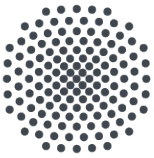


Universität Stuttgart

Gaze-based Transformer for Improving Action Recognition in Egocentric Videos

Master Thesis

Mengze Lu

15.01.2024 - 15.07.2024

Supervisor: Dr. Lei Shi

Second Examiner: Prof. Dr. Andreas Bulling

Prof. Dr.-Ing. Kai Peter Birke
Fachgebiet Elektrische Energiespeichersysteme (EES)
Institut für Photovoltaik
Pfaffenwaldring 47
D-70569 Stuttgart

Contents

Abstract

This thesis investigates the integration of gaze information into a transformer-based model to enhance action recognition in egocentric videos. The study focuses on enhancing the dynamic class token generator module with additional gaze information. The proposed Gaze-Enhanced dynamic class token generator (DCTG) module fuses gaze features with hand-object features to create comprehensive gaze-hand-object features. These features generate a specialized class token that, along with the video frames, guides the transformer to focus on action-related segments. The proposed model maintains a similar architecture to the original EgoViT, handing the temporal relationships between short-term phases effectively. A modified Video Swin Transformer serves as the backbone for processing local spatial relationships.

The EGTEA Gaze+ dataset is used for model training and testing. A series of experiments evaluates the impact of gaze information on EAR accuracy. The key objectives are to propose a novel Gaze Extrator module, which generates gaze-box image from input gaze points, and then fed them into a swquential convolutional networks to obtain the gaze features. Results demonstrate that the Gaze-Enhanced EgoViT model achieves a top-1 accuracy of 52.0% and a top-5 accuracy of 76.3%, surpassing the baseline EgoViT model. Higher quality gaze features significantly improve performance, with experiments showing that gaze information alone yields a top-1 accuracy above 50%. These findings highlight the potential of the Gaze-Enhanced EgoViT model to advance action recognition in egocentric videos and suggest further exploration into the role of gaze information in enhancing EAR accuracy.

Chapter 1

Introduction

The chapter presents a comprehensive overview of the study. It begins by introducing the research background and motivation, followed by a discussion of the objectives and limitations. The chapter concludes with an outline of the remaining structure of the thesis.

1.1 Background of The Study

Action Recognition is a computer vision task, which aims to identify and categorize the actions performed by human in video sequences. The task focus on analyzing the spatiotemporal dynamics of the actions and mapping them to predefined classes. The data is commonly collect as images and videos.

Previous researches have demonstrated significant success of neural networks in action recognition, with deep convolutional neural networks having long dominated visual modeling in computer vision [1], [2]. However, with the advent and success of transformer, a new architecture of neural networks, many researchers now consider transformers to be a promising solution for understanding actions in images and videos.

Recent research [3] introduced the Vision Transformer (ViT), which achieves excellent results in image classification task using a purely transformer-based architecture. Inspired by ViT's success in image applications, researchers explored transformer-based architectures for video recognition problems. In 2021, Arnab et al. [4] presented a pure-transformer based models for video classification. Bertasius et al. [5] proposed TimeSformer, which adapts the standard transformer architecture to learn spatiotemporal feature of a video. Liu et al. introduced Swin Transformer

and Video Swin Transformer for image and video classification using shifted Windows [6], [7]. These studies indicate that the Multi-Head Self-Attention mechanism in transformers not only has the ability to understand text but can also effectively interpret images and videos.

However, these studies primarily focused on standard videos. With advancements in hardware, many videos are now captured by wearable devices. Some applications in Virtual Reality (VR) and Human-Robot Interaction (HRI) require identifying actions in real-time video. These videos have different characteristics from standard videos, making action recognition more challenging. The video captured from a wearable camera provides a first-person perspective, also known as egocentric video. Unlike standard video, the camera in egocentric video is not stationary and depends on the wearer's viewpoint. It captures not only objects but also the interactions occurring between objects and subjects, causing frequent and rapid motions in the scene. This presents a challenging problem for egocentric action recognition (EAR).

The hand plays a crucial role in human interactions with the world [8]. Furthermore, the hand and its interactive object occupy a large percentage of the egocentric video frames. This is a significant difference between standard and egocentric video. Therefore, recognizing the hand and the objects it interacts with is essential for EAR.

In a recent study, Pan et al. introduced the EgoViT, a model designed to consider the special properties of egocentric videos, which can seamlessly integrated with different video transformers [9]. EgoViT integrates the features of hand and the object it interacts with, forcing the model to understand egocentric video by focusing on hand-object interactions. And the experiments outlined in [9] demonstrate that hand-object information proves to be more valuable for EAR. The major contributions of Pan et al.'s works include the incorporation of DCTG and the Pyramid Architecture with a Dynamic Merging (PADM) module into different transformers. The DCTG will detect hands and objects in video frames, then the extracted hand-object features will be used as class tokens sent to transformers. This module forces the transformer to focus on specific features from the original video. Since DCTG can generate the class token from given features, and as mentioned in [9], it has the potential to process features from videos beyond just hand-object interactions. Future studies could explore how other features impact the accuracy of action recognition.

A possible additional feature in egocentric videos is gaze information. Understanding visual attention is significantly valuable across various applications. A

growing number of devices in applications like VR and HRI are capable to record the gaze data from user. Research by Hayhoe et al. [10] indicated eye movements are crucial for understanding human intention in daily activities. Another study by Land et al. [11] demonstrated that in object-related actions, the direction of gaze is closely connected with the specific act. These studies provide a theoretical basis for the importance of visual information in egocentric videos. This perspective suggests that integrating gaze information has the potential to enhance the accuracy of EAR.

Furthermore, several studies have examined gaze information in egocentric video. Huang et al. [12] developed a computational model for predicting the camera wearer's point-of-gaze from egocentric video. In [13] explored bottom-up and top-down attentional cues involved in guiding first-person gaze. Lai et al. [14] introduced a transformer based model that calculates spatiotemporal global-local correlation for egocentric gaze estimation. Despite interest in gaze information in egocentric video, none of these studies have used gaze data collected from capture devices in their models.

The commonly used large-scale dataset for human action recognition in standard videos include Kinetics-400 and Something-Something v2. EPIC-KITCHENS-100 is a widely used dataset for egocentric videos, but the datasets mentioned above lake of gaze data. Several newly collected first-person view datasets, such as GTEA Gaze+ [15] and Ego4D [16], including the gaze information at the frame level. The gaze point of the person is recorded simultaneously during video capture.

Although many studies have hinted at the impotence of gaze information in EAR, there is a limited body of research that specifically addresses gaze information within transformer based models. As highlighted in [9], EgoViT, specifically the DCTG module, has the potential to integrate additional information, including gaze data.

In this thesis, the gaze points collected from the dataset will be utilized to explore their impact on the accuracy of EAR in a transformer-based model. The proposed model will build upon the hierarchical architecture of EgoViT, using Video Swin Transformer as the backbone. The EGTEA Gaze+ dataset will be employed for this study.

1.2 Objectives

The goal of this thesis is to study the impact of additional gaze information on a transformer-based model in EAR. The architecture of EgoViT is specifically designed for egocentric video, with its PADM module could effectively be processing the sequential phases in the video. With its ability to better understand rapid scene changes in videos, EgoViT has significant potential for this task. Therefore, the proposed model in this thesis is base on the EgoViT framework.

The key component of EgoViT is the DCTG module, which includes a Hand and Object Detector (HOD). The HOD extracts hand-object features from the detected hand-object parts in a frame. This thesis aims to enhance the DCTG module by integrating additional gaze information. The original DCTG in EgoViT will be extended to fuse gaze information, processed from gaze points, with hand-object information from HOD. By combining gaze, hand, and object data, the model will extract gaze-hand-object features. Finally, a class token will be generated from these gaze-hand-object features. This special class token, along with the video frames, will be sent to the subsequent layers in the transformer. This process helps the transformer-based model concentrate on the action-related parts of the video.

EGTEA Gaze+ dataset is used for model training and testing. A series of experiments will be conducted to study how gaze information affect the accuracy of EAR. Consequently, the objectives of this study are as follows:

- Propose a transformer-based method for EAR, notable for its novel incorporation of gaze information. The model incorporates both hand-object interactions and gaze information simultaneously.
- Enhanced the DCTG with additional gaze information. The Gaze-Enhanced DCTG module include a gaze-box cropper and sequential convolutinal networks to extract gaze features. This enhancement will guide the transformer to concentrate more effectively on the most informative segments of the video.
- Explore how to combine the class token and normal tokens in the transformer to improve the results of EAR.

1.3 Thesis Structure

With the completion of the first chapter, the following paragraphs provide a comprehensive overview of the subsequent chapters.

Chapter 2: Related Works reviews the existing literature and studies related to the research topic, providing an overview of transformer-based model in egocentric video.

Chapter 3: Methodology outlines the structure of the proposed model and the procedures employed in the study, detailing the approach, data collection, and key component explanations.

Chapter 4: Experiments and Results describes the practical implementation of the proposed model, including the experiments and results obtained from the EGTEA Gaze+ dataset. At end of this chapter, the results will be discussed.

Chapter 5: Summery summarizes the study, highlighting the key findings and contributions. This chapter also discusses the potential applications of the proposed model in battery production.

Chapter 2

Related Works

This study builds upon several previous works. A brief presentation of the relevant literature will aid in understanding the contributions of this thesis. This chapter provides an overview of the researches on egocentric action recognition as well as the theoretical framework of transformer-based models.

2.1 Action Recognition

In many real-world applications, including gaming, HRI, and behavior analysis, it is essential to understand human actions in images or videos. Action recognition involves identifying and localizing human behaviors. Recognizing actions in videos, particularly in real-time, is a challenging task in computer vision. Over the last decade, there has been a significant amount of research focused on action recognition using deep learning methods. Zhu et al. [17] summarized the development of deep learning methods in this area, as shown in Figure 2.1. Karpathy et al. [18] introduced the DeepVideo, which was the first attempt to use convolutional neural networks (CNNs) to address this problem. Following this, CNNs dominated the field of action recognition for a long time. The primary advantage of this model is its ability to operate directly on raw data without any hand-crafted feature extraction.

