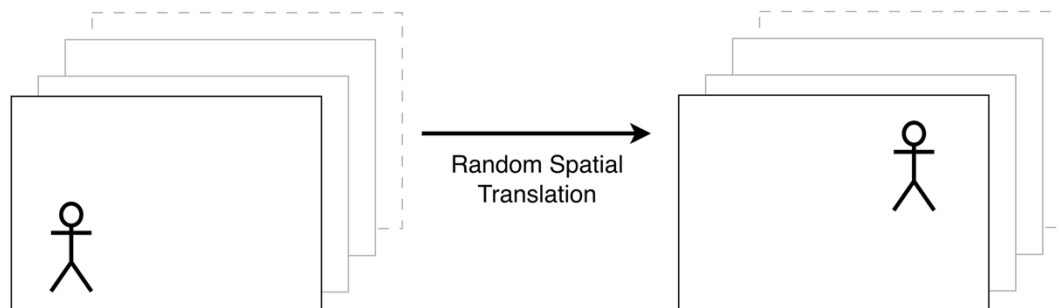


Figure 3.7: Random Spatial Translation of videos

The second type of affine transformation randomly translates the input by up to ± 16 pixels horizontally and vertically as shown in Figure 3.7. This technique simulates the spatial jitter that occurs due to the inaccurate detection of the object, small camera movements, or even small changes in the location of the subject in the scene. In the practical application of yoga exercise, it cannot always be expected that the subject will occupy the central position in the image. A robust detector should be tolerant of such shifts. Such a transformation technique is inspired by research conducted by Krizhevsky

et al. [38] in the area of image classification and its applicability in the case of videos has been proved for architectures like C3D [20] and I3D [21]. The displacement ensures that the body key points stay within the image boundary.

Using both of these operations separately results in an augmentation space which is created combinatorially. This means that a single training example can be horizontally flipped alone, translated alone, both of these, or neither, producing a four-fold increase in the augmented training data set per epoch. The use of randomization in augmentation is similar to that used by RandAugment [39], where no fine-tuning is required.

3.4.2 Augmentation of 3D Skeletal Sequences

The geometry-based pipeline operates on normalized 3D skeletal data, which has dimensions of $C \times T \times V$, where $C = 3$ (world coordinates in X, Y, Z directions) and $V = 33$ (MediaPipe joints). Here, we encounter an entirely different type of data augmentation, as this time the coordinates are continuous and refer to the actual world rather than being pixels from the screen. We use only one type of transformation – adding isotropic Gaussian noise with a standard deviation of $\sigma = 0.01$ meters.

3.4.2.1 Coordinate Noise Injection

Noise injection is a proven strategy for making skeleton-based action recognition networks more robust [6], [18]. Introducing small quantities of random noise into normalized joint coordinates prevents the network from learning exact numeric relationships and forces it to learn the actual structure of the kinematics involved. When it comes to 3D pose estimation, there is intrinsic noise in the normalized coordinates due to the inaccuracies of monocular 3D lifting [27]; therefore, injecting additional noise during training can be seen as a way of simulating the inherent variability at test time. This type of data augmentation is akin to input-level dropout and ensures that $\sigma = 0.01$ m represents about 1 cm of jitter in the world coordinate system, which is realistic considering that current state-of-the-art 3D pose estimation methods are accurate within the same range [40]. Our decision to use isotropic Gaussian noise, as opposed to joint- or axis-wise schedules, is a relatively modest regularizer since it takes into account the equal scaling factor of the metric space. Recent studies involving ST-GCN [15] and its derivatives have shown that joint-coordinate noise significantly boosts cross-subject and cross-view generalization performance.

Since the skeleton coordinates are pre-normalized by subtracting the hip center point, the noise addition does not result in any translational movements but rather mimics sensor inaccuracies. Such behavior is especially beneficial in the case of yoga, since fine changes in limb positions are used to differentiate between different postures, while the network needs to focus only on limb relative positions.

3.5 THE DUAL-MODAL DATASET

3.5.1 Modality-Specific Preprocessing and Representations

The joint data comprises the two modality repositories built separately using independent pipelines that were explained previously.

- **Appearance modality (Video Tensors):** Each trimmed action clip has been transformed into a fixed-size tensor of shape $T \times C \times H \times W$, where $T = 30$ frames, $C = 3$ RGB channels, and $H = W = 224$ spatial dimensions. In this way, the temporal resizing allows for having the same length input tensor irrespective of the initial duration of the action instance, whereas the spatial resizing aligns the tensor dimensions to the Swin3D-T video backbone [23].
- **Geometry modality (3D Skeletons):** For the same action instance, the landmark extraction process produces a variable-length sequence of 33 world-coordinate landmarks. This sequence is temporally resampled to $T = 30$ frames using a linear or edge-padding strategy, yielding a tensor of shape $3 \times 30 \times 33$ (coordinates $\times$ frames $\times$ landmarks). The coordinates are subsequently normalised to a hip-centred reference frame, removing absolute translation and leaving only the intrinsic pose configuration.

The modality-specific datasets such as *YogaTensorDataset* for videos and *YogaSkeletonDataset* for skeletons can independently support one single stream of models. Yet, the real strength of the framework lies in its combination.

3.5.2 Constructing a Paired Multimodal Instance

The DualYogaDataset represents (as shown in Figure 3.8) the logical connection between both datasets. Within, it creates both visual and geometry datasets on the same set of IDs and labels. Most importantly, it determines the set of IDs in common among the two datasets, thus discarding all sequences in which one modality is not present. This is crucial to ensure the one-to-one relationship between samples: every sample passed to the model contains both video tensor and skeleton tensor extracted from the same action clip.

For a training sample, the data loader will first load the tensor corresponding to the video frame and its associated label from the visual dataset, as well as the tensor for the skeleton from the geometric dataset. The label will only be loaded once (from the visual modality, as both use the same mapping) and will be combined along with the two inputs into a tuple of values $(\mathbf{V}, \mathbf{S}, y)$. No synchronization needs to be performed as both data modalities were already synchronized onto the same 30-frame timeline. The result is an avoidance of time warping for heterogenous datasets and ensures that, at each time step, the visual frame and skeleton pose match up to the real-world time.

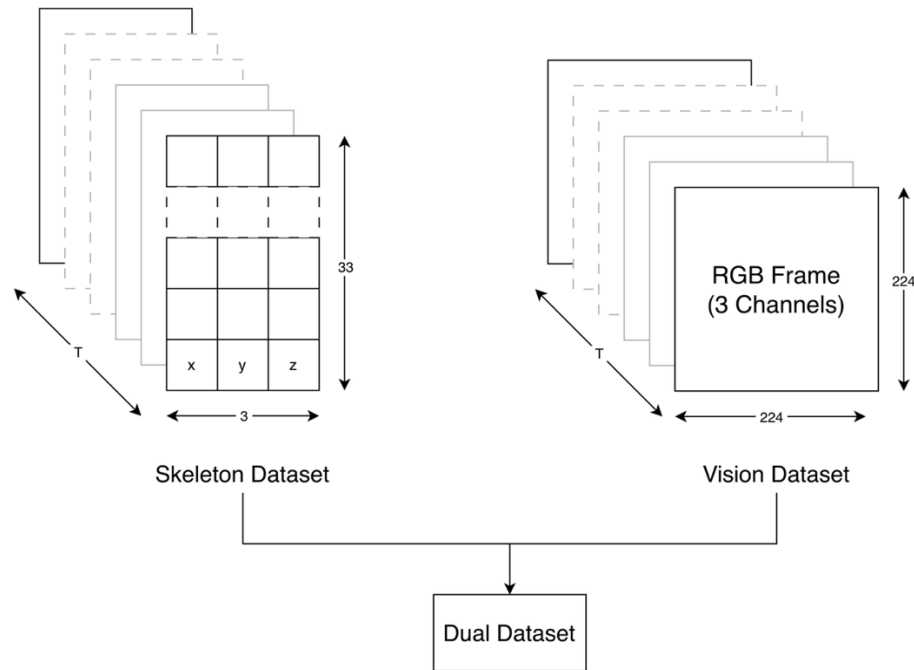


Figure 3.8: Dual Multi-modal Dataset formation

3.6 CLASS REBALANCING STRATEGIES

In real-world action recognition datasets, the distribution of the different classes is typically not balanced as shown in Figure 3.9. The 3DYoga90 dataset is no different; some yoga postures appear more frequently in the data than others, resulting in a long tail distribution which affects the training process [4]. Without intervention, this will make the model biased towards learning majority classes, sometimes even ignoring the smaller classes which could hold more significance. In order to overcome this challenge, this study employs two strategies of rebalancing; loss weighting and sampling weighting.

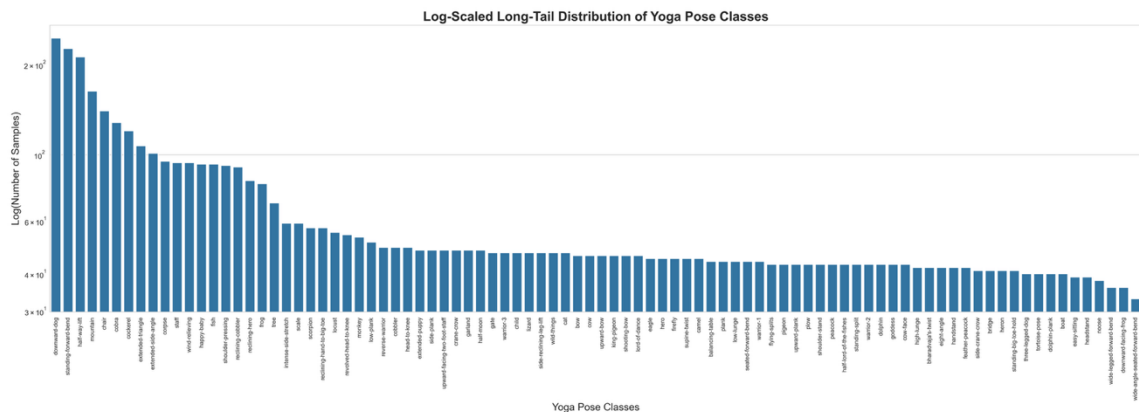


Figure 3.9: Log-Scaled Long-Tail Distribution of Yoga Poses showing significant class imbalance

3.6.2 The Class Imbalance Problem in Supervised Learning

Mitigation techniques can be classified into two broad groups: techniques at the data level where the distribution of samples in the training dataset is adjusted, and techniques at the algorithm level where the loss function is modified so as to assign higher penalties for misclassification in minority classes [41]. The two techniques included in this codebase implement one example of each classification method.

3.6.3 Loss-Based Rebalancing: Inverse-Frequency Class Weights

Inverse frequency-based weighting techniques have been around in machine learning algorithms for quite some time now. Cost-sensitive learning approaches [42] as well as modifications made for deep neural networks [43] use similar strategies of giving high weights to losses of less frequent classes in order not to overlook them while training. Such an approach makes sense whenever the minority classes have distinctive meaning and value, just like different yoga poses that require different levels of mechanical complexity without having any hierarchy. A downside of using inverse frequency weights lies in the possibility of amplifying noise and anomalies within the less represented data. However, since all instances in the 3DYoga90 database are unique poses without any noise and are properly edited, the problem becomes irrelevant [44].

3.6.4 Sampling-Based Rebalancing: Weighted Random Sampling

A weighted random sample has various benefits compared to undersampling. Firstly, weighted random sampling retains all the samples from the majority classes without losing any information but rectifies the class balance of the dataset [45]. Secondly, due to the fact that weighted random sampling occurs at the level of the data loader, the technique does not affect the model and the loss function; hence, it works well with any architecture without needing changes in design like output biases by class. It has also been empirically found that sampling techniques perform on par with or better than loss weighting in image classification [46].

3.7 TRAIN-TEST SPLITTING: THE CROSS-SUBJECT PROTOCOL

A strong evaluation protocol requires that training data and testing data be split in such a way that they cannot easily learn the traits specific to subjects and can be tested for their ability to generalize beyond them. In this case, the data split that satisfies this criterion for the 3DYoga90 dataset [4] is source-based cross-subject splitting where the basis of split is the unique source from which the videos have been captured (instructor).

3.7.1 The Perils of Random Instance-Level Splitting

If video instances are divided randomly into training and testing sets, clips with the same performer could end up in either set. Although computationally easy to perform, it creates an implicit and very dangerous form of leakage between datasets. Both CNNs and transformers are good at picking up correlations within a particular individual: facial

appearance, body form, attire, and even motion style. Thus, when evaluating the model using the random split method, it would be possible to get misleadingly high accuracy based not on the action itself but the identity of its performer [5].