There are several challenges in developing effective video action recognition algorithms. First, some human actions are closely related and exhibit similar movement patterns, making it difficult for algorithms to distinguish between them. The second significant challenge is that the model needs to simultaneously understand both short-term information and long-term temporal information.

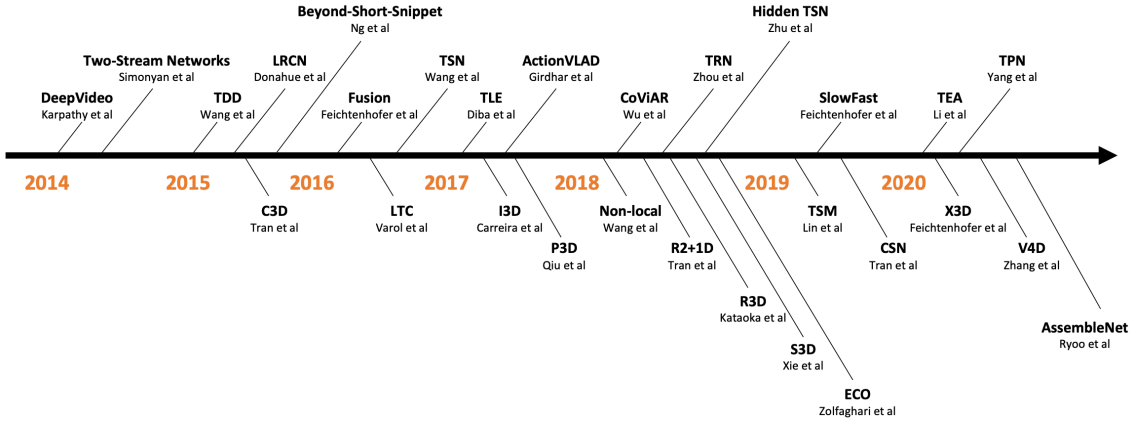


Figure 2.1: A chronological overview of representative work in video action recognition before 2020 [17].

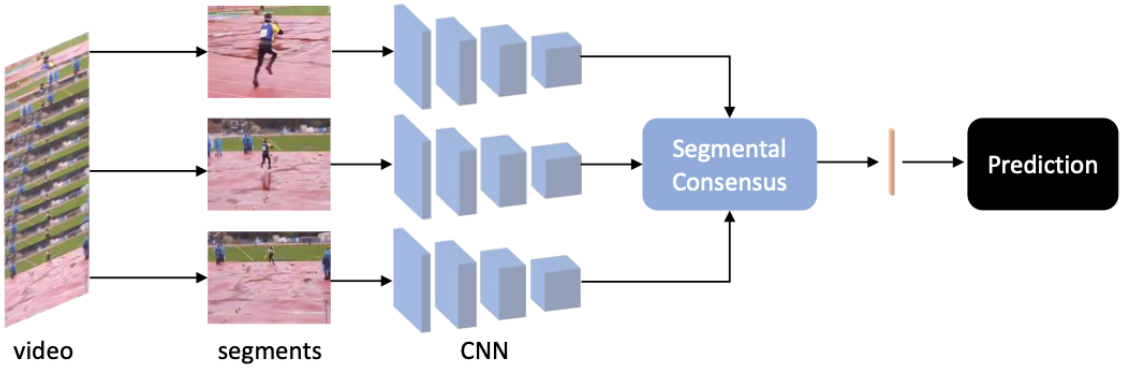


Figure 2.2: The structure of Temporal Segment Networks (TSN) [17]

Wang et al. [19] proposed the Temporal Segment Networks (TSN) to address the challenge of simultaneously understanding short-term and long-term information. The structure of TSN is a common example of CNNs-based models. Figure 2.2 shows a simplified structure of TSN. TSN performs video-level action recognition, by taking the entire video as input and segmenting it into several parts, uniformly distributed along the temporal dimension. TSN then randomly selects a frame within each segment and sends it to subsequent layers. Finally, the information from the sampled frame is aggregated in the Segmental Consensus module. This segmental structure allows TSN to observe content throughout the entire video. Many follow-up studies have handled short-term and long-term content using a similar strategy. Resently, Lin et al. [20] introduced the termed temporal shift module (TSM). The part of the channels will be shifted along the temporal dimension, thereby facilitating the exchange of information among neighboring frames.

2.2 Egocentric Action Recognition

Egocentric video, most captured by wearable cameras, are characterized by frequent and large camera movements, along with complex background scenes. These features present unique challenges for general video transformer models. Several high-quality egocentric video datasets have been collected in previous studies, including [21], [22] and [22]. Notably, datasets such as Ego4D [16] provide additional information like gaze, stereo and audio. This section will review relevant research in egocentric video action recognition. Herzi et al. [23] proposed an object-centric module for integration with video transformers. Wang et al. [24] propose Symbiotic Attention with Privileged information for EAR. Additionally, Huang et al. [25] collected a new egocentric video dataset and developed a graphical model for joint attention detection. Despite the progress made, previous research has often not fully addressed the specific challenges inherent in egocentric videos. To this end, Pan et al. [9] proposed a pyramid video transformer structure, showing promising results for egocentric video applications.

Other studies have focused on the role of gaze information in egocentric video. Huang et al. [12] incorporated temporal attention transition into a CNN-based saliency model for gaze estimation. Tavakoli et al. [13] explored both bottom-up and top-down attentional cues involved in first-person gaze guidance. Furthermore, Lai et al. [14] introduces a transformer-based model that explicitly embeds global context and calculates spatio-temporal global-local correlation for egocentric gaze estimation. In this thesis, our aim is to develop a transformer for egocentric video, building upon the EgoViT framework and incorporating DCTG with gaze information.

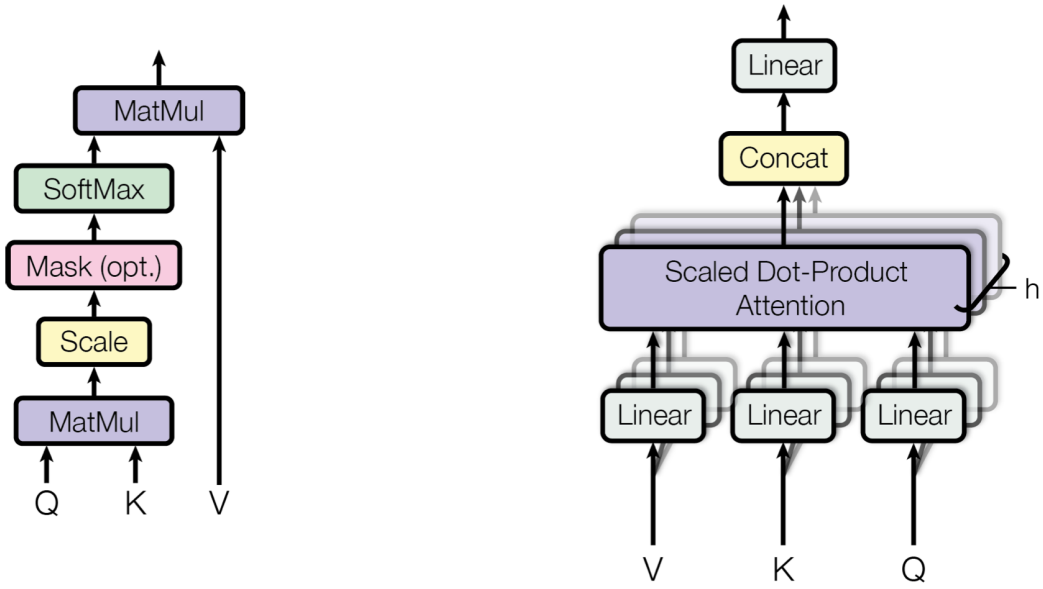
2.3 Transformer

Transformer is a new deep learning architecture first introduced in the paper "Attention is All You Need" by Vaswani et al. [26] in 2017. Originally used primarily for natural language processing (NLP) tasks, the Transformer architecture has demonstrated the ability to handle large amounts of text data effectively. Its breakthrough achievements have made Transformers a fundamental component in NLP. The key component of the Transformer is its Attention Mechanism. The attention function is defined as mapping a query and a set of key-value pairs to an output, with the query, keys, values, and output all being vectors [26]. Figure 2.3 visualizes the prin-

ciple of the attention mechanism. The scaled dot-product attention is calculated as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

Where Q is the query, K is keys, V is the values, and d_k is the dimension of the keys. Multiple single attention function with d_{model}-dimension keys, values and queries



(a) Scaled Dot-Product Attention (b) Multi-Head Attention

Figure 2.3: The schema of Scaled Dot-Product Attention (a) [26] and Multi-Head Attention (b) [26]

are applied, and their outputs are concatenated to form the Multi-Head Attention. Hence, the Multi-Head can be described as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.2}$$

Where $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

The transformer can be divided into two parts: the Encoder and the Decoder. Each part consists of stacked identical layers, a multi-head self-attention mechanism and a feed-forward neural network. The decoder additionally includes a masked multi-head self-attention mechanism, which ensures that predictions for a given position depend only on the known outputs at previous positions [26]. The structure of the transformer is shown in Figure 2.4. The self-attention mechanism has demonstrated its ability to learn the long-range dependencies. Building on the success of

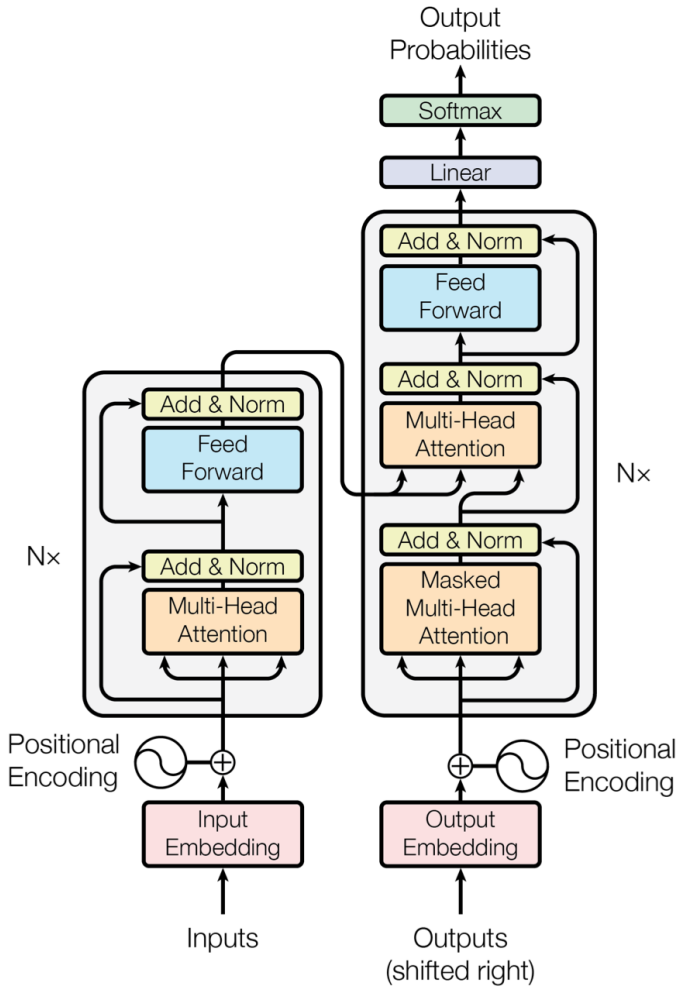


Figure 2.4: The structure of transformer [26].

transformers in NLP, several studies have explored combining CNNs-based architectures with self-attention [27], [28]. However, a major drawback of these models is that The global properties of self-attention may not be fully utilized during the integration of two mechanism. While they are effective at capturing local features and extracting spatial information in small areas through convolution kernels, they are less effective at capturing global information.

2.4 Transformers in Video Recognition

Dosovitskiy et al. [3] proposed the ViT for image recognition. ViT, which globally models spatial relationships on non-overlapping image patches with minimal modifications from standard transformer [26], achieves excellent results compared

to state-of-the-art convolutional networks. Their work has demonstrated that a pure transformer architecture is a promising solution for image recognition. Following the success of ViT in image classification, multiple works have tried to explore transformer-based architectures for video recognition. Extending these advancements to video data, which adds only a spatial domain compared to image data, researchers have adapted these transformative techniques on the transformer-based model. Recently, Arnab et al. [4] presented a pure-transformer based model for video classification. The Video Vision Transformer (ViViT) calculates a sequence of spatiotemporal tokens from input video, then processes them with self-attention. By employing several methods to factorize the model along spatial and temporal dimensions, ViViT is able to handle long-distance dependencies and complex spatiotemporal interactions effectively.

As a result, Video Transformers [5], [7], [23], [29], which inherit the advantages of image understanding from their predecessors, have shown remarkable success in video recognition benchmarks [30], [31]. Despite the excellent success video transformer have achieved, some video transformers like [4], [5], [29] suffer from high computation costs, primarily due to extending the image spatial domain into a global spatiotemporal domain. Another disadvantage is that the performance of these video transformers heavily relies on pretrained 2D spatial models on super large dataset JFT-300M [32]. The third challenge for video transformers is the efficiency of processing long-range dependencies. In practice, when processing long videos, important temporal information may be diluted or overshadowed by a large number of attention weights.

To address this issue, Liu et al. [7] attempted to overcome this limitation by broadening the focus of local attention computation. Instead of solely concentrating on the spatial domain, their approach encompasses both spatial and temporal domains. Pan et al. [9] emphasize the importance of both local and global temporal attention, especially for egocentric videos, which typically contain large and frequent movements. To effectively address these problems, they proposed a pyramid structure that successfully integrates both local and global temporal attentions. This structure provides an inductive bias on grouping both local temporal attentions and the global temporal attentions [9]. In section 2.6, the Video Swin Transformer will be discussed in detail, as it serves as the backbone of the proposed model in this thesis.

2.5 Object Detection-Oriented Action Recognition

A number of studies [33], [24], [34], [35] have demonstrated that models incorporating object detection and interaction features, particularly those focusing on object-human interactions, achieve considerable success in the field of video understanding. More recently, research has focus on video action recognition with egocentric video. Shvetsova et al. [36] proposed a modality-agnostic transformer that integrates information from various sources into a single multi-modal representation. Herzig et al. [23] introduced the Object-Dynamics Module, which achieved state-of-the-art performance in video action recognition. Drawing inspiration from Herzig et al. [23], Pan et al. [9] designed the first method to incorporate object-human interaction features into a transformer. This is achieved through a dynamic class token, embedding these features directly into the class token [9]. Their method addresses two key issues: firstly, how to incorporate object-subject interaction features, and secondly, how to embed locality inductive bias within the self-attention module.

2.6 Video Swin Transformer

The Video Swin Transformer builds on the Swin Transformer, extending the local attention computation from solely the spatial domain to the spatiotemporal domain. The overall architecture is shown in Figure 2.5. Video Swin Transformer consist of four stages, each containing a patch merging layer and multiple Video Swin Transformer Block. The transformer block will be repeated N times according to the model configuration. It does not down-sample along the temporal dimension, but performs $2\times$ spatial down-sampling in the patch merging layer of each stage. The input videos will be divided into several 3D patches. The patches are then processed through the stages in series. The output of the last stage is defined as the normal token, which is used for classification in subsequent head layers.

The Video Swin Transformer Block inherit the structure of standard transformer, only replacing the multi-Head Self-Attention (MSA) with the 3D shifted window based multi-head self-attention module. As shown in Figure 2.6, a Video Swin Transformer Block consists of a 3D shifted windows based MSA module, a feed-forward network (FFN), and two Layer Normalization (LN) layers before the 3D SW-MSA and FFN. Through the 3D shifted windows based MSA, the Video Swin

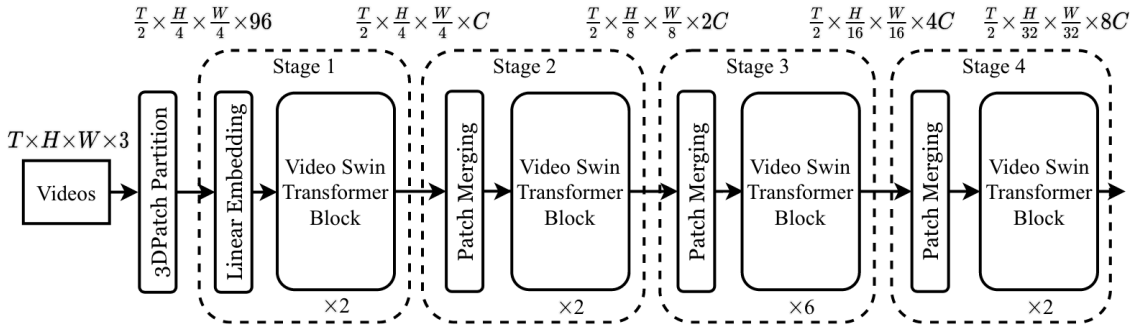


Figure 2.5: The architecture of Video Swin Transformer (Swin-T version) [7]. Transformer introduces a locality inductive bias to the self-attention module. The shifted windows allow the non-overlapping 3D windows to exchange information with each other. Figure 2.7 illustrate the mechanism of the 3D shifted windows based MSA module.

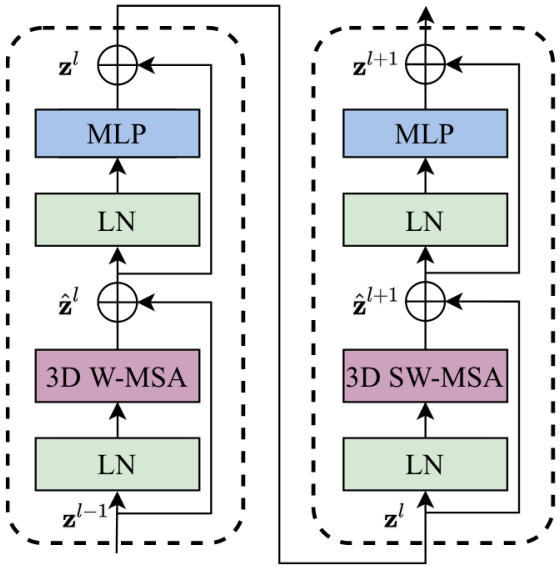


Figure 2.6: The structure of Video Swin Transformer Block [7].

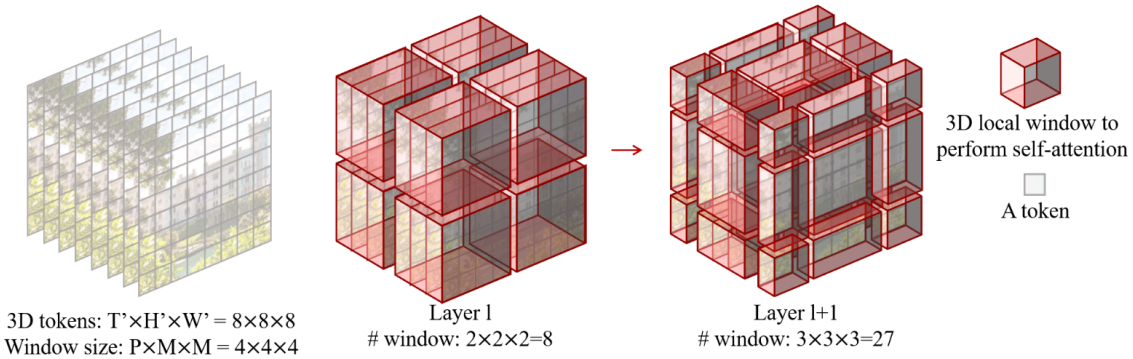


Figure 2.7: The mechanism of 3D shifted windows [7].

2.7 EgoViT

In this section, the structure and key components of EgoViT will be explained in detail. Since the proposed model in this thesis is strictly based on EgoViT, it is important to thoroughly understand the structure of EgoViT.