This issue becomes more critical in cases like yoga because only a few trainers provide the most samples. This would mean that there will be samples from the same trainer in both sets, and the network can use postures unique to the trainer (such as a specific angle of the knee in a particular pose performed by one trainer). In effect, this evaluation measures closed-set person identification instead of open-set pose classification. The benchmark becomes irrelevant to the practical application because the system needs to generalize beyond the training set.

3.7.2 Source-Based Splitting as a Proxy for Subject Independence

In order to control for actor confounding, the dataset is split using source ID, which is a unique identifier for the source video that corresponds to a video clip. In the 3DYoga90 dataset, each source video contains one person performing various sequences of yoga poses [4]. By putting all clips belonging to the same source in either the training set or the validation set, we can avoid having the same person included in both datasets. This strategy follows the conventional methodology used in cross-subject benchmarking of 3D human action analysis, e.g., NTU RGB+D [7] and NTU RGB+D 120 [47].

The algorithm works in two steps. In the first step, all instance IDs are grouped based on their source, which gives us a mapping between the source and the list of its instances. The sources are then randomly shuffled and split into an 80% training and 20% validation set. Lastly, the instances in each source are put together in order to create the partitions. Since the split was done at the source level, both the training and validation datasets include disjoint sets of teachers, and therefore test the ability of the model to detect yoga postures regardless of body shape and style of practice.

3.7.3 Statistical Rationale and Validation

The 80/20 rule has become an accepted heuristic which allows the acquisition of enough samples for convergence of the model, while still having a sufficient number of samples for the validation set to achieve stable estimates on performance. The 80/20 rule is applied to the number of sources instead of the number of instances to ensure that there is representation of poses, though from different subjects in each partition. Source-overlap checks then follow to make sure that none of the sources are common in the two partitions. Figure 3.10 shows a Sankey diagram for the train-test split.



Figure 3.10: Sankey Diagram showing Cross Subject Dataset Split

3.8 EVALUATION METRICS

Assessing the performance of a multi-class action recognition model is a process that entails using a number of metrics which together measure the overall accuracy of the model, the sensitivity of the model to unbalanced class distributions, and the capability of the model to discriminate between various pose classes. The Evaluation metrics utility calculates five metrics based on the results generated by the model when presented with the test dataset. These metrics include accuracy, balanced accuracy, macro average F1-score, confusion matrix, and recall for each pose class.

3.8.1 Overall Accuracy and Its Shortcomings

Overall accuracy is the most straightforward and easy to interpret performance measure, which is given by the ratio of correctly classified samples to the total number of testing samples as shown in the Equation 3.1:

$$Accuracy = \frac{1}{N} \sum_{i=1}^N 1 [\hat{y}_i = y_i] \quad (3.1)$$

where N is the total number of test samples, $\hat{y}_i$ is the predicted class, and y_i is the ground truth. Although this gives a quick overall picture of how good a classifier is performing, there is a serious limitation of this measure, namely, it fails to give us any information about poor classification results for less prevalent classes [43]. When working with an unbalanced yoga dataset where one class can appear hundred of times, while another just a few (for example, firefly pose versus mountain pose), one can build a classifier which just predicts the majority class and achieves high accuracy, although being completely meaningless clinically.

3.8.2 Balanced Accuracy: Equalising Class Influence

The calculation of the balanced accuracy aims to compensate for the bias towards majority classes through averaging the recall per class and thus treating each pose equally irrespective of its occurrence frequency [48] as shown in Equation 3.2:

$$\text{Balanced Accuracy} = \frac{1}{C} \sum_{c=1}^C \frac{TP_c}{TP_c + FN_c} \quad (3.2)$$

where C is the total number of classes, TP_c is the true positives of class c , and FN_c is the false negatives. In fact, this measure represents the accuracy expected when the test set is perfectly balanced. Balanced accuracy is an especially suitable choice in the case of the 3DYoga90 dataset since the cross-subject split in Section 3.7 deliberately leads to a skewed distribution of the training set compared to the distribution of the validation set in terms of subject as well as possibly class occurrences. Comparing the metrics alongside raw accuracy can demonstrate the reason behind improved results: if they are attributed to better results for majority classes only or not [32].

3.8.3 Macro-Averaged F1 Score: Harmonising Precision and Recall

The F1 score represents the harmonic mean of precision and recall for one particular class. However, when computing macro-averaged F1 scores in a multi-class setup, the F1 scores of all classes are computed separately before averaging their values equally [49] as shown in Equation 3.3:

$$F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (3.3)$$

$$\text{Macro F1} = \frac{1}{C} \sum_{c=1}^C F1_c \quad (3.4)$$

where,

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \quad (3.5)$$

The macro-F1 score takes into account both the false positives and false negatives, which makes it a stricter measure compared to balanced accuracy. In case of pose estimation for yoga practice, having a high macro-F1 implies that the algorithm not only correctly identifies most of the occurrences of the pose (recall), but also doesn't confuse poses with each other (precision). It is very important in the process of instructing since a false positive may cause the algorithm to provide incorrect recommendations. The previous studies in skeleton-based action recognition have used the macro-F1 score as an

evaluation measure since it negatively rewards both over- and underestimating specific actions [18].

3.8.4 Combined Interpretation and Practical Use

The use of all five measures together obeys the principle that a number cannot fully represent the multidimensional nature of multi-class classification tasks [50]. While accuracy provides a simple snapshot, it should be supplemented by other measures for proper evaluation. Insensitive to the distribution of classes, balanced accuracy and F1 provide the first two numbers that determine the quality of the classifier and used as main criteria for hyperparameter tuning and model selection. For the dual-modal yoga recognition task, all three metrics are calculated for vision-based, skeleton-based, and fusion approaches alike.

In compliance with the evaluation procedure suggested in the action recognition literature [7], the current study contributes to reliable and reproducible research.

3.9 CTR-GCN ARCHITECTURE: SKELETON-ONLY BASELINE

Graph Convolutional Neural Networks (GCNs) have emerged as the default technique for skeleton-based action recognition through the use of a graph that represents the human body. Although previous work such as the ST-GCN [18] has proved the value of such a representation, it was based on predefined adjacency matrices of the bones of the human skeleton. However, it is evident that such fixed representations fail to account for action-dependent kinematics. For example, there is no connection between the hands and the feet in the physical skeleton; however, their spatial correlation determines the difference between yoga poses like Warrior I and Warrior II.

As a solution to this limitation, we select the Channel-wise Topology Refinement Graph Convolutional Network (CTR-GCN) [6] as our geometry stream baseline model. Contrary to previous adaptive networks which learn only one global topology [19], the CTR-GCN architecture learns unique adjacency matrices per channel using a cross-channel attention approach. The pairwise similarity of the joints per channel is computed based on a given feature map, and as a result, each channel focuses on particular kinematic features, such as joints of limbs in case of balance yoga poses compared to torso joints in seated poses [6]. Such refined channel information is used in addition to the existing static anatomical priors from MediaPipe.

3.9.1 Overall Architecture and Data Adaptation

The entire network architecture (Figure 3.11) is tailored for processing the sequences of 3D world coordinates obtained using 33-joint MediaPipe [51] pipeline. The initial input tensor, (C_{in}, T, V) , where each dimension corresponds to coordinate, frames, and joint respectively, is successively projected to higher-level features using six spatial-temporal layers. The global average pooling operation collapses the two dimensions into a 256-dimensional vector.

The CTR-GCN benefits from channel-wise topology optimization and functions as an extremely compact yet powerful geometry expert. Within the scope of our multimodal architecture, this particular network is pretrained separately and frozen thereafter. The logits predicted by this network serve as the viewpoint invariant and kinematically sound ground truth for the refinement of our cross-attention fusion.

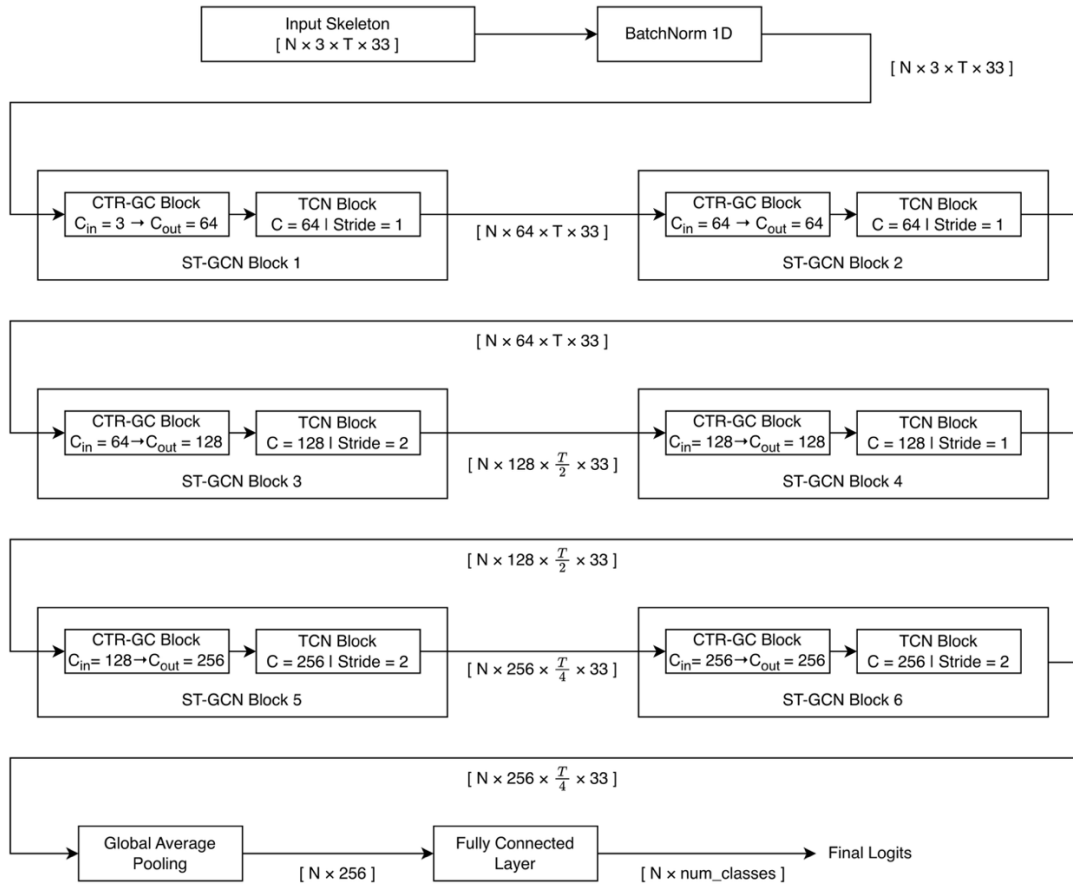


Figure 3.11: CTR-GCN: Overall Architecture

3.9.2 Role as a Baseline and Foundation for Multimodal Fusion

Within the larger multimodal pipeline, the CTR GCN operates as the geometry stream, feeding on the hip centred 3D skeletal coordinates obtained by analysing each trimmed yoga clip. The choice of the CTR GCN is informed by its superior state of the art performance on skeleton based datasets [3], while also being flexible enough to enable extraction of feature tokens. This can be done through pooling of the feature tensor produced by the final spatial temporal block, which has dimensions $256 \times T' \times V$, where T' represents a reduced temporal dimensionality. Later on, the CTR-GCN will produce its feature maps in order to project them into a common joint embedding space, and then

fuse them with visual feature vectors using a cross-attention network. To precisely measure the contribution made by multimodal fusion and cross-modal attention, it is first necessary to establish CTR GCN as a competitive unimodal baseline.

3.10 VIDEO SWIN TRANSFORMER: APPEARANCE-BASED BASELINE

Although skeletons offer a concise and viewpoint-independent description of human motions, this approach neglects the additional context information available in video data such as the practitioner's alignment to the surroundings, clothing signals, and detailed visual cues on the texture of the mat and the background, which could help distinguish between subtle poses. As an accompanying unimodal input channel that complements the geometric CTR-GCN, we use a vision-only baseline consisting of the state-of-the-art video transformer architecture known as the Video Swin Transformer [23]. This network takes the preprocessed video tensor as input, capturing visual information in terms of appearance and motion in an end-to-end manner.

3.10.1 Transfer Learning from Kinetics-400

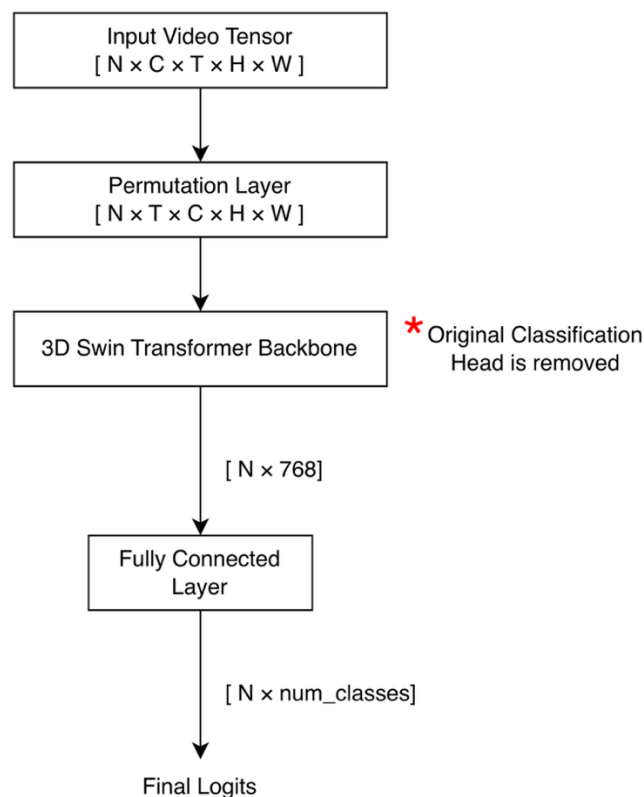


Figure 3.12: Video Swin Transformer: Updated classification head to support Yoga Pose Classification

Starting from scratch while training the model based on a dataset like 3DYoga90 is not a wise choice since it may not be very data efficient, and overfitting may be one issue. As a result, the backbone will be initiated using a pre-trained model on Kinetics 400 [26], a large-scale dataset used for action recognition tasks with 400 classes of human actions covering various actions from sports, daily activities, and interacting people. This dataset offers a wide variety of motion types and backgrounds, and pre-trained models have been observed to generalize well to other video tasks [23]. The model pre trained on this dataset has encoded a vocabulary of features related to optical flow, motion boundaries, and body part detectors, all of which would serve as a good foundation for yoga-specific features.

All the layers of the backbone architecture are carefully tuned because freezing earlier layers is generally disadvantageous in case the target and the source domains differ greatly in terms of the look (yoga studios, indoor environment) and action semantics (static postures versus dynamic moves). The original classifier is substituted with a simple linear model that takes a 768-dimensional pooled feature and outputs predictions for C yoga poses as shown in Figure 3.12.

3.10.2 Role as a Vision-Only Baseline and Future Fusion

Video Swin baseline is trained separately to determine the upper bound of performance attainable from appearance features alone. It will provide a benchmark against which performances from the skeleton only CTR-GCN model and our proposed multimodal fusion model will be compared. For the fusion architecture, Video Swin will be pruned before the global pooling block and the multiscale feature maps extracted by the network will be converted to a sequence of visual tokens. Geometry tokens from the CTR-GCN will then be added to the visual tokens to construct a cross-attention layer for the purpose of aligning visual and skeletal feature embeddings. The hierarchical structure of Video Swin and window-based approach allows this task to be efficiently accomplished since the intermediate feature maps outputted by the backbone network can be split into small spatiotemporal patches that behave as individual tokens. By having a vision stream that has been pre trained to extract very strong features, the subsequent cross modal attention mechanism will not have the extra burden of compensating for poor unimodal feature representations.

3.11 MULTIMODAL FUSION VIA LOGIT-LEVEL CROSS-ATTENTION

Two unimodal baselines have been introduced in previous sections, namely, Skeleton based CTR-GCN working in 3D world coordinates and Video-based Swin Transformer trained on RGB frames. Both perform well individually, but there is scope for improving performance further by utilizing the complementary information offered by appearance and geometric cues via a fusion approach. The last architecture proposed in this work, referred to as Logit-Level Cross-Attention Fusion, is the result of fusing the pre-trained frozen backbone models using a cross-attention head trained on the outputs produced by the logit heads of these backbones.

3.11.1 Design Rationale: Frozen Experts and Late Fusion

The task of combining two deep neural networks into one is usually solved at the feature-level stage by concatenating or aligning the intermediate representations before feeding them into the final classifier [8], [28]. Nevertheless, this strategy requires joint training or fine-tuning of both networks simultaneously, which entails high computational and memory costs, particularly in the case of video transformers using 3D attention. Another strategy, based on recent research on combining pre-trained models [29], [52], is freezing both encoders completely and using their outputs to feed into a separate decision-level fusion network. Thus, we decouple the process of tuning the streams separately from their combination in the latter, thus making it possible to focus each encoder specifically on its domain of expertise without interfering with other components.

There are various reasons why the combination of models is performed at the logit level rather than at the feature level. First, logits capture compactly the information contained in a set of features representing the beliefs of each modality regarding all possible classes in a single vector of evidence. Second, combining models at the level of logits is consistent with the idea of classifier stacking and mixtures of experts [9]. In the latter case, the meta-classifier modifies the outputs produced by the expert models. Finally, such an approach allows us to train a small fusion network.

Another vital decision made in the above framework is treating the skeleton stream as a prior estimator, and the video stream as a corrective term to the skeleton logits. The rationale behind such a decision is based on the fact that the CTR-GCN, which uses 3D coordinates as input, is less prone to variation from the background and clothing; hence, the skeleton stream will produce better estimates than the video stream. On the other hand, the video stream excels at capturing appearance-based information that can be used to discriminate poses with similar joint positions but distinct hand positions.

3.11.2 The *LogitPerceiver*: Cross-Attention over Class Predictions

The fusion head, which we call the *LogitPerceiver*, is based on the idea of the Perceiver model [53], which employs a learnable query mechanism to focus on a collection of input tokens. Here, the input tokens correspond to projected logit vectors for the two modalities, while the query corresponds to a learnable vector combining information from both modalities. The resulting output from the cross-attention layer is then used to compute a residual correction on the logits corresponding to the skeleton.

3.11.2.1 Tokenisation of Logits

Each backbone produces a logit vector $\mathbf{l}_{vis}, \mathbf{l}_{geo} \in \mathbb{R}^C$, where C is the number of classes. These vectors are projected into a common D -dimensional embedding space (with $D = 384$ in our implementation) using a shared linear layer:

$$\mathbf{t}_{vis} = \mathbf{W}_p \mathbf{l}_{vis}, \quad \mathbf{t}_{geo} = \mathbf{W}_p \mathbf{l}_{geo} \quad (3.6)$$

The two token vectors are stacked to form a set $\mathbf{T} \in \mathbb{R}^{2 \times D}$ that serves as the key-value input to the cross-attention module. The use of a shared projection encourages the model to represent both modality logits in a comparable metric space, facilitating meaningful attention interactions. Figure 3.13 shows the formation of the stacked tokens.

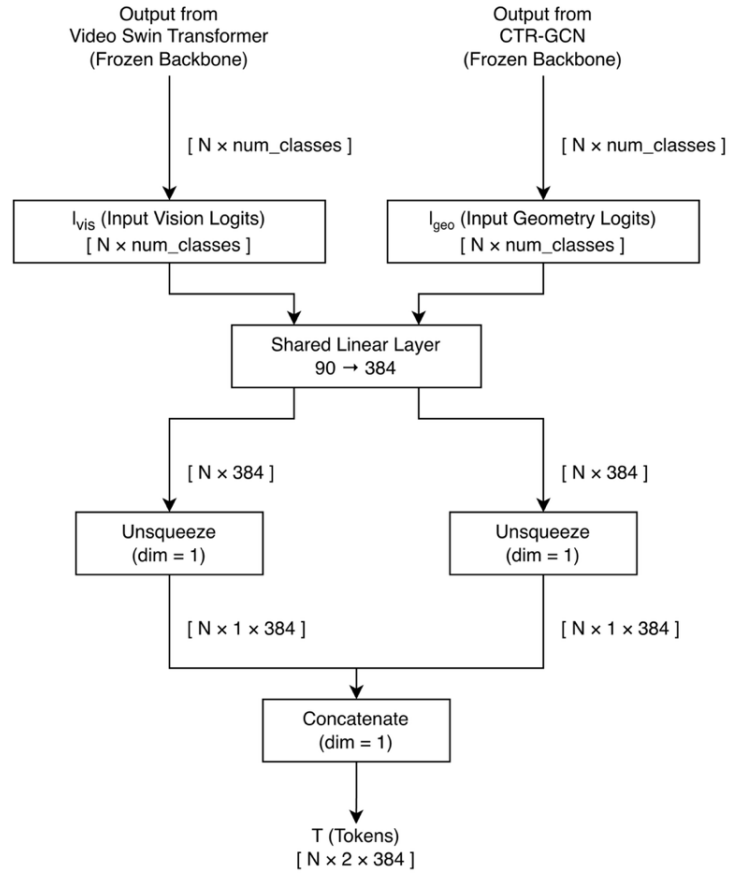


Figure 3.13: LogitPerceiver: Tokenisation of Logits

3.11.2.2 Learnable Query and Cross-Attention

A single learnable query vector $\mathbf{Q} \in \mathbb{R}^{1 \times D}$ is initialised randomly and expanded across the batch. Multi-head cross-attention [11] is then applied with $\mathbf{q}$ as the query and $\mathbf{T}$ as both key and value as shown in Equation 3.7:

$$\mathbf{Q}' = \text{CrossAttn}(\mathbf{Q}, \mathbf{T}, \mathbf{T}) \quad (3.7)$$

This operation allows the query to pool information from both the visual and geometric logit tokens, weighting their relevance dynamically. The query can be interpreted as a learned meta-learner that asks: "given the two experts' opinions, what correction should be applied to the skeleton's prediction?" Because the cross-attention is computed with only two tokens, its computational cost is negligible, enabling rapid

training and inference. A residual connection followed by layer normalisation [54] is applied to the query as shown in Equation 3.8 and Figure 3.14.

$$\mathbf{Q}'' = \text{LayerNorm}(\mathbf{Q} + \mathbf{Q}')$$
(3.8)

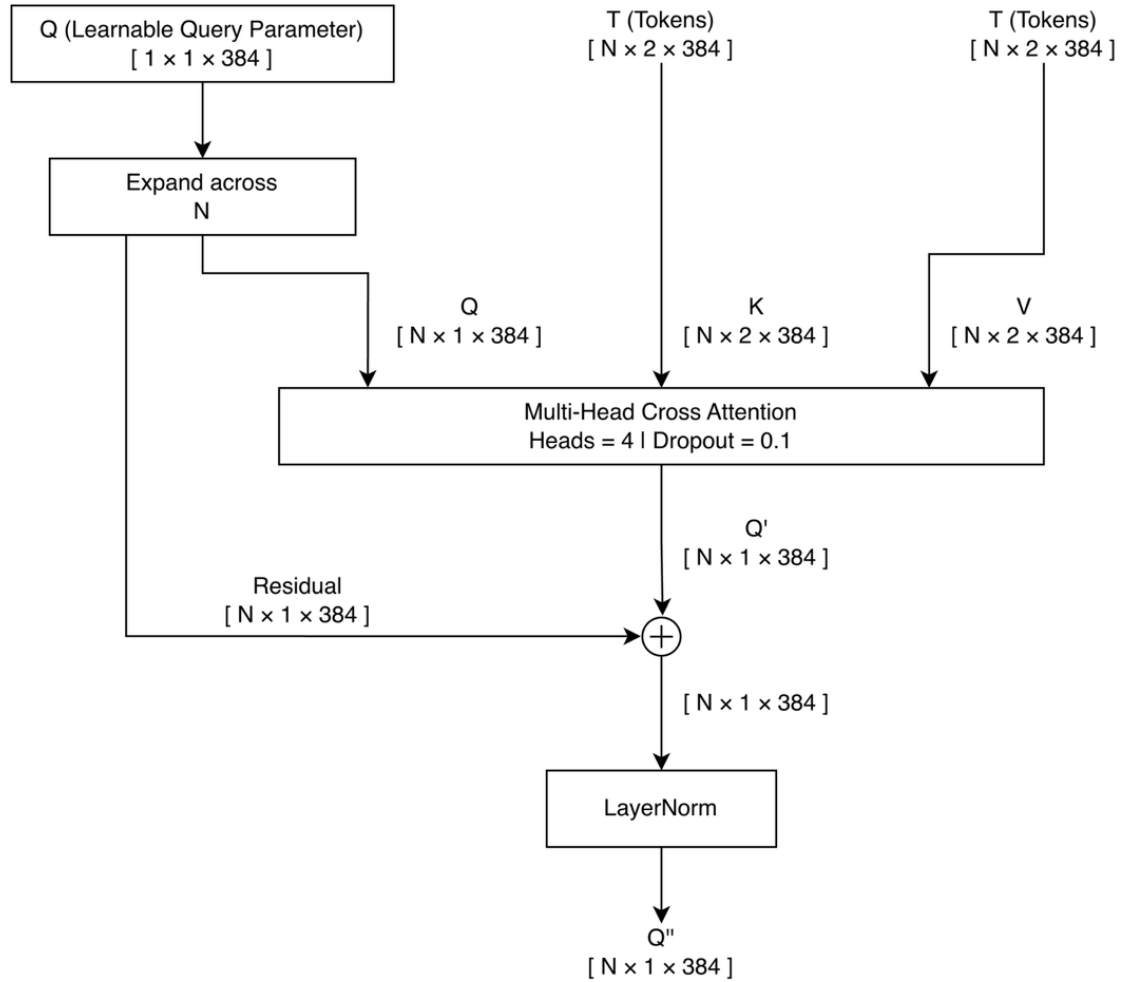


Figure 3.14: LogitPerceiver: Learnable Query and Multi-head Cross-Attention

3.11.2.3 Feed-Forward Refinement

The attended query is further processed by a feed-forward network (FFN) consisting of a linear expansion to $4D$, a GELU activation [55], dropout [56], and a linear projection back to D as shown in Equation 3.9:

$$\mathbf{Q}''' = \text{FFN}(\mathbf{Q}'')$$
(3.9)

A second residual connection and layer normalisation are applied, yielding the final hidden representation $\mathbf{H} \in \mathbb{R}^D$. This FFN provides the non-linear capacity needed to

model complex interactions between the two modalities' class beliefs. Figure 3.15 and Figure 3.16 shows the FFN architecture and residual connections.