EgoViT features a hierarchical pyramid structure that effectively provides an inductive bias for grouping both local and global temporal attentions. This design enables it to successfully process the rapid motions typically found in egocentric videos. The overall architecture of EgoViT is shown in Figure 2.8. EgoViT consists of three main parts: the Short-term Stage, Intermediate Connection and Long-term Stage.

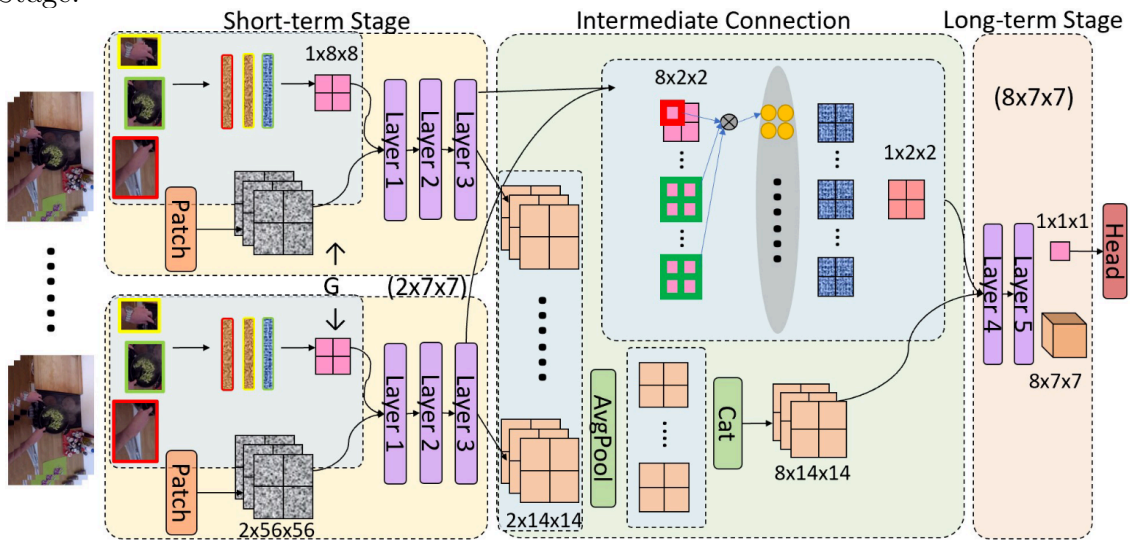


Figure 2.8: The architecture of EgoViT with Dynamic Class Token Generator[9].

The Short-term Stage is designed to capture the local temporal information. It is divided into G groups, and each group integrates a DCTG module. The DCTG contains a pretrained HOD module, which detects the hands and objects interacting within the frames. The detected hands and objects are transformed as a combined hand-object features. These combined hand-object features are then processed in subsequent layers to generate the Dynamic Class Token (DCT). The DCTs are concatenated with the embedded frames for the following transformer layers. After processing in these layers, the G DCTs provide a summary of the semantic meaning of each short video phase. The processed DCTs from each group are then merged in the Intermediate Connection. During this merging process, a larger weight is assigned to the DCTs that represent crucial short-term actions. The merging process can be expressed in Eq. 2.3

$$\begin{aligned}
\alpha_{g,s,g'} &= \frac{1}{\|\mathbf{x}_{g,s}^{cls}\|} \sum_{s'} \frac{\mathbf{x}_{g,s}^{cls} \cdot \mathbf{x}_{g',s'}^{cls}}{\|\mathbf{x}_{g',s'}^{cls}\|}, \\
\alpha_{g,s} &= \sum_{g' \neq g} \alpha_{g,s,g'}, \\
\mathbf{x}_{s}^{cls} &= \sum_{g} \mathbf{x}_{g,s}^{cls} \frac{\exp(\alpha_{g,s})}{\sum_{\bar{g}} \exp(\alpha_{\bar{g},s})}
\end{aligned} \tag{2.3}$$

Where g and g' are group indices, s and s' are spatial indices, $\alpha_{g,s,g'}$ is the score of one group, $\alpha_{g,s}$ is the total score of all groups, and x_s^{cls} is the weighted class token.

The weighted class token contains information about the actions from the G phases. The layers in the long-term stage aim to perceive actions over long durations and under significant scene changes by exploring the inter-relationships of the short-term actions [9]. The structure of the EgoViT addresses the challenges of egocentric video action recognition, such as large-scale scene changes between distant frames and the varying contributions of nearby frames.

Chapter 3

Methodology

In this chapter, the proposed model based on the EgoViT architecture is presented. In this model, the original DCTG module is extended with the ability to integrate gaze information. Section 3.1 presents the overall architecture of the proposed model. The extraction of gaze-hand-object features is explained in detail in Section 3.2. Modifications made to the Video Swin Transformer [7] to integrate it with the architecture of EgoViT [9] are detailed in Section 3.3. The dynamic merging algorithms are explained in the last section 3.4.

3.1 Overall Architecture

In this thesis work, a Gaze-Enhanced EgoViT model is proposed. The overall architecture is shown in Figure 3.1 and is strictly inherited from the original EgoViT architecture. The model can be divided into three parts: the Short-term Stage, Intermediate Connection, and Long-term Stage. The key components of the model are the Gaze-Enhanced DCTG module, the modified Video Swin Transformer, and the Dynamic Merging module. In the following sections, the DCTG in EgoViT will be referred to as the "original DCTG", while the DCTG in the proposed model will be referred to as the "gaze-enhanced DCTG".

A Video Transformer commonly converts the input video frames into a sequence of feature vectors. Thus, the input frames can be denoted as the vector $I \in \mathbb{R}^{T_{sampled} \times H \times W \times C}$, where $T_{sampled}$ is the number of frames sampled from the video, H and W are the height and width of the frame, and C is the number of channels. The proposed model includes an additional input of gaze points vector $J \in \mathbb{R}^{T_{sampled} \times 2}$ representing the coordinates (x, y) of the gaze points in the $T_{sampled}$ frames.

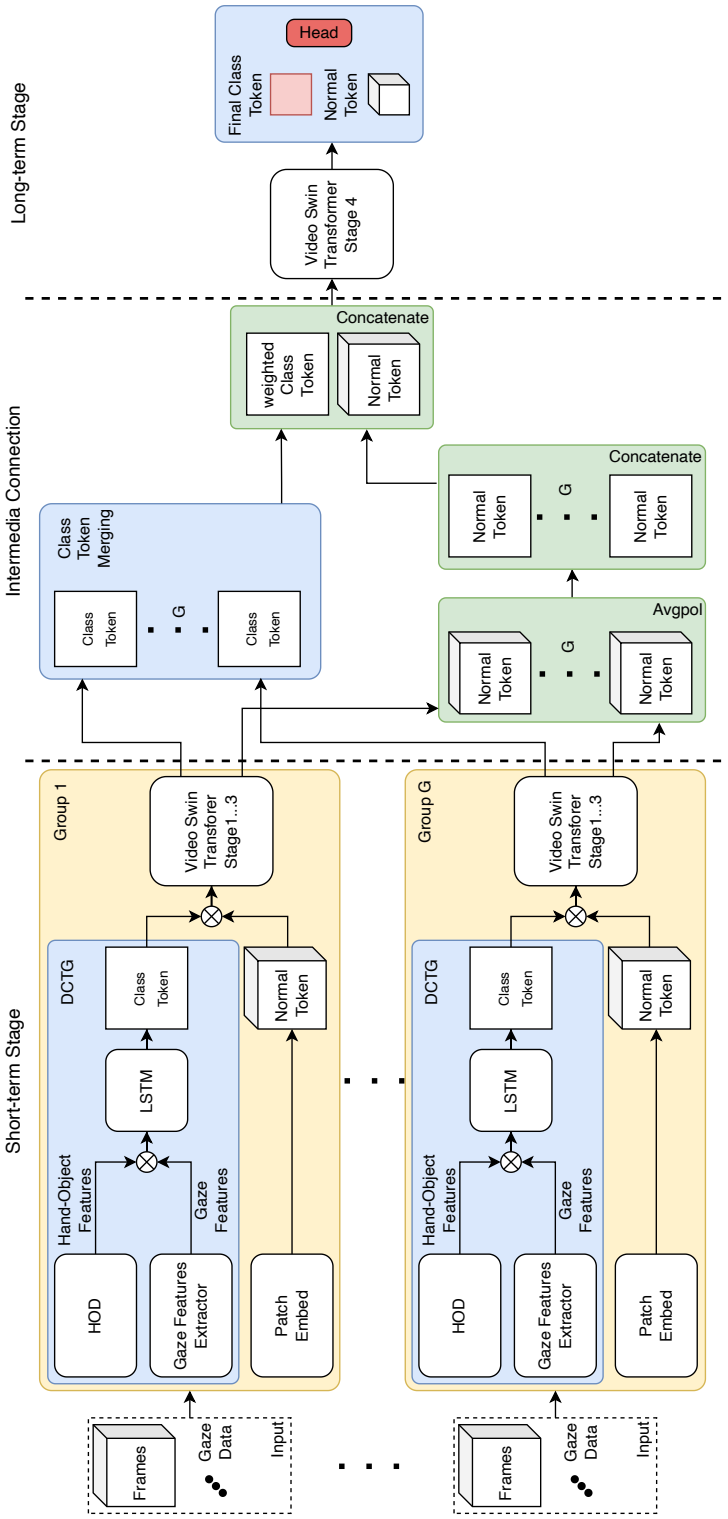


Figure 3.1: Overall architecture of the Gaze-Enhanced EgoViT.

The $T_{sampled}$ frames and their associated $T_{sampled}$ gaze points data are first split into G groups, uniformly distributed along the temporal dimension. The short-term phase of a video clips is denoted as $I_g \in \mathbb{R}^{T\times H\times W\times C}$ and the vector of gaze points is denoted as $J_g \in \mathbb{R}^{T\times 2}$, where $D = \frac{D_{sampled}}{G}$. The vectors I_g and J_g are then fed into the Gaze-Enhanced DCTG module to extract the gaze-hand-object features. The function of Gaze-Enhanced DCTG will be discussed in detail in section 3.2. In parallel, the I_g is fed into the PatchEmbedding module in the Video Swin Transformer to extract the features of the video frames.