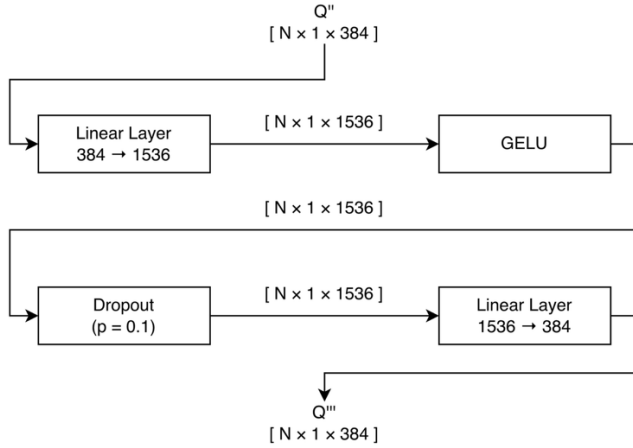


Figure 3.15: LogitPerceiver: Feed-Forward Refinement

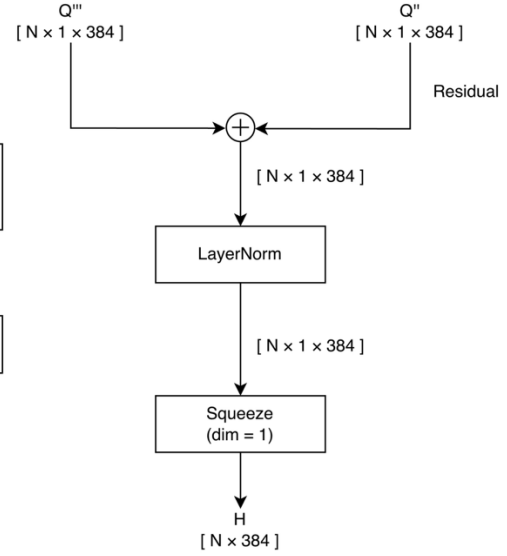


Figure 3.16: LogitPerceiver: Building the hidden representation from Residual

3.11.2.4 Residual Delta Prediction and Gating

The hidden representation $\mathbf{h}$ is then mapped to a delta logit vector $\boldsymbol{\delta} \in \mathbb{R}^C$ through a two-layer projection with intermediate GELU and dropout as shown in Equation 3.10:

$$\boldsymbol{\delta} = \text{Proj}_{\boldsymbol{\delta}}(\mathbf{h}) \quad (3.10)$$

The final fused logits are obtained by a residual connection to the skeleton logits, controlled by a learnable scalar gate α as shown in Equation 3.11:

$$\mathbf{l}_{fused} = \mathbf{l}_{geo} + \tanh(\alpha) \cdot \boldsymbol{\delta} \quad (3.11)$$

The gating mechanism is pivotal. Initialised at zero, $\tanh(\alpha) \approx 0$ at the start of training, meaning the model initially outputs exactly the skeleton logits. As training progresses, the gate can open to allow the visual-derived correction to influence the prediction, but the hyperbolic tangent bounds the gating factor to $(-1,1)$, preventing the delta from overwhelming the geometry base. This design reflects the intuition that the skeleton stream is the primary predictor, while the vision stream provides a bounded refinement.

Residual architecture, similar to ResNet [36], helps in reducing optimization difficulties for the model, making it learn the identity function in case extra modeling power is not required. Figure 3.17 and Figure 3.18 shows the Delta projection and fused logits.

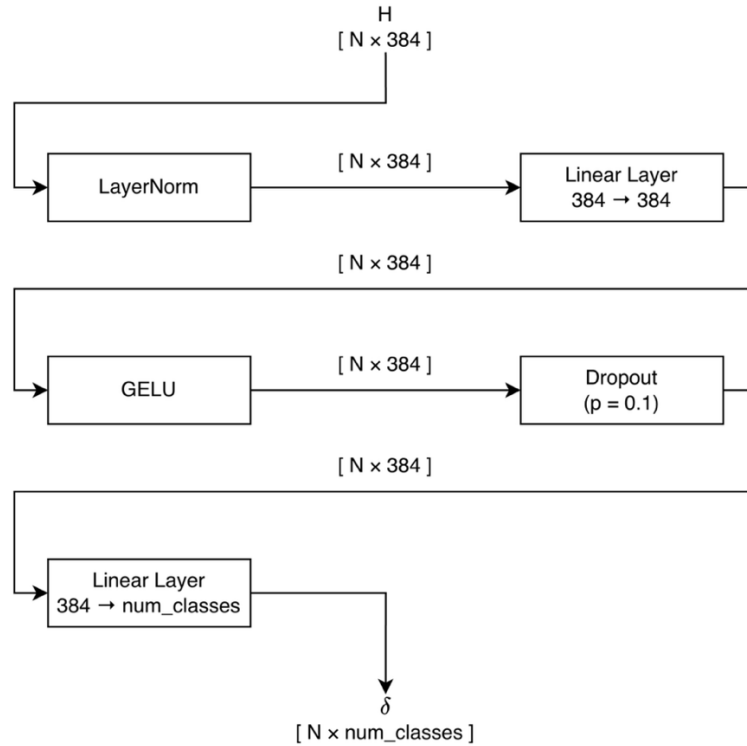


Figure 3.17: LogitPerceiver: Delta Projection

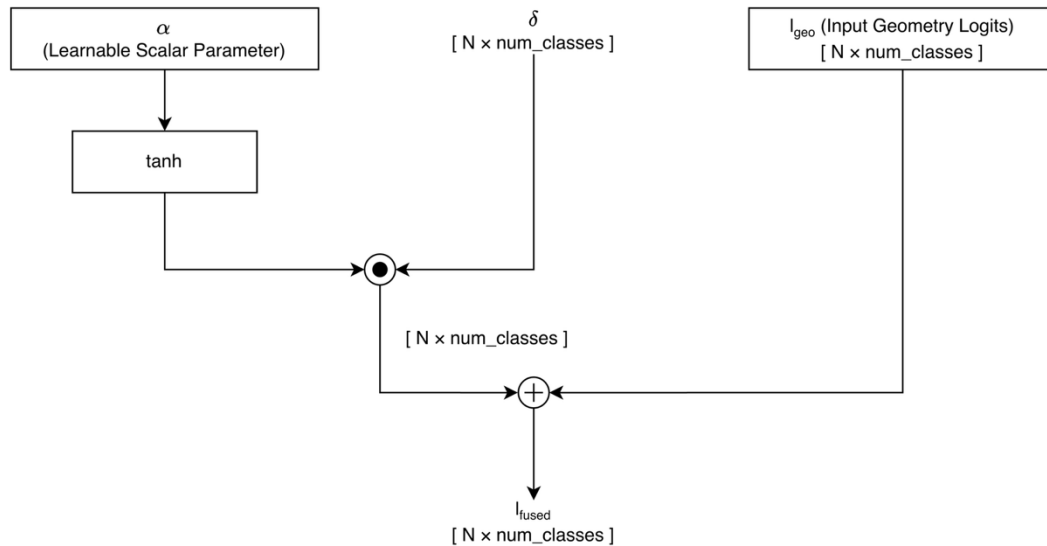


Figure 3.18: LogitPerceiver: Gating and Fused Logits

3.11.3 Significance as the Final Model

The logit-level cross-attention fusion network represents the final architectural structure used in the multiform yoga pose recognition framework. In the process of fusing information from the 3D geometric shape and RGB imagery modalities, the method leverages the strengths of both forms of representation without having to fine-tune the joint information. The motivation behind this approach is based on domain knowledge, as the essence of yoga poses lies in their joint structure, though visual information can be helpful in resolving certain ambiguities. As a result, the model can decide, through the learnable cross-attention query, whether or not to give weight to either modality of representation depending on the situation.

Practically speaking, studies have proven that this kind of logit-level fusion performs at least equally well as feature-level fusion when the separate models themselves are strong enough and properly calibrated. As such, leveraging existing state-of-the-art backbones as a frozen module, this architecture can be considered an efficient way of doing 3D yoga pose recognition, and its modularity allows for further exploration into other modalities or encoders in future work.

CHAPTER 4

EXPERIMENTAL SETUP AND RESULTS

4.1 HARDWARE AND IMPLEMENTATION DETAILS

Training and testing of all three architectures: the CTR-GCN skeleton network, Video Swin Transformer, and the logit-level cross-attention fusion framework were conducted using a NVIDIA RTX 5070 GPU instance available on cloud platforms. The purpose of this section is to outline the computation resources, software settings, and memory management tools utilized for the implementation of spatiotemporal deep learning using only a single GPU.

4.1.1 Software Environment and Library Stack

The whole source code was developed using Python 3.10 and depended on PyTorch deep learning library (v.2.1), which is the principal library for tensor computations and auto differentiation. The main libraries used in the process of pipeline creation are listed below:

- **torchvision (v.0.16):** Contributed the pre-trained Video Swin Transformer model backbone and associated weights on the Kinetics-400 classification task, as well as affine and flipping transformation for data augmentation purposes.
- **torch.amp:** Automatic mixed precision module in PyTorch, used to cast forward/backward passes in half precision (float16) and thus decrease GPU memory usage and speed up training using tensor cores [59].
- **CUDA/cuDNN:** Deterministic cuDNN routines were enforced to ensure reproducibility, while cuDNN auto-tuning was switched off to avoid non-deterministic algorithms being selected.
- **NumPy and Pandas:** Underpinning libraries used for all array manipulation and working with tabular data, especially in the skeleton dataset class where 3D coordinates of landmarks are reshaped and resampled.
- **scikit-learn:** Source of implementations of balanced accuracy, macro-average F1-score, and confusion matrix to measure classifier performance.
- **MediaPipe (v.0.10):** Pose landmarker is utilized at the stage of skeleton extraction, providing 33 landmarks expressed as coordinates in world coordinates from each frame of video.
- **OpenCV (cv2):** Used for reading video files, flipping frames, and converting between colour spaces at the stage of skeleton extraction.
- **tqdm:** Progress bars were provided by this library during data loading and training stages.
- **Matplotlib and Seaborn:** For visualization purposes

4.1.2 GPU Configuration and Mixed-Precision Training

Because of the limited capacity of the RTX 5070 with 12 GB RAM, it was necessary to train the video-based models using mixed precision. For this, the GradScaler from the PyTorch library scaled the loss to avoid underflow of the gradients for half precision, while the `torch.autocast` feature converted eligible mathematical operations (such as convolutions and matrix multiplication) into float16 data type [57]. The graph convolution network (CTR-GCN), having a light-weight structure, could work with batch size 32 using full precision, but mixed precision was still utilized.

In case of the Video Swin Transformer that could take in only 4 instances for batch size in float32, mixed precision helped double the available memory, making sure that the gain in speed was achieved while keeping the batch size unchanged. For the fusion model, training of the light-weight cross attention head on top of frozen backbones meant that even a batch size of 16 worked just fine.

Further stabilising the training process, gradient clipping was done up to the maximum of norm equal to 1.0 for all model architectures. Flags for the CUDA memory allocator were used to decrease memory fragmentation in lengthy training processes. Data tensors were transferred to the GPU using pinned memory and multi-workers for data loading.

To conclude, with the aid of a contemporary consumer-level GPU, a well-packaged software application, and meticulous memory management through mixed precision, three deep networks were successfully trained at low cost, illustrating that it is possible to achieve state-of-the-art multimodal yoga pose detection without requiring large-scale resources like those found in data centres.

4.2 COMMON TRAINING PROTOCOLS

4.2.1 Optimisation Algorithm: AdamW

Network updates are made through the use of the AdamW Optimizer [58], which is a modified version of Adam and separates weight decay from the gradient descent update rule. For regularized Adam training where there is an L_2 regularization penalty on top of the usual Adam learning scheme, the effect is that the weight decay component gets multiplied with the adaptive learning rate scheme. The net result is that there is an unintended increase in regularization for parameters whose history of gradients was small compared to the rest. The key difference with AdamW is that weight decay acts independently of the adaptive gradient descent.

4.2.2 Learning Rate Schedule: Cosine Annealing

The learning rate is decayed from the initial value following a cosine annealing schedule without restarts [59] as shown in Equation 4.1:

$$\eta(t) = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\pi \frac{t}{T}\right)\right) \quad (4.1)$$

where t is the epoch counter, $T = 50$ is the number of epochs, $\eta_{max} = 10^{-3}$ and $\eta_{min} = 10^{-6}$. Cosine annealing was selected on the basis of its gradual and monotonic decay pattern, which guarantees a natural decrease in the step size as the network moves towards the local minimum, compared to a sudden drop of the step-based schedules, that could lead to unstable optimisation [59]. The minimum step size of 10^{-6} is critical in allowing the network to update its weights throughout all the epochs, especially since early stopping is used here.

4.2.3 Early Stopping and Model Selection

To ensure the absence of overfitting to the training dataset and selection of the model with optimal generalisation capabilities, the training process employs an approach called early stopping with patience. The metric on the basis of which the model is selected is balanced accuracy on the validation dataset, taking into account that all poses are treated equally, not considering their frequencies. A new model is saved only if its balanced accuracy exceeds the accuracy of the previous model by at least 0.1%. This value of threshold prevents selecting a new model because of random fluctuation. Training is stopped if there is no improvement in 10 consecutive epochs.

4.3 TRAINING PROTOCOL FOR SKELETON-ONLY CTR-GCN

The effectiveness of a deep neural network depends on both the model's architecture and the choice of the optimisation approach, regularisation schemes, and data handling methods used throughout training. The following section discusses the training methodology used in training the CTR-GCN model for skeleton-based yoga pose recognition. Each of these decisions is informed by the unique characteristics of the 3DYoga90 dataset: relatively small size, imbalanced classes, and cross-subject generalisability.

4.3.1 Data Sampling Strategy

In addition to the loss-level correction, a balance at the data level is obtained via a weighted random sampling approach. Sampling weights of training samples are computed based on the inverse of the frequency of the corresponding classes. As mini-batches are selected for creating epochs, rare classes will be oversampled. Thus, at each iteration of the learning process, a roughly balanced set of examples will be observed for computing gradients, which helps the classifier stay away from majority-class bias [41]. This approach works along with the use of class-based weighted loss; although there is an overlap between the two techniques, their combined use guarantees a safer path towards balanced training and has proven to work consistently well [60].

4.3.2 Summary of Hyperparameters

Table 4.1: Training Hyperparameters for CTR-GCN

Parameter	Value
Optimiser	AdamW
Learning rate (initial)	1×10^{-3}
Weight decay	1×10^{-2}
LR schedule	Cosine annealing to 10^{-6}
Epochs (maximum)	50
Batch size	32
Loss function	Cross-entropy + label smoothing (0.03)
Class weighting	Inverse frequency
Training sampler	Weighted random (balanced)
Mixed precision	Float16 (AMP)
Gradient clipping max	1.0
Early stopping patience	10 epochs
Early stopping threshold	0.1% balanced accuracy

Table 4.1 shows the list of hyperparameters used for training. Through this well-designed training process, the CTR-GCN is able to perform excellently in the skeleton-only yoga pose recognition while remaining stable and repeatable, thus forming an excellent baseline for future research in multimodal fusion of yoga recognition.

4.4 TRAINING PROTOCOL FOR VIDEO SWIN TRANSFORMER

Selection of Video Swin Transformer [23] for the appearance-based stream of the multimodal framework took place. With respect to the moderate number of samples in the 3DYoga90 dataset, training a deep three-dimensional transformer de novo is data hungry and leads to an extremely poor generalization, thus, overfitting. Instead, we have resorted to a transfer learning approach, which consists in fine-tuning the pre-trained network on action recognition on the target task.

4.4.1 Transfer Learning from Kinetics-400

The Video Swin backbone model was pretrained using the weights obtained from Kinetics-400 [26] database comprising around 240,000 videos belonging to 400 different classes of human actions. The use of Kinetics dataset for pretraining results in the acquisition of a diverse feature vocabulary by the model including detectors of edges, motion boundaries, and even parts, which are generally useful for video analysis tasks [61]. All layers of the Video Swin model, including the early stages of 3D patch embeddings and later transformer modules, were fine-tuned on the 3DYoga90 dataset.

Freezing of early layers is a common strategy adopted when there is a high visual similarity between the source and target datasets; however, the use of this approach is not recommended in the present case due to the significant differences between the indoor yoga environment and Kinetics.

4.4.2 Summary of Training Hyperparameters

Table 4.2: Training Hyperparameters for Video Swin Transformer

Parameter	Value
Pre-training dataset	Kinetics-400
Fine-tuning strategy	Full (all layers)
Optimiser	AdamW
Learning rate (initial)	1×10^{-4}
Weight decay	1×10^{-2}
LR schedule	Cosine annealing to 10^{-6}
Epochs (maximum)	30
Batch size	4
Loss function	Cross-entropy + label smoothing (0.03)
Class weighting	Inverse frequency
Training sampler	Weighted random (balanced)
Mixed precision	Float16 (AMP)
Gradient clipping max	1.0
Early stopping patience	5 epochs
Early stopping threshold	0.1% balanced accuracy

Table 4.2 shows the list of hyperparameters used for training. The well-thought-out strategy facilitated the adaptation of Video Swin Transformer's highly potent spatiotemporal representations to yoga, allowing for excellent results as a baseline unimodal model and forming a solid foundation for the multimodal approach that was used afterward.

4.5 TRAINING PROTOCOL FOR MULTIMODAL FUSION MODEL

Training of the cross-attention fusion model at the logit level aims at discovering the meta-learner, which should merge the predictions of two trained models. As the backbone models, i.e., Video Swin Transformer, and CTR-GCN, do not undergo changes during training, optimization can be performed only on the fusion head, which makes the process highly efficient, both from the memory and computational point of view. The following subsections describe the training procedure, including optimizer selection and evaluation metrics, providing theoretical rationale behind each step.

4.5.1 Meta-Learning over Frozen Backbones

The fusion model treats the two unimodal backbones as fixed function approximators that output logit vectors $\mathbf{l}_{\text{vis}}$ and $\mathbf{l}_{\text{geo}}$. At the time of training, the disabled torch gradient calculation mechanism ensures that there is zero gradient flow through such large networks, rendering it unnecessary to retain any activations from these networks to perform backpropagation through them. The approach adopted here, which is similar to the concept of classifier stacking [10] and meta learning from mixtures of experts [9], enables the model to utilize the representation capacity of existing advanced video and skeleton modeling architectures without having to train them. All the trainable parameters of the model exist in the LogitPerceiver head, including logit projection, cross attention mechanism, feed forward network, delta projection, and gating scalar; all this accounts for less than 2 million parameters.

4.5.2 Learning Rate Schedule with Warm-Up

The learning rate schedule consists of a linear warm up period, followed by a cosine decay. In the initial three epochs, the learning rate is increased linearly from 0 to 1×10^{-3} . The warming up process is an important step in the training of transformer models due to the fact that the query and key vectors are initialized randomly, hence employing a high learning rate right away would cause the update of attention matrices to be too drastic [62]. Following the warming up phase, the learning rate decays along a half-cosine curve, with the end epoch having a learning rate value of around 5×10^{-7} [11], [63].

4.5.3 Summary of Hyperparameters

Table 4.3: Training Hyperparameters for Logit-Level Cross-Attention Fusion

Parameter	Value
Trainable component	LogitPerceiver head (≈ 1.8 M params)
Backbones	Frozen Video Swin, Frozen CTR-GCN
Optimiser	AdamW
Learning rate (initial)	1×10^{-3}
Weight decay	5×10^{-3}
LR schedule	Linear warm-up (3 epochs) $\rightarrow$ cosine decay to $\sim 5 \times 10^{-7}$
Epochs (maximum)	30
Batch size	16
Loss function	Cross-entropy + label smoothing (0.05)
Class weighting	Inverse frequency
Training sampler	Weighted random (balanced)

Mixed precision	Float16 (AMP)
Gradient clipping max	1.0
Early stopping patience	20 epochs
Early stopping threshold	0.05% balanced accuracy

Table 4.3 shows the list of hyperparameters used for training. The precisely engineered training protocol enables the logit-level cross-attention fusion model to effectively learn how to fuse the complementary capabilities of the two streams without having to incur the exorbitant cost of end-to-end fine-tuning, thus obtaining outstanding results on the 3DYoga90 dataset in a cross-subject setting.

4.6 EVALUATION AND RESULTS

This section presents a comprehensive empirical evaluation of the three architectures: the CTR-GCN skeleton-only baseline, the Video Swin Transformer vision-only baseline, and the proposed logit-level cross-attention fusion model, on the 3DYoga90 dataset under the rigorous cross-subject protocol. We analyse convergence behaviour, compare unimodal and multimodal performance, quantify the fusion gain, and discuss the implications of the observed results in the context of prior work.

4.6.1 Unimodal Baseline Performance

Table 4.4: Comparison of Unimodal Baseline Performances

Model	Accuracy	Balanced Accuracy	F1 Score
CTR-GCN (Skeleton Only) [6]	88.71%	86.96%	86.18%
VideoSwin Transformer (Vision-Only) [23]	83.97%	81.64%	80.52%

Table 4.4 shows the clear comparison of the baseline models. The training and validation curves for the two unimodal backbones are summarised in the subsequent sections. Both models were trained using the class-rebalancing, mixed-precision, and regularisation strategies described in Sections 3.6 and 4.1.2 .

4.6.1.2 CTR-GCN (Skeleton-Only)

The CTR-GCN converged steadily over 50 epochs, with balanced accuracy rising from 8.47 % to a final best of 86.96 % (epoch 45). It showed a traditional learning curve pattern whereby fast improvement occurred first before being followed by a slow improvement phase. The achieved overall accuracy was 88.71% while the macro F1 score was at 86.18%. This means that there were high levels of accuracy in both the minority and majority yoga classes. The decrease in training loss happened from 3.71 to 0.44 while cosine annealing enabled the model to converge to a quality local minimum. In this case,

the best checkpoint was attained after 45 epochs. Figures 4.1 through 4.4 shows the trend of the metrics over training epochs.