The Vector I_g is segmented into non-overlapping patches, of size $\mathbb{R}^{P_T\times P_H\times P_W\times C}$. These patches are projected in a flatte layer into sequence data of $\mathbb{R}^{N_P\times(P_T\times P_H\times P_W\times C)}$, where $N_P = T_P \times H_P \times W_P = \frac{T}{P_T} \times \frac{H}{P_H} \times \frac{W}{P_W}$. A liner layer uses a matrix of size $\mathbb{R}^{(P_T\times P_H\times P_W\times C)\times D}$ to project the sequence data into a lower dimension D. The embedded vector is denoted as $X_P \in \mathbb{R}^{N_P\times D}$. The output of the Gaze-Enhanced DCTG module is treated as the class token x_{cls} with the shape $\mathbb{R}^{1\times D}$, and is concatenated with the sequence data X_P to form the input of the Video Swin Transformer. The Combination of x_{cls} and X_P is performed along the temporal dimension, positioning the class token as the first vector in this dimension. The sequence data X_P is defined as normal token in this model. Therefore, the input to the transformer is defined as:

$$x = [x_{cls}, X_P] \in \mathbb{R}^{(N_P+1)\times D} \tag{3.1}$$

The class token is designed to leverage the information exchange capabilities of the 3D Shifted Window Self-Attention mechanism in Video Swin Transformer, which facilitates the sharing of information between the class token and the normal token.

The Video Swin Transformer is modified to build the short-term, intermediate and long-term architecture. It is divided in two parts: the PatchEmbedding layers and stage 1 to stage 3 are in the short-term stage, which is responsible for extracting local temporal information. The stage 4 is in the long-term stage, which is responsible for extracting global temporal information. Between the two parts of the Video Swin Transformer, the Dynamic Merging module is used to merge the output of the short-term stage. It calculates the score α of short-term class token for each group. Then, all scores are summed and normalized to obtain the total score α_{total}. A weighted class token is then calculated from α_{total} and the class tokens x_g^{cls} of each group. In this module, the weights are assigned, and the class tokens are aggregated based on the short-term actions. The Dynamic Merging module is discussed in detail in section 3.4.

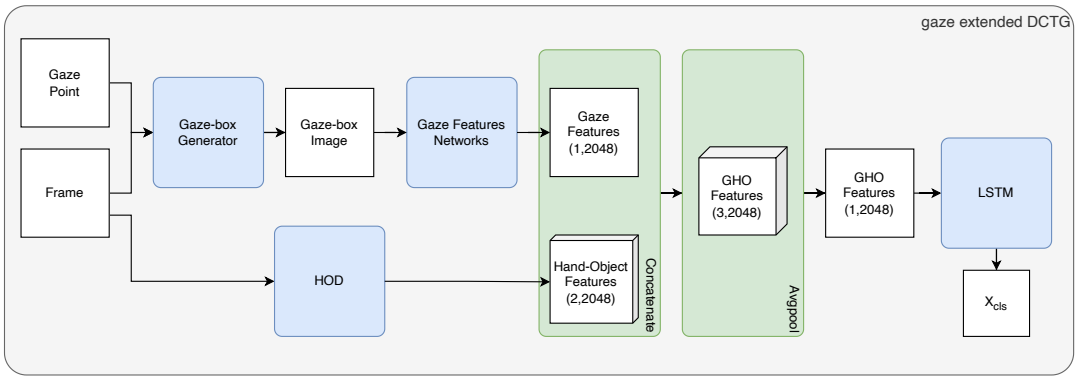


Figure 3.2: The data pipeline of the Gaze-Enhanced DCTG module.

The input of long-term stage is the concatenation of the weighted class token x_{st}^{cls} and the normal token X_{st}^{P}, where the subscript "st" for short-term. Therefore, the input of the long-term stage is defined as:

$$x_{lt} = [x_{st}^{cls}, X_{st}^{P}] \in \mathbb{R}^{(N_P+1)\times D_{st3}} \tag{3.2}$$

Where D_{st} is the dimension of the output of the stage3 in Video Swin Transformer and the subscript "lt" denotes for long-term.

The x_{lt} is fed into the stage 4 of Video Swin Transformer to extract the global temporal information. The final score of the action prediction is calculated from the Head, which computes the score of each action class. The action class with the highest score is selected as the final prediction.

3.2 The Gaze-Enhanced DCTG Module

To integrate gaze information, the Gaze-Enhanced DCTG module was developed. This module is a key component of the proposed model, responsible for extracting gaze features from the given gaze points and merging them with hand-object features. Finally, a class token is generated from the gaze-hand-object features and fed into subsequent layers. The module consists of three submodules: the Gaze Feature Extractor, the Hand-Object Feature Extractor, and the long short-term memory (LSTM) module. The structure of the Gaze-Enhanced DCTG module is shown in Figure 3.2.

The Gaze Feature Extractor is responsible for extracting the gaze features from the given gaze points and frames. The gaze points and frames are first fed into the Gaze-Box Generator. The gaze-box represents the region of the gaze area in

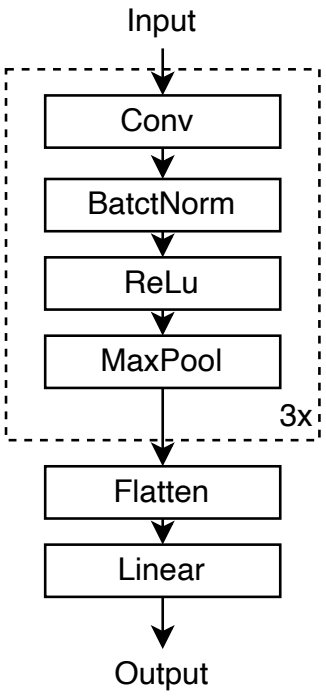


Figure 3.3: The structure of the Gaze Feature Networks.

the frame, with the gaze point being the center of the gaze region. A square box of size 35×35 is generated. Based on the gaze-box, a gaze image is cropped from the frame. The gaze image is then fed into the Gaze Feature Networks to extract the gaze features. The T frames and their associated gaze points are sent to the Gaze-Enhanced DCTG, and the input of the Gaze Feature Networks can be denoted as $I^{gb} \in \mathbb{R}^{T \times 35 \times 35 \times C}$, where "gb" stands for gaze-box.

The Gaze Feature Networks is a convolutional neural network, as shown in Figure 3.3. The network consists of three convolutional blocks, each containing a convolutional layer, a batch normalization layer, a ReLU activation layer, and a max-pooling layer. At the end, the gaze features pass through a flatten layer and a linear layer to generate the output in D-dimensions. The output of the Gaze Feature Networks is denoted as $I^{gaze} \in \mathbb{R}^{T \times D}$, where D is the dimension of the gaze features.

A modified Hand and Object Detector module [8] is applied offline in this thesis. The HOD is a pre-trained hand-object detector based on Faster-RCNN networks [37]. It is designed to detect and classify hands and the objects they interact with in videos. The output consists of the bounding boxes of the hands and objects, as well as their class labels. The code of HOD is modified to also provide the features

of the hands and objects.

Let $I \in \mathbb{R}^{T\times H\times W\times C}$ be the input frames. The HOD predicts bounding boxes BB for hands and objects, and a feature map I^{base} is generated by the base part of HOD. The bounding box predictions are overlapped and, according to the credibility ranking, the top-M hand and top-M object detections with confidence scores $\theta > 0.5$ are chosen. The feature maps, along with the detections, are fed into the "RoIAlign" layers and the "top feature refine" module to generate 2048-dimensional vectors for each detected hand and object.

The hand-object features are denoted as $F^{HO} \in \mathbb{R}^{T\times 2M\times 2048}$, where $2M$ represents the M hand (including left and right) features and M object features. Average pooling is then applied to the M hand features and M object features to generate the final hand-object features, denoted as $I^{HO} \in \mathbb{R}^{T\times 2\times 2048}$. The hand-object feature extraction process in the Gaze-Enhanced DCTG module can be described by the following equation:

$$
\begin{aligned}
&cls_t, BB_t = HOD(I_t, \theta_t), \quad \text{for } t \in [1, T] \\
&I_t^{\text{base}} = HOD_{\text{base}}(I_t) \in \mathbb{R}^{1024\times H^b\times W^b}, \\
&I_t^{\text{align}} = \text{ROIAlign}(I_t^{\text{base}}, BB_t) \in \mathbb{R}^{2M\times 1024\times H^a\times W^a}, \\
&I_t^{HO} = HOD_{\text{top}}(I_t^{\text{align}}) \in \mathbb{R}^{2M\times 2048}, \\
&F_t^{HO} = \text{AvgPool}(I_t^{HO}) \in \mathbb{R}^{2\times 2048}, \\
&F^{HO} = [F_1^{HO}, \ldots, F_T^{HO}] \in \mathbb{R}^{T\times 2\times 2048}
\end{aligned}
\tag{3.3}
$$

where I_t is the t-th frame along temporal axis, cls_t are the class labels of the hands and objects, which are not needed in this thesis. H^b and W^b are the height and width of the feature maps I_t^{base}, H^a and W^a are the height and width of the feature maps I_t^{align}. M is the number of hands and objects detected in each frame.

The extracted gaze features $F^G \in \mathbb{R}^{T\times 2\times 2048}$ and hand-object features $F^{HO} \in \mathbb{R}^{T\times 1\times 2048}$ are concatenated. Thus, the gaze-hand-object features are denoted as $F^{GHO} \in \mathbb{R}^{T\times 3\times 2048}$. The mean value of these three types of features is then calcu-

lated as:

$$F^{GHO} = [F^{G}, F^{HO}] \in \mathbb{R}^{T\times 3\times 2048},$$
$$F_t^{GHO} = \frac{1}{3}\sum_{t=1}^{3} F_{t,i}^{GHO}, \quad \text{for } t \in [1,T], \quad i \in [1,3] \tag{3.4}$$

The final gaze-hand-object features are denoted as $F^{GHO} \in \mathbb{R}^{T\times 2048}$. According to [9], applying the LSTM achievs better performence in aggregating knowledge from the temporal dimension. Therefore, the LSTM module is applied to generate the class token $x_{cls} \in \mathbb{R}^{T\times D}$ from the gaze-hand-object features F^{GHO}. The gaze-hand-object features is projected from 2048-dimensional features to D-dimensional features. The LSTM provided in the pytorch liberary is used in this thesis. The last state of the output of the LSTM is defined as the class token x_{cls}. The procedure for generating the class token can be described as follows:

$$F^{GHO'} = \text{LSTM}([F_1^{GHO}, \ldots, F_T^{GHO}]) \in \mathbb{R}^{T\times D},$$
$$x_{cls} = F^{GHO'}[-1] \in \mathbb{R}^{D} \tag{3.5}$$

where $F^{GHO'}$ is the output of the LSTM module, and $F^{GHO'}[-1]$ is the last state of the output along the temporal axis.