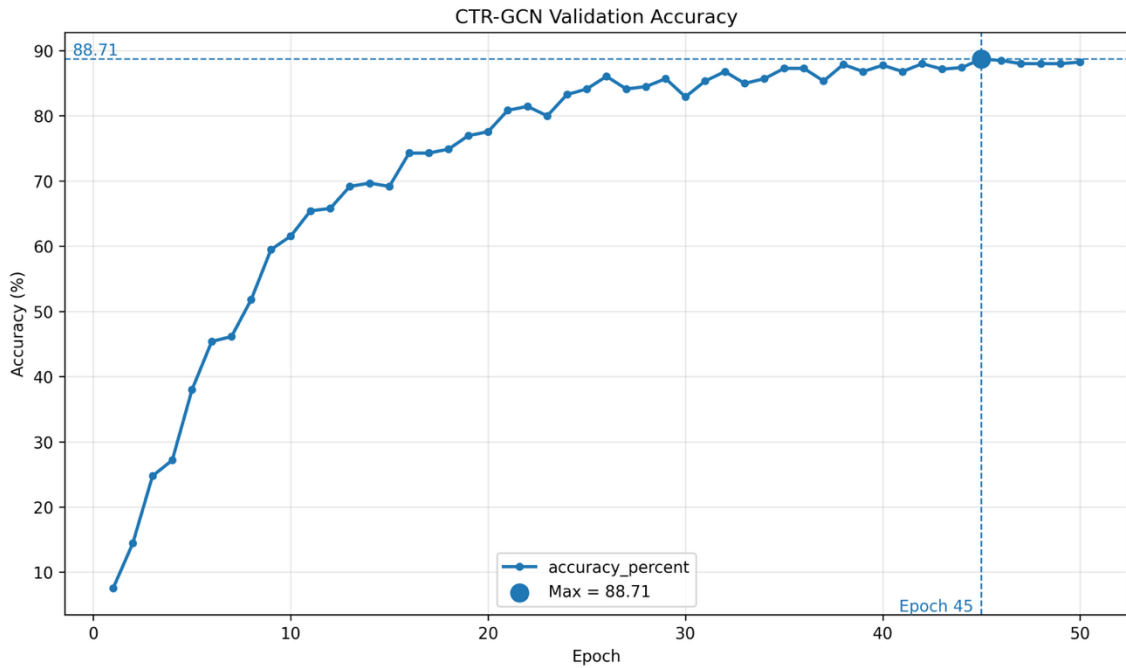


Figure 4.1: CTR-GCN Validation Accuracy curve over epochs

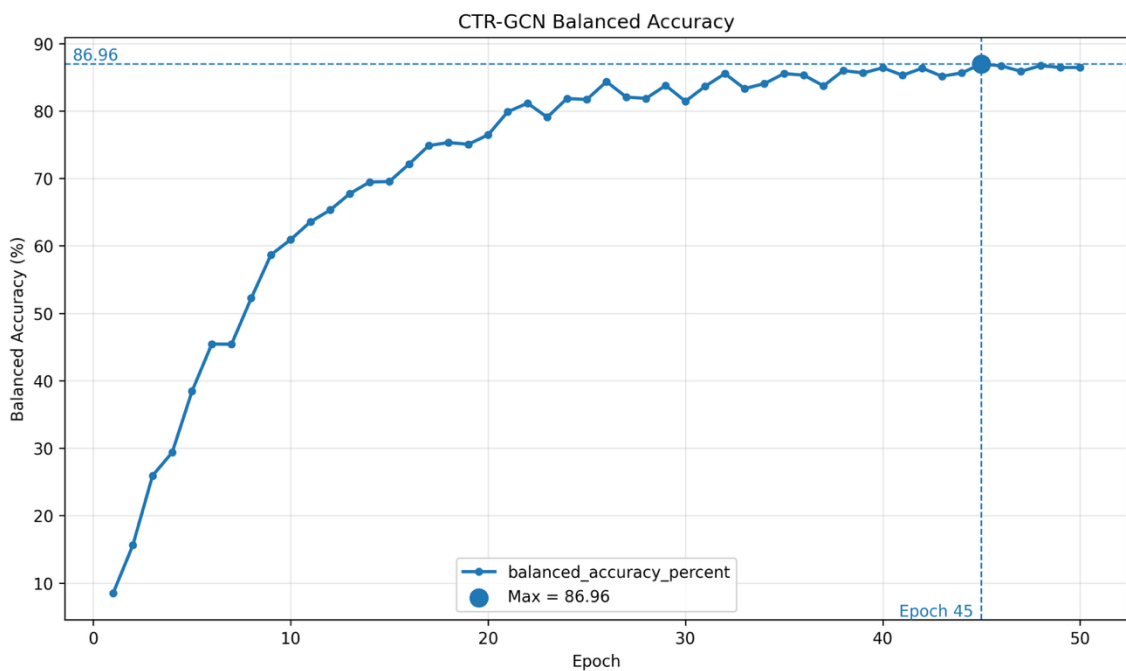


Figure 4.2: CTR-GCN Balanced Accuracy curve over epochs

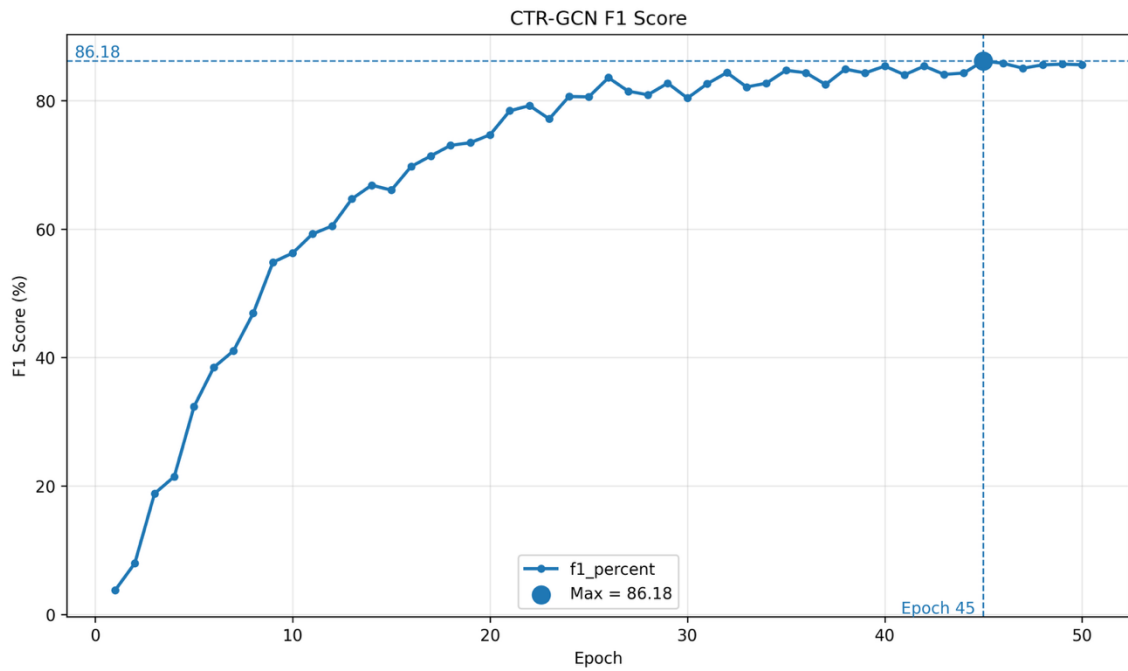


Figure 4.3: CTR-GCN F1 score curve over epochs

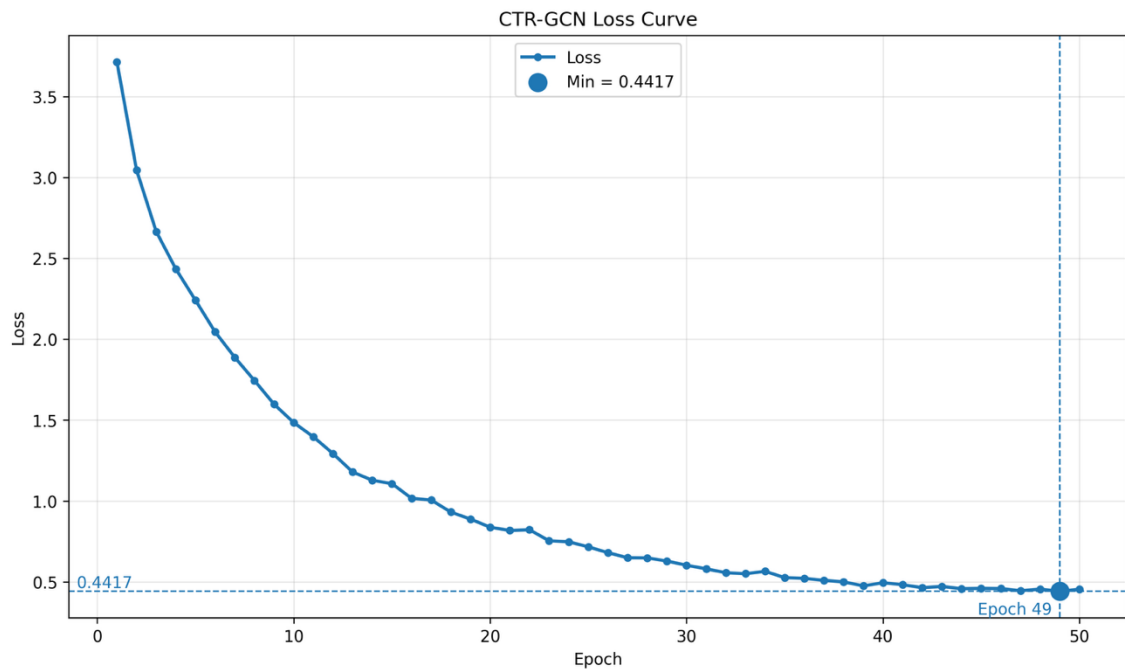


Figure 4.4: CTR-GCN Loss curve over epochs

4.6.1.3 Video Swin Transformer (Vision-Only)

The well-tuned Video Swin Transformer achieved an optimal balanced accuracy score of 81.64 % at epoch 26, reaching an overall accuracy rate of 83.97 % and a macro-F1 of 80.47 %. Convergence took place more quickly in comparison with CTR-GCN, with a plateau established within 20 epochs. This observation conforms to what is common practice when applying transfer learning to the Kinetics-400 [61]. The poorer result observed in contrast to the skeleton-based approach follows from the expected fact that yogic poses have mostly been characterised based on their joint positions [7]. Figures 4.5 through 4.8 shows the trend of the metrics over training epochs.