3.3 Integration of Video Swin Transformer

The Video Swin Transformer is used as the backbone in both the short-term and long-term stages, separated by the Dynamic Merging module. The Video Swin Transformer is modified to fit the architecture of the Gaze-Enhanced EgoViT by dividing it into two parts. The PatchEmbedding layers and stages 1 to 3 are in the short-term stage, responsible for extracting local temporal information. Stage 4 is in the long-term stage, responsible for extracting global temporal information. The configuration of the Video Swin Transformer follows the Swin-B model in [7].

The structure of the Video Swin Transformer in the short-term stage is shown in Figure 3.4. The 3D local window is defined as $WI \in \mathbb{R}^{WI_T\times WI_H\times WI_W}$. The patches are denoted as $X^P \in \mathbb{R}^{T_P\times H_P\times W_P\times D}$, where $T_{WI} = \frac{T_P}{WI_T}$, $H_{WI} = \frac{H_P}{WI_H}$, and $W_{WI} = \frac{W_P}{WI_W}$. There are a total of $N_{WI} = T_{WI} \times H_{WI} \times W_{WI}$ windows. Unlike a classic vision transformer, the Video Swin Transformer does not use the class token as the first token in the sequence. Instead, the normal token from 3D shifted windows is used to aggregate information. Therefore, the dimension of the class

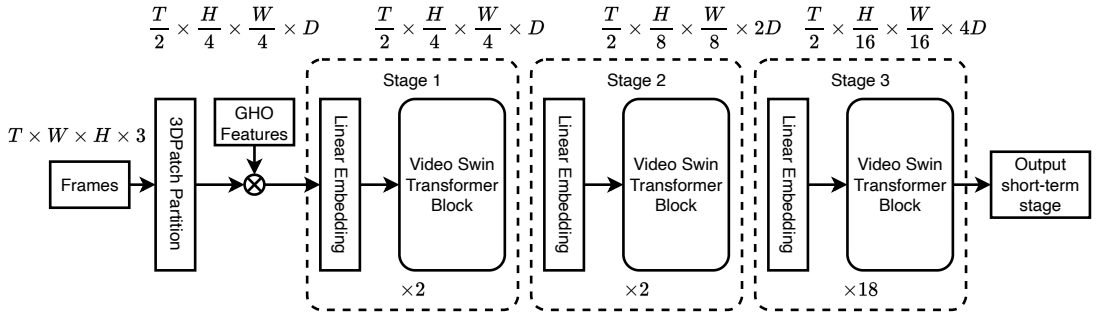


Figure 3.4: The structure and the data pipeline of the short-term stage.

token in the Gaze-Enhanced EgoViT is expanded to assign the same class token to all 3D windows. After the assignment, each 3D window has a class token and updates it independently.

The input to stage 1 is denoted as:

$$\begin{aligned} I_{WI,i} &= [x_{cls,i}, x_{WI,i}] \quad i \in [1, N_{WI}], \\ I_{WI,i} &\in \mathbb{R}^{((1+WI_T)\times WI_H \times WI_W)\times D} \end{aligned} \tag{3.6}$$

where i denotes the i^{th} window and $x_{WI,i}$ is the normal token of the i^{th} window.

The operation of the dynamic class token in a transformer block is expressed in the following equation:

$$\begin{aligned} \hat{I}^{l}_{WI,i} &= \text{W-MSA}(\text{LN}(I^{l-1}_{WI,i})) + I^{l-1}_{WI,i}, \\ I^{l}_{WI,i} &= \text{MLP}(\text{LN}(\hat{I}^{l}_{WI,i})) + \hat{I}^{l}_{WI,i}, \\ \hat{I}^{l+1}_{WI,i} &= \text{SW-MSA}(\text{LN}(I^{l}_{WI,i})) + I^{l}_{WI,i}, \\ I^{l+1}_{WI,i} &= \text{MLP}(\text{LN}(\hat{I}^{l+1}_{WI,i})) + \hat{I}^{l+1}_{WI,i}, \end{aligned} \tag{3.7}$$

Where LN refers to Layer Normalization, (S)W-MSA refers to the (shifted) Windowed Multi-head Self-Attention, and MLP refers to the Multi-Layer Perceptron. $\hat{I}^{l}_{WI,i}$ and $I^{l}_{WI,i}$ are the output of (S)W-MSA and MLP in the l^{th} block, respectively.

The class tokens are attached to the first position in the temporal dimension of the 3D windows. This produces a hierarchical representation similar to the Video Swin Transformer. The class token neighborhoods within a 2×2 spatial space are concatenated, which downsamples the spatial dimension by a factor of 2 and increases the D-dimensional features by a factor of 4. A linear layer is then applied to project the features to half of their dimension. For example, the output of stage

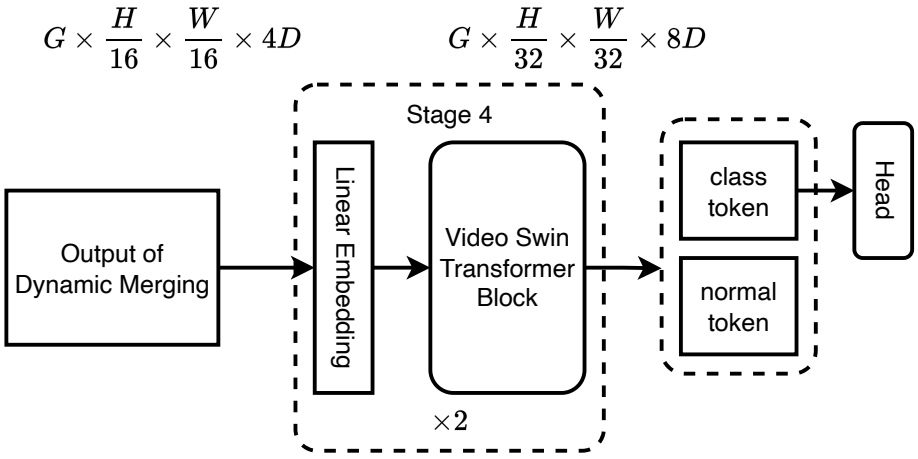


Figure 3.5: The structure the data pipeline of long-term stage.

1 is $\frac{T}{2} \times \frac{H}{4} \times \frac{W}{4} \times D$, then after the Linear Embedding and Transformer Block, the output of the stage 2 becomes $\frac{T}{2} \times \frac{H}{8} \times \frac{W}{8} \times 2D$. Note that downsampling is not applied in the temporal dimension.

Therefore, the proposed model combines the characteristics of the shifted Window Multi-head Self-Attention and the EgoViT architecture. The shifted Window Multi-head Self-Attention exchanges information in the spatial dimension, while each group in the short-term stage of EgoViT exchanges information in the local temporal dimension.

The long-term stage consists of the stage 4 of the Video Swin Transformer. The structure and goal of this stage follow the approach in [9], aiming to perceive actions over long durations by exploring the inter-relationships established in the short-term stage. The input tensor x_{mrg} to stage 4 is the concatenation of the weighted class token x_{mrg}^{cls} and the normal token x_{mrg}^{P}, which is expressed as:

$$x_{merge} = [x_{mrg}^{cls}, x_{mrg}^{P}] \in \mathbb{R}^{(1+G)\times\frac{H}{16}\times\frac{W}{16}\times 4D} \tag{3.8}$$

Figure 3.5 shows the structure of the Video Swin Transformer in the long-term stage. After processing in stage 4, the output is separated into two parts: the class token $x_{lt}^{cls} \in \mathbb{R}^{1\times H_{wi}^{s4}\times W_{wi}^{s4}\times 4D}$ and the normal token $x_{lt}^{P} \in \mathbb{R}^{G\times\frac{H}{32}\times\frac{W}{32}\times 4D}$, where the subscript "lt" denotes the long-term stage, and H_{wi}^{s4} and W_{wi}^{s4} are the height and width of the window in stage 4. Only the class token is fed into the Head to calculate the score for each action class. The Head consists of a sequence of layers, including an Average Pooling layer, a Flatten layer, and a Linear layer, which together transform the class token from 2048-dimensional features to a score between 0 and 1, representing the final score for each action class.

3.4 Dynamic Merging Module

After being processed in the short-term stage, the class token serves as a summary of the short-term actions. The Dynamic Merging module is designed to assign a larger weight to the class tokens representing key short-term actions [9]. The structure of the Dynamic Merging module is shown in Figure 3.6. The input tensor is $x_{st} \in \mathbb{R}^{G\times(1+\frac{T}{2})\times\frac{H}{16}\times\frac{W}{16}\times 4D}$. Before merging, the class token x_{cls} and the normal token x_P are separated. The class token x_{cls} from each group are fed into an AvgPool3d layer to obtain the class token for each 3D window. According to the window size of the Video Swin Transformer (2, 7, 7), the kernel size of AvgPool3d is set to (1, 7, 7). Then the weighted class token x^{cls}_{mrg} is calculated from the G downsampled class tokens.

The merging algorithm is expressed in Equation 2.3. First, the dot product between the class tokens of different groups is calculated. The score $\alpha_{g,s,g'}$ is the dot product divided by their l2 norms. The total score $\alpha_{g,s}$ for $x^{cls}_{g,s}$ is calculated by summing the scores along the spatial and group axes. The scores for all class tokens are normalized along the group axis using the softmax operator. The weighted class

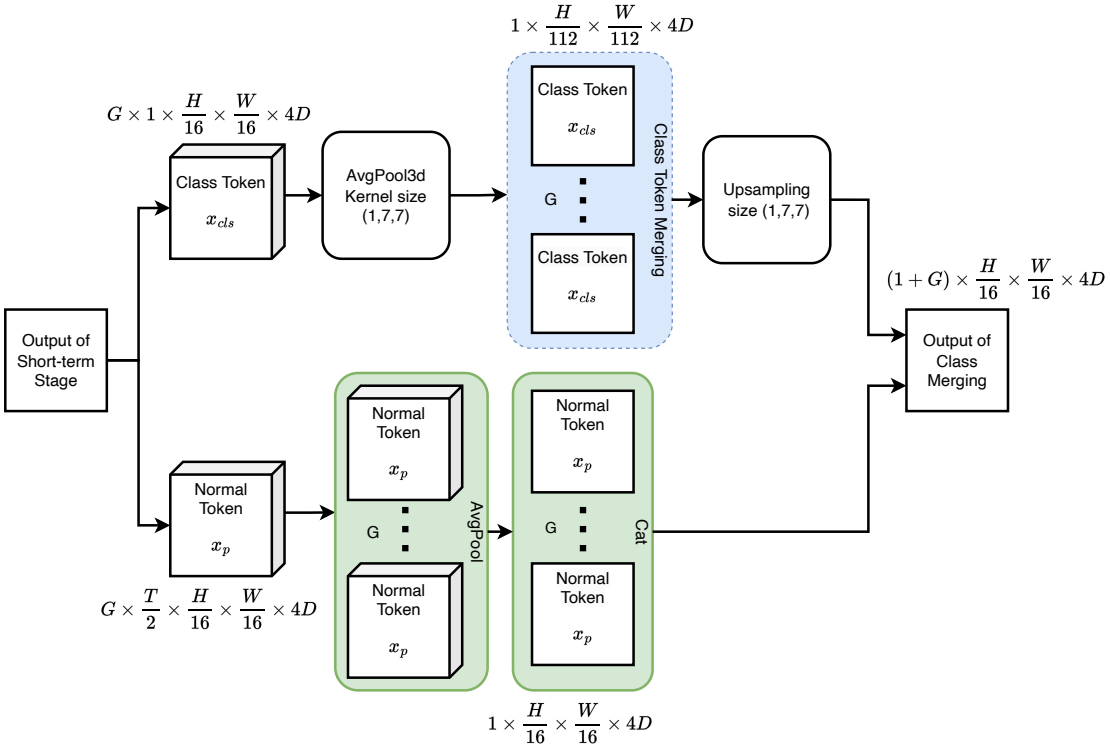


Figure 3.6: The structure of the Dynamic Merging module.

token x_{mrg}^{cls} for the lont-term stage is then obtained by computing the weighted sum of class tokens along the group axis. Finally, the weighted class token is sent to an upsampling layer with size (1, 7, 7) to match the size of $1 \times \frac{H}{16} \times \frac{W}{16} \times 4D$.

An AvgPool layer is first applied to the G normal tokens x_P along the temporal axis, and then the G normal tokens are concatenated to form the normal token with a shape of $G \times \frac{H}{16} \times \frac{W}{16} \times 4D$. At the end of this module, the class token x_{mrg}^{cls} and the normal token x_{mrg}^{P} are concatenated to form the input tensor $x_{lt} \in \mathbb{R}^{(1+G)\times\frac{H}{16}\times\frac{W}{16}\times 4D}$ for the long-term stage.

Chapter 4

Experiments and Results

In this chapter, the experiment setup and results are described in detail. In section 4.1, the datasets, developing environment, metrics, and implementation details are explained. Section 4.2 presents the results of the original EgoViT on the EGTEA Gaze+ dataset. In Section 4.3, the results of the enhanced EgoViT training with gaze information are presented. Finally, the results of the experiments are analyzed and discussed in Section 4.5.

4.1 Setup

Datasets The largest commonly used egocentric video dataset is EPIC-KITCHENS [38], which contains videos of daily activities from multiple participants in their natural environment. However, this dataset does not provide gaze data from the videos. For this reason, the EGTEA Gaze+ [15] dataset is utilized in this thesis. EGTEA Gaze+ is a large and comprehensive dataset for first person view (FPV) actions and gaze tracking. It includes HD videos, gaze tracking data, and frame-level action annotations. The dataset consists of 86 unique sessions from 32 subjects across 7 recipes. The annotations include 10321 action instances from 106 action categories, with an average action instance duration of 4.2 seconds. There are three non-overlapping train and test sets available, with 8299/2022, 8299/2022 and 9230/2021 samples (train/test). These splits are generated through random sampling, ensuring that approximately 80% of the samples per category are included. The split set 1 is used in this thesis for training and testing. For all Experiments, this study follows prior work by reporting top-1 accuracy, top-5 accuracy and average mean class accuracy.

Developing Environment The experiments are conducted on a GPU server in SimTech Stuttgart. The server runs Ubuntu operating system, and the model with its experiments are implemented in Python 3.9. The deep learning framework PyTorch 2.2 and CUDA 12.1 are used for the implementation. The important ependencies and their versions are listed below:

- torch==1.10.0
- torchvision==0.17.0
- numpy==1.26.4
- pandas==2.2.1
- opencv-python==4.10.0.82

Implementation Details The video clips in EGTEA Gaze+ dataset have an average length of 3.2 seconds, although some clips are significantly longer. To handle this, the video clips are uniformly sampling into 32 frames. The frames are resized to 224×224 pixels, resulting in an input tensor of shape $32 \times 3 \times 224 \times 224$. Dimension 3 represents the RGB channels of the image. Another input tensor contains gaze tracing data with a shape of $32 \times 1 \times 1$, where 1×1 represents gaze coordinates (x, y)in each frame. The gaze coordinates are normalized to the range of $[0, 1]$.

Frame extraction from the video clips and the processing of gaze features can be time-consuming during training. To reduce the training time, frame extraction and gaze-hand-object features processing are performed offline and saved as a NumPy zipped (.npz) file. And the action label is also read and saved in this zipped file. Thus, the image, gaze-hand-object features, and action label of one video clip are saved in a single file. The structure of the NumPy zipped (.npz) file is as follows:

$$\text{Preprocessed Data:}(\text{frames} : [32, 3, 224, 224], \quad \text{features} : [32, 3, 2048], \quad \text{label} : [1])$$

The training process opens the data file only once for each video clip, significantly reducing the training time. All experiments are conducted on the prepared data.

The models are trained using the AdamW [39] optimizer. The learning rate is set to $1e{-}5$, and the batch size is set to 4. The layers the in Video Swin Transformer use the pretrained weights from KINETICS400_V1. The models are trained for 20 epochs. In some experiments, a cosine decay learning rate scheduler with a linear warm-up of 2.5 epochs is used, following the approach in [7]. The stochastic depth

rate and weight decay are adopted as in [7], set to 0.3 and 0.05 respectively. The configuration of Swin-B is used. The architecture hyperparameters of Swin-B are as follows:

$$\text{Swin-B:} \quad C = 128, \quad \text{layer numbers} = \{2, 2, 18, 2\} \tag{4.1}$$

Where C represents the C-dimentional features (see Figure 2.5), and $\{2, 2, 18, 2\}$ represents the number of times the Transformer Block is repeated in each stage.

Metrics For inference, the same data structure as used in training is applied. The model is evaluated on the test split1 of the EGTEA Gaze+ dataset. The top-1 accuracy, top-5 accuracy, and mean class accuracy are calculated. The top-1 accuracy is the proportion of correctly predicted class (with the highest predicted probability) among all samples. It is the most straightforward accuracy metric. The calculation of top-1 accuracy is:

$$\text{Top-1 Accuracy} = \frac{\text{Number of correctly predicted samples}}{\text{Total number of samples}} \tag{4.2}$$

Top-5 accuracy refers the percentage of instances where the true class label is within the top five predicted class label. The formula for top-5 accuracy is:

$$\text{Top-5 Accuracy} = \frac{\text{Number of correctly predicted samples in top-5}}{\text{Total number of samples}} \tag{4.3}$$

The mean class accuracy is the average accuracy for each class. It accounts class imbalance by assigning equal weight to each class. The formula for mean class accuracy is:

$$\text{Mean Class Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{Number of correctly predicted samples for class i}}{\text{Total number of samples in class i}} \tag{4.4}$$

Where N is the number of classes.

In this thesis, a series number of experiments are conducted. Different models are trained with different features and configurations. For a simple discription of the experiments, the experiments ID are denoted and explained in the Table 4.1. The experiments ID are used to identify the experiments in the following sections. When not mentioned, the experiments are trained with gaze version 1, with pretrained weights from KINETICS400_V1, and a fixed learning rate of 1e−5.

Table 4.1: Experiments ID and Description

Experiment ID	Description
Orig_HO_no_pretrain	Original EgoViT: without pretrained weights
Orig_HO	Original EgoViT: pretrained weights
Orig_HO_sched. LR	Original EgoViT: pretrained weights and scheduler LR
Enh_GHO	Enhanced EgoViT: GHO features
Enh_G	Enhanced EgoViT: G features
Enh_GHO_v2	Enhanced EgoViT: GHO features and gaze_v2
Enh_G_v2	Enhanced EgoViT: G features and gaze_v2
Enh_v2_GHO_v2	Enhanced EgoViT_v2: GHO features and gaze_v2
Enh_v3_HO	Enhanced EgoViT_v3: HO features
Enh_v3_GHO_v2	Enhanced EgoViT_v3: GHO features and gaze_v2
Enh_v3_G_v2	Enhanced EgoViT_v3: G features and gaze_v2

4.2 Training the Original EgoViT with EGTEA Gaze+ Dataset

Because the code for the original EgoViT is not publicly available, an approximate model is first built in this thesis. To compare the effect of gaze information, the original EgoViT from [9] is trained on the EGTEA Gaze+ dataset and used as the baseline for all experiments. Therefore, only frames and hand-object features in the video clips are fed into the original EgoViT model.