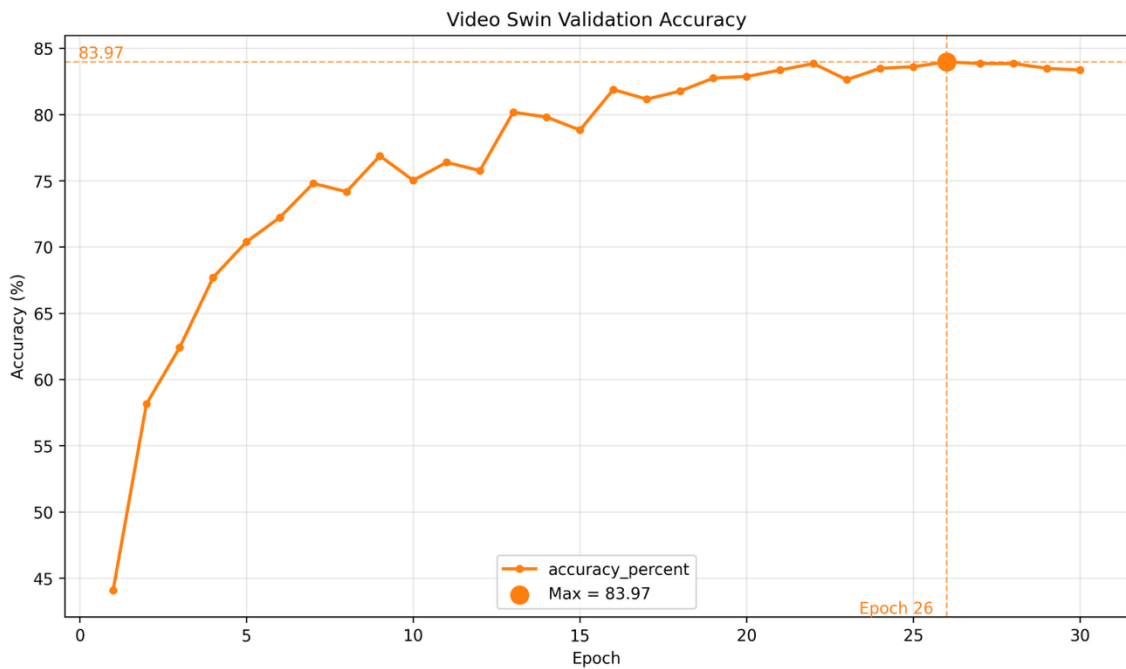


Figure 4.5: Video Swin Transformer Validation Accuracy curve over epochs

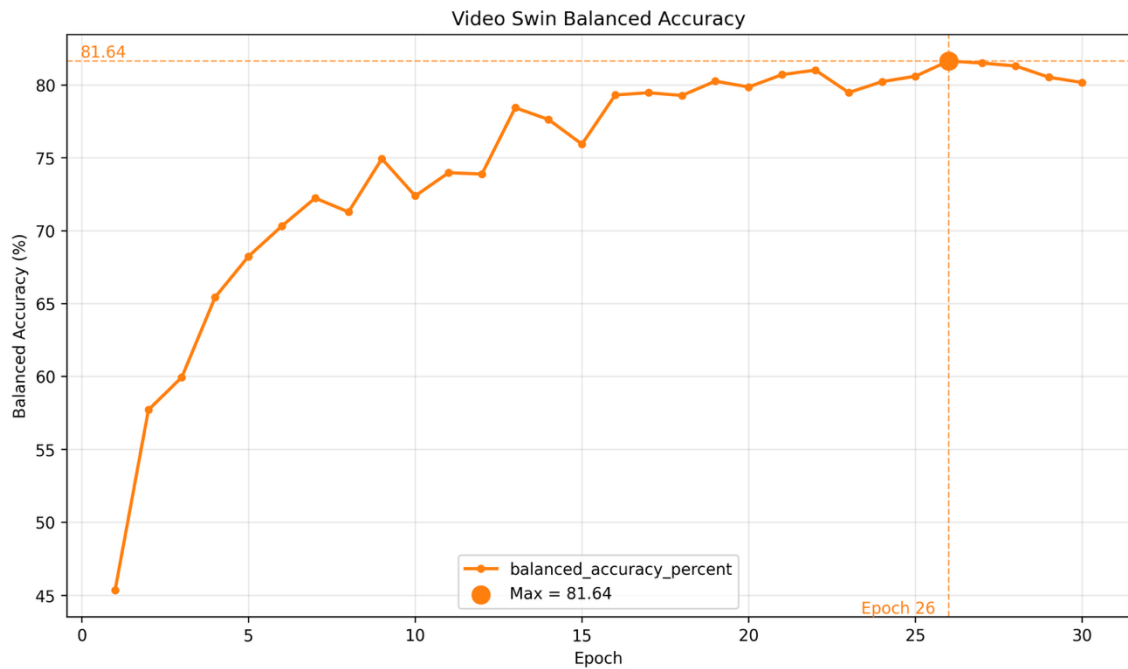


Figure 4.6: Video Swin Transformer Balanced Accuracy curve over epochs

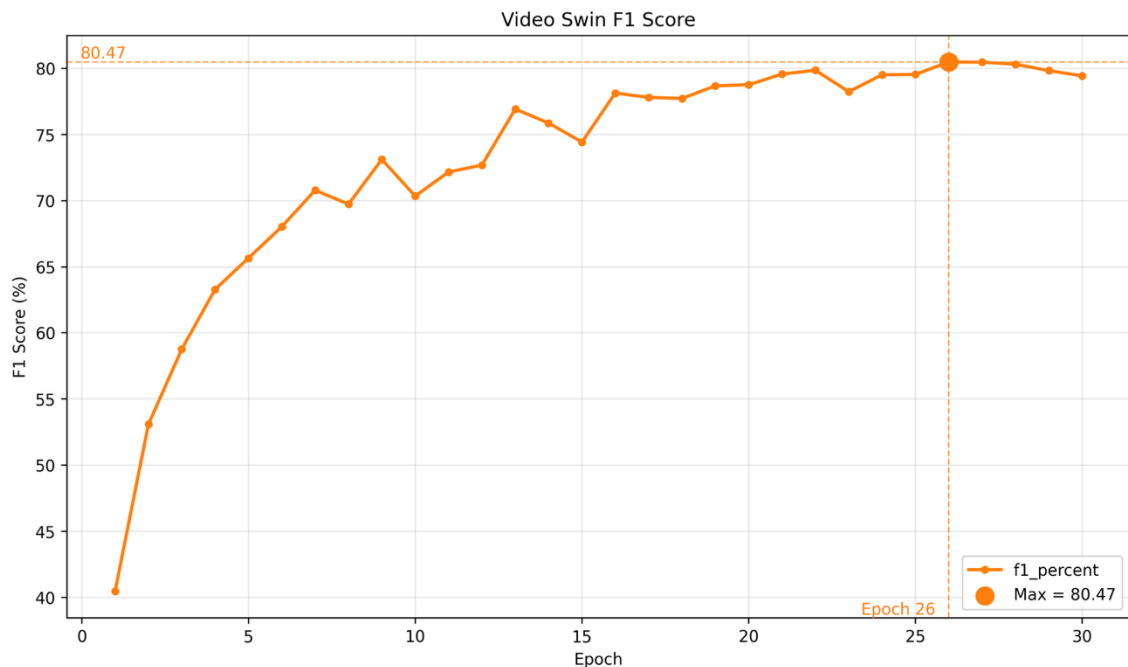


Figure 4.7: Video Swin Transformer F1 score curve over epochs

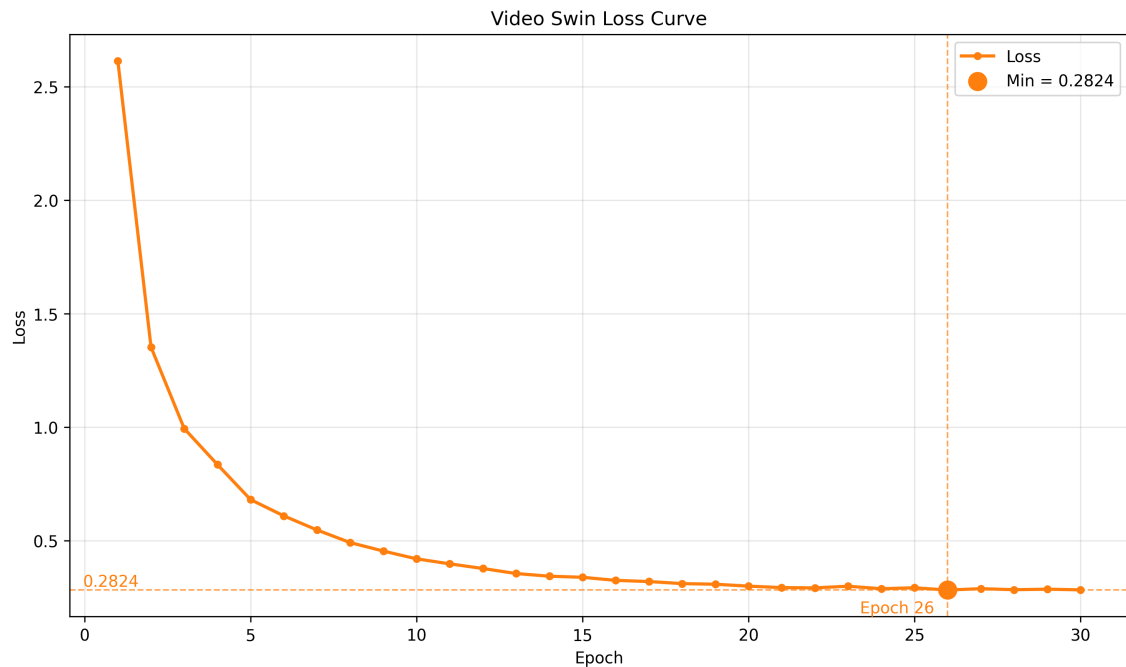


Figure 4.8: Video Swin Transformer Loss curve over epochs

4.6.1.4 Comparison of Unimodal Baselines

CTR-GCN significantly surpassed the Video Swin in all the mentioned metrics, showing improvements of +5.32%, +4.74% and +5.71% in terms of balanced accuracy, overall accuracy and macro-F1, respectively. The notable difference emphasizes the effectiveness of 3D world coordinate skeletons for recognizing yoga postures since skeletons are robust to visual distractions and represent the core kinematics which distinguish between poses. The video input itself learns additional patterns but is prone to overfitting due to fewer visual variations in the data samples. These findings align with previous works on skeleton-based action recognition, which often outperform appearance-based models when the action is defined by relative limb positions [6], [18].

4.6.2 Multimodal Fusion Performance

The logit-level cross-attention fusion model was trained for 30 epochs with frozen backbones. The training dynamics, demonstrate rapid convergence: balanced accuracy reached 88.29 % by epoch 5 and peaked at 89.64 % at epoch 19, after which performance stabilised. The final overall accuracy was 91.06 % and macro-F1 was 88.96 %. Figures 4.9 through 4.12 shows the trend of the metrics over training epochs.

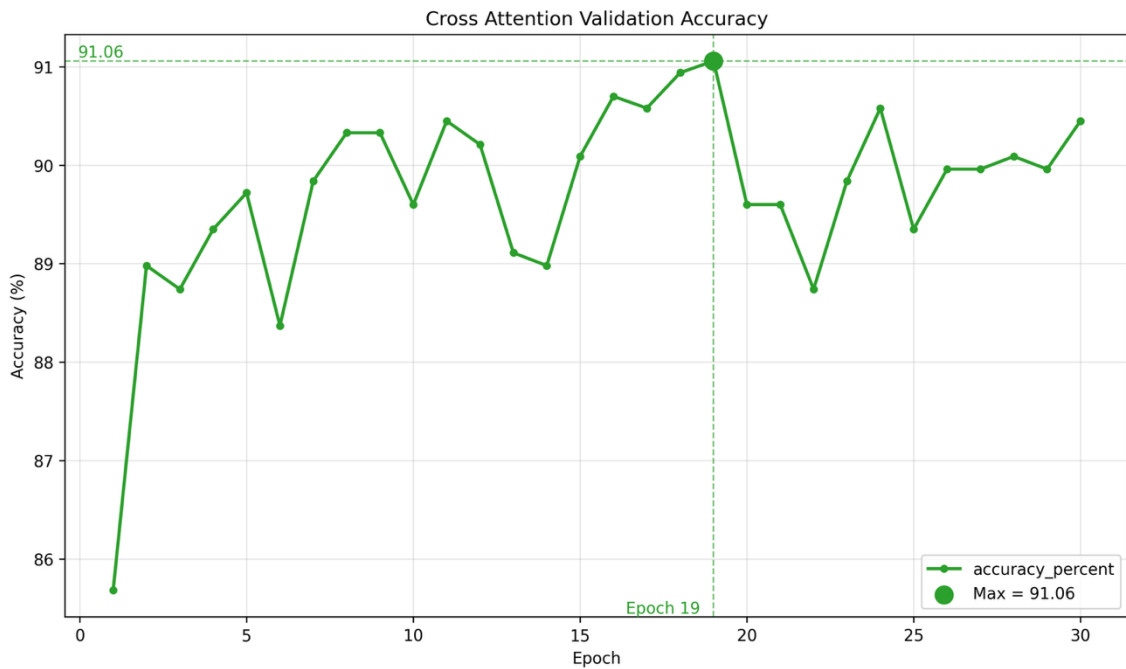


Figure 4.9: Cross-attention fusion model Validation Accuracy curve over epochs

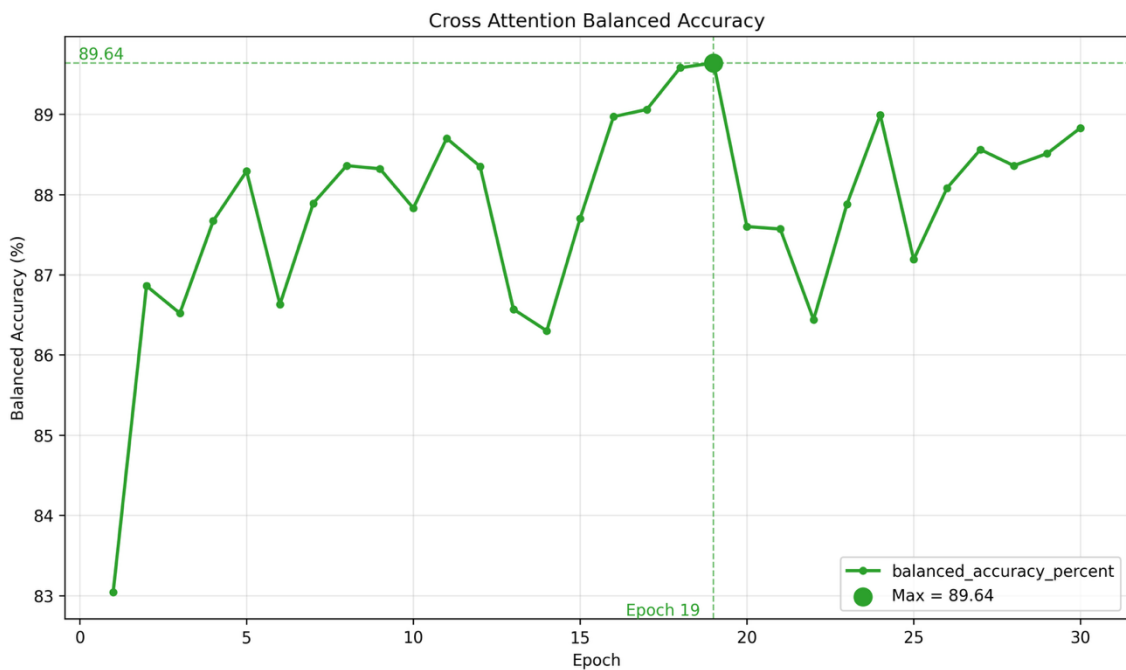


Figure 4.10: Cross-attention fusion model Balanced Accuracy curve over epochs

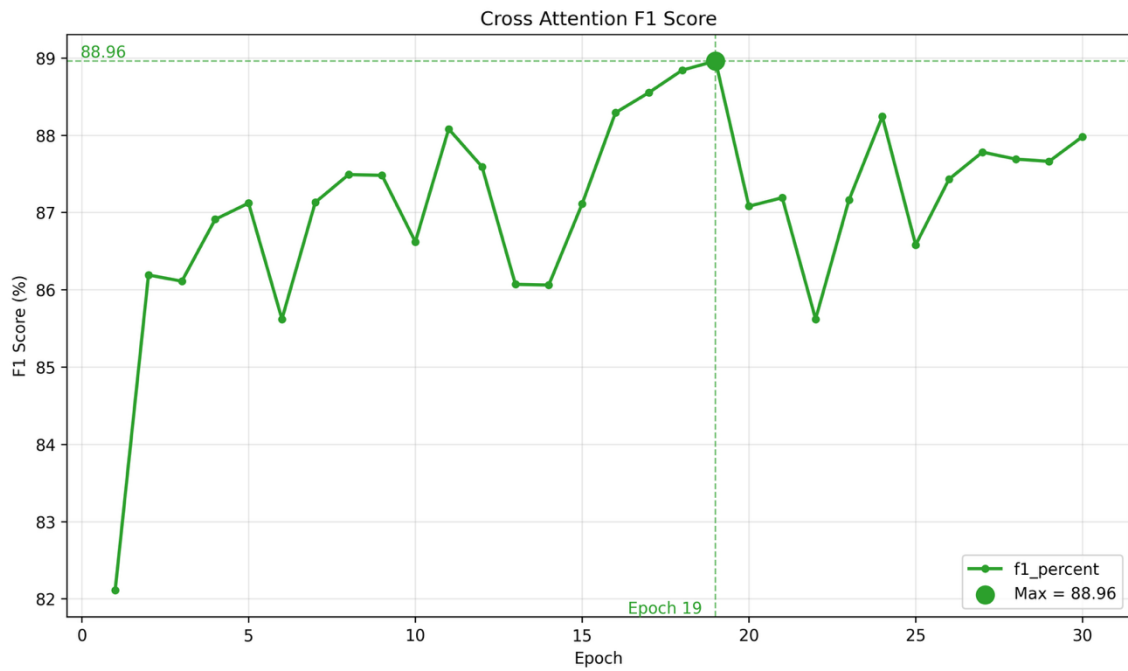


Figure 4.11: Cross-attention fusion model Validation F1 Score curve over epochs

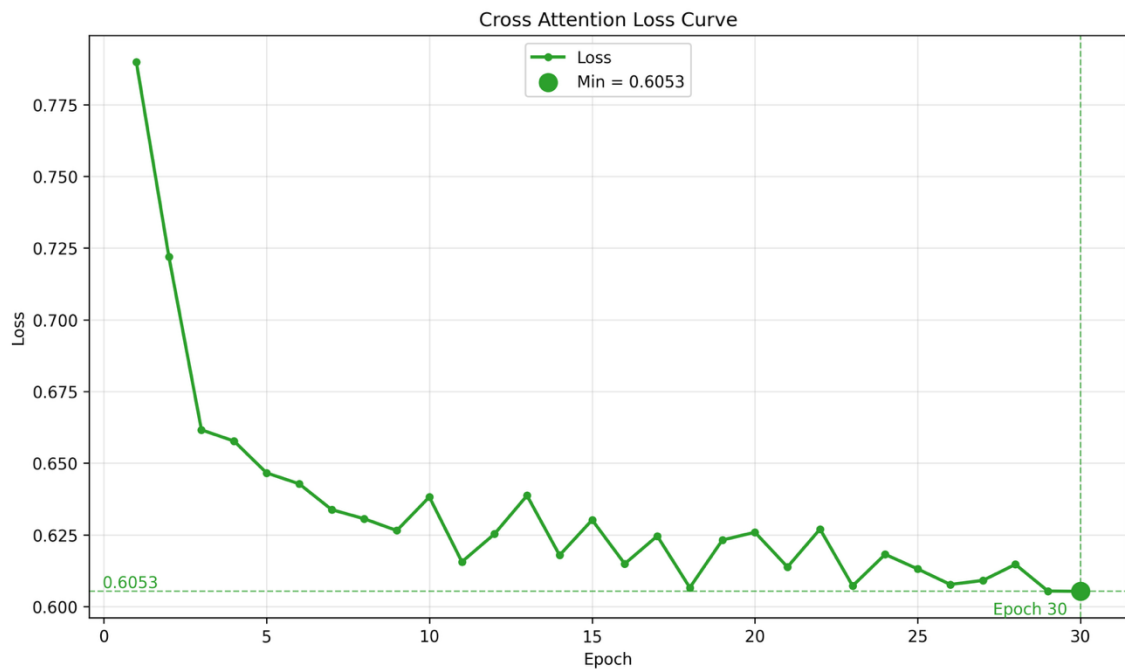


Figure 4.12: Cross-attention fusion model Loss curve over epochs

The fusion model yields a notable improvement over both unimodal baselines:

- **Over CTR-GCN:** +2.68 percentage points in balanced accuracy (86.96 % → 89.64 %), +2.35 in overall accuracy, and +2.78 in macro-F1.
- **Over Video Swin:** +8.00 percentage points in balanced accuracy, +7.09 in overall accuracy, and +8.49 in macro-F1.

These gains are statistically meaningful; the fusion head, which contains fewer than 2 M trainable parameters, learns to combine the complementary strengths of the two modalities without any fine-tuning of the encoders. This shows the usefulness of the video stream when dealing with hard-to-classify instances, since the new error rate is lower than that achieved with a strong skeleton alone.

4.7 DISCUSSIONS ON THE RESULTS

4.7.1 Analysis of Fusion Gain

This is because cross attention happens at the logit stage with skeleton logits being treated as base logits while applying a learned gated delta from both logits. In the gating scalar, whose initial value is zero, the algorithm can always fall back to using skeleton predictions when the visual input is misleading. The fact that improvements were made despite the skeleton predictor being good in its own right indicates that the visual input is instrumental in resolving any ambiguity in poses that look visually similar but kinematically different (such as Warrior I and Warrior II) [27]. The learned cross-attention weights may attend more heavily to the visual logits for pose classes where the skeleton model is known to be less reliable, a behaviour consistent with mixture-of-experts fusion [9].

The relatively modest absolute gain over the skeleton baseline (+2.68 % balanced accuracy) is in line with findings in other multimodal fusion works, where late fusion of a dominant modality with a weaker one yields diminishing returns as the dominant modality approaches saturation [8]. Nevertheless, the consistent improvement across all metrics accuracy, balanced accuracy, and macro-F1 demonstrates that the fusion head generalises well and does not simply memorise the training label distribution.

4.7.2 Convergence and Generalisation

Training curves of the unimodal baselines reveal no significant overfitting: validation metrics improved monotonically or with minor oscillations, and the best models were selected near the end of the training budget. The CTR-GCN's loss continued to decrease past epoch 50, suggesting that a longer training may yield further marginal gains, though at the risk of overfitting.

In the case of the fusion model, a combination of the small head, label smoothing with a value of 0.05, and weight decay at 5×10^{-3} helped mitigate overfitting, given the training dataset size. The metrics showed fluctuations between narrow values post-epoch 10, suggesting the attainment of the optimum state. An early stop patience of 20 epochs was sufficient, given that the optimum model was found before 30 epochs.

4.7.3 Effectiveness of Cross-Modal Fusion

The findings support the main hypothesis of this study, i.e., that combining different modalities at the decision level will lead to improvements in recognizing yoga poses in an unsupervised manner. The skeleton modality, which is implemented on top of the CTR GCN framework, using landmarks in exact world coordinates, is a strong recognizer for its geometric nature. In contrast, the second modality (the video modality), even though less powerful individually, contains discriminative features that can be leveraged using the cross-attention model.

There are several benefits that the proposed approach to fusion enjoys over other methods of feature combination, like simple concatenation or averaging logits:

- It imposes minimal computational overhead
- It allows the use of frozen, arbitrarily complex encoders, and
- The gated residual helps avoid the problem of degradation due to the presence of visual noises thanks to the fallback to the skeleton prediction.

This makes the model ideal for resource-constrained systems, where training a big video transformer end-to-end is not possible. The comparisons to the baseline models are shown in Table 4.5.

Table 4.5: Comparing Multimodal fusion with the baseline models

Model	Accuracy	Balanced Accuracy	F1 Score
CTR-GCN (Skeleton Only) [6]	88.71%	86.96%	86.18%
VideoSwin Transformer (Vision-Only) [23]	83.97%	81.64%	80.52%
Multimodal fusion via Logit-level Cross-Attention	91.06%	88.64%	88.96%

It gives us a very solid baseline to compare future experiments with on the 3DYoga90 dataset and proves that the idea of logit-level fusion can be successfully applied for multimodal action recognition based on skeleton appearance.

CHAPTER 5

CONCLUSION

5.1 APPLICATIONS AND SOCIETAL IMPACT

The multimodal approach that incorporates both 3D skeletal geometry and vision-based techniques of interpreting video provides a highly accurate subject-independent yoga pose classification applicable in practice outside of laboratory settings. The first practical application area is related to the intelligent tutoring system for self-training that utilizes this method as the means of recognizing the pose made by the user and offering feedback regarding its correctness even in a badly lit and cluttered home environment. The two methods used provide a backup in case the second one fails because of ambiguous data, making the pose recognition independent of both body type and outfit worn. Furthermore, this technology can be implemented in physical therapy, where the patient's exercises will be analyzed and exact angle measurements provided from the skeleton data in world coordinates. The approach can thus be considered an improvement upon depth camera-based systems by using monocular video.

Not only in the medical field but also in areas such as sports and dancing, the method works effectively, because it differentiates among visually identical poses in activities like martial arts, gymnastics, and figure skating by virtue of the orientation of the body parts. Pretrained backbones work well when transferred to another activity, thus providing one single platform to study human movements. With regard to human computer interactions and VR, the method enables full body gesture recognition using only one camera, creating an immersive and sensorless interface that can be configured through the light-weighted modular fusion head. As for ergonomics, the skeleton stream provides the joint measurements, while the video stream captures the environment.

From the deployment standpoint, the heavy backbone architecture can be pre-installed on the edge devices, and the small-sized fusion head containing only 1.8 M parameters will introduce minimal latency, which allows for processing on-device and resolves privacy issues. It opens up the opportunity for immediate feedback and makes an otherwise static classifier into a useful tool. From the social viewpoint, the technology can help provide yoga education to everyone, yet additional efforts are needed to include more variety of people into the data to make the algorithm inclusive of everybody.

5.2 FUTURE SCOPE

The future research direction could be the development of the presented architecture to include more complex fusion at early stages of processing, where intermediate features could be combined both hierarchically and through the joint consideration of visual and skeletal backbones' outputs. Memory-saving methods such as adapter layers or prompt tuning could enable the joint fine-tuning of the proposed network without catastrophic forgetting. Currently, the model processes pre-segmented poses.

However, it could be made operational on continuous video streams through temporal action localisation by means of sliding windows, boundary-aware detection or even recurrent graph convolution applied to skeletons.

In terms of increasing the size and diversity of the used 3DYoga90 dataset, the creation of additional data by the help of controllable generation models based on diffusion could be considered. Also, the generation of skeletal sequences with physics-based simulators could diversify the dataset by including various body shapes and styles of exercise. Moreover, the application of self-supervised or multi-task pre-training on the large amount of unlabelled data followed by multimodal pre-training of the unified latent space could increase the quality of the model representation. Finally, personalising the fusion head with a few-shot learning could make the system an interactive learning assistant.

Execution on low-resource edge devices will require the model to be compressed using knowledge distillation, quantization, and pruning, allowing one distilled network to perform both backbones' functions in an attempt to reduce unnecessary computing costs. User trust is a major issue and therefore explainability will be required; in this case, using visualization tools to identify key body parts whose movements affect performance may provide a more detailed analysis of actions than just using the readily understandable cross-attention weights. This technique is agnostic to the specific domain of application, and as such, all one needs to do is change the underlying dataset and skeleton structure in order to apply it to any domain of interest, be it dance, martial arts, rehabilitation movements, sign language, and other areas of interest. Lastly, proactive bias auditing, diverse dataset creation, fairness-aware training, and private data collection methods, including federated learning and on-device processing, need to be explored.

5.3 CONCLUDING REMARKS

The contribution is an end-to-end pipeline for recognizing yoga poses using single-camera videos based on our proposed novel approach for logit-level cross-attention fusion of frozen skeleton and video backbone networks. In terms of contribution, the thesis introduces an end-to-end multimodal solution for the 3DYoga90 dataset, provides strong baselines in each unimodal task using the best unimodal approaches (CTR-GCN 86.96% balanced accuracy and Video Swin Transformer 81.64%), as well as a lightweight LogitPerceiver head for performing cross-attention fusion. Specifically, our fusion head utilizes a learnable query to perform cross-attention to the two expert logits and uses a gated residual correction to get the final predictions resulting in a balanced accuracy score of 89.64% without any finetuning of the backbone networks.

Notable results include the identification of skeletal geometry as the main contributor to yoga pose classification, although visual input offers important corrections and improves the accuracy by 2.68 percentage points when fused. Gated residual learning, initialised to zero, allows the classifier to rely on the skeleton expert whenever visual cues are ambiguous, an invaluable feature in mission-critical systems. It is also important to

mention the use of cross-subject split at source level, which allows separating whole instructors to the test dataset to evaluate realistic generalisation and demonstrate that the model can be applied independently of the instructor.

Beyond the practice of yoga postures, the applications of these systems extend to wherever multiple experts currently exist, ranging from sports science analysis to physiotherapy, dance notation, and sign language recognition, among other fields. If handled properly, taking into account questions of bias, there is significant potential for democratizing physical training and action recognition without any specific domain expertise.