The original EgoViT model is first trained without pretrained weights from KINETICS400_V1, using a learning rate of 1e−5 for all layers. The model is trained for 40 epochs. Figure 4.1 shows the training loss and accuracy of the original EgoViT model. The rot line represents the training loss, and the blue line represents the training accuracy. The model converges rapidly in the first 5 epochs and the loss decreases consistently between 5 and 30 epochs. After 30 epochs, the model converges slowly to the minimum loss. The accuracy follows a similar trend, inversely related to the loss.

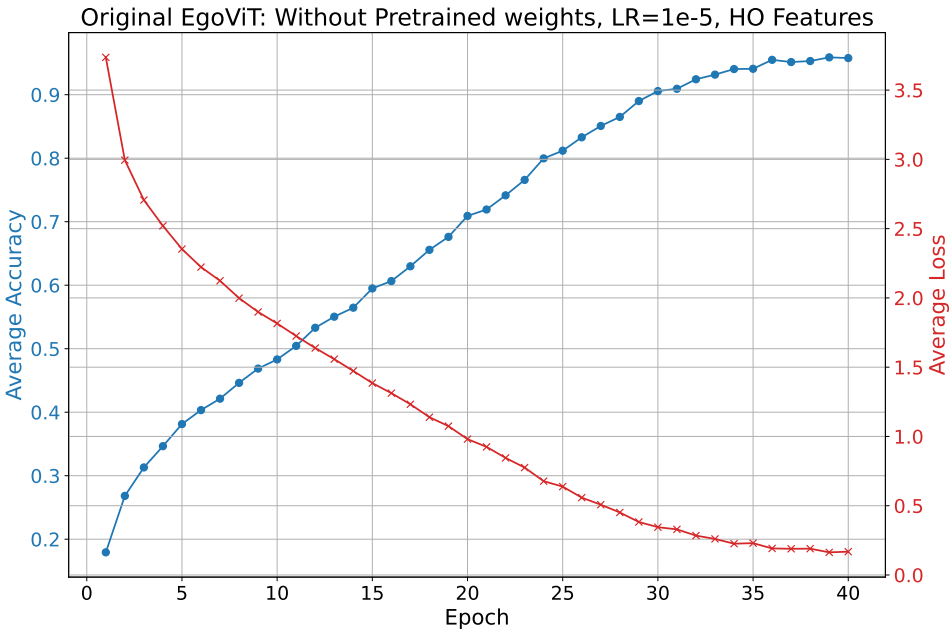


Figure 4.1: The training loss and accuracy curves for the original EgoViT model trained without pretrained weights. The model is trained over 40 epochs with a learning rate of 1e-5. Blue curve: training accuracy, red curve: training loss.

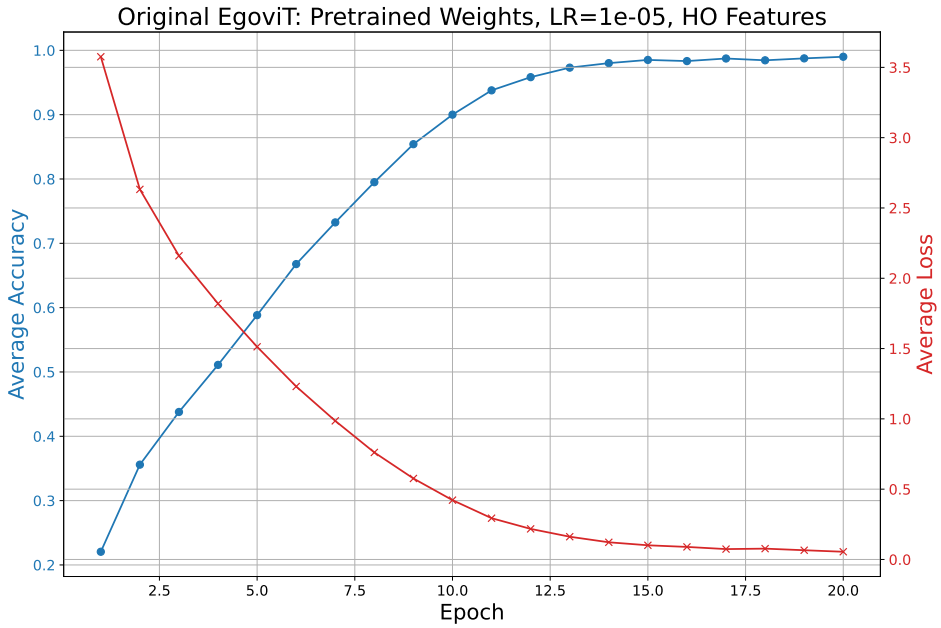


Figure 4.2: The training loss and accuracy curves for the original EgoViT model trained with pretrained weights. The model is trained over 20 epochs with a learning rate of 1e-5. Blue curve: training accuracy, red curve: training loss.

To improve the training performance, the original EgoViT model is trained with pretrained weights from KINETICS400_V1, while maintaining a learning rate of 1e−5. The model is trained for 20 epochs. Figure 4.2 shows the training loss and accuracy of the original EgoViT model with pretrained weights. The loss decreases rapidly in the first 3 epochs and then follows a shorter, consistently decreasing period. After 20 epochs, the model converges slowly to the minimum loss. The accuracy follows a similar trend, inversely related to the loss. By the 13th epoch, the model is almost fully converged and then decreases slowly to the minimum. This result indicates that the model with pretrained weights performs better than the model without them. Therefore, in the following experiments, the model with pretrained weights is used.

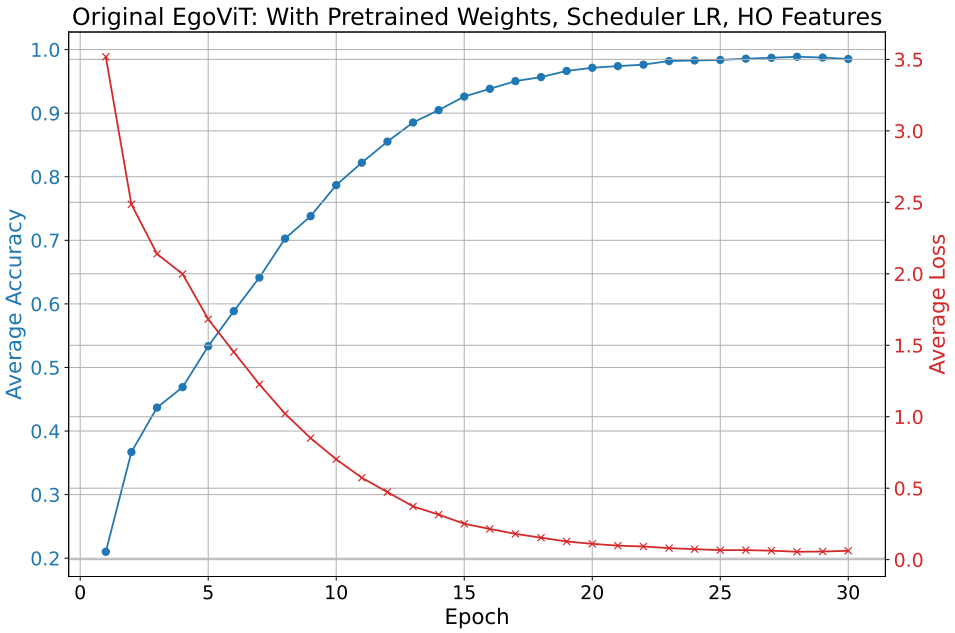


Figure 4.3: The training loss and accuracy curves for the original EgoViT model trained with pretrained weights. The model is trained over 30 epochs with a scheduler learning rate. Blue curve: training accuracy, red curve: training loss.

The learning rate is another hyperparameter that affects model performance. A cosine scheduler learning rate, as described in [7], is applied to fine-tune the learning rate. The model is trained for 30 epochs. Figure 4.3 shows the training loss and accuracy of the original EgoViT model with pretrained weights and scheduler learning rate. In the first 10 epochs, the loss decreases rapidly. After 15 epochs, the model converges slowly to the minimum loss. Between 20 and 25 epochs, the model achieves maximum accuracy and minimum loss.

The testing results of three training methods are shown in Table 4.2. The model with pretrained weights and a scheduler learning rate achieves the highest top-1 accuracy of 0.517, followed by the model with pretrained weights and a fixed learning rate of 0.515. The model with pretrained weights and a scheduler learning rate has the worst performance, achieving the lowest top-1 accuracy, top-5 accuracy, and mean class accuracy of 0.484, 0.744, and 0.358, respectively. All three metrics follow a similar trend, with the model using a fixed learning rate of 1e−5 showing better performance overall. While the pretrained weights from KINETICS400_V1 did not improve EAR in inference, they accelerated the training process. One possible reason for the lower performance of the scheduler learning rate is that the Video Swin Transformer is divided into two parts and the model has many additional, making the scheduler learning rate configuration from [7] potentially unsuitable for the original EgoViT model. A new configuration for the scheduler learning rate could be explored in future work.

Table 4.2: Test Results of original EgoViT with HO Features

Experiment ID	Top-1 Acc.(%)	Top-5 Acc.(%)	Mean Class Acc.(%)
Orig_HO_no_pretrain	51.5	78.5	38.8
Orig_HO	51.7	75.2	40.6
Orig_HO_sched. LR	48.4	74.4	35.8

4.3 Training the Enhanced EgoViT with Gaze Information

In this section, the enhanced EgoViT is trained with additional gaze information using different methods to evaluate the effect of gaze information on the model. In this series of experiments, two types of gaze data are used for training and testing. The first type, referred to as gaze version 1, contains only the gaze tracking type of fixation. Thus, some sampled frmes may not have gaze data. The second type, referred to as gaze version 2, includes both fixation and saccade types of gaze tracking. Gaze version 2 has more collected gaze data from the dataset but also includes some frames that lack gaze data. For these frames, the missing gaze data is ignorded and gaze features are randomly generated, resulting in a highter overall quality of gaze data compared to gaze version 1.

For an ablation study, the enhanced EgoViT is trained with gaze-hand-object features and only gaze features seperetly. The model is trained for 20 epochs with a learing rate of 1e−5. The training loss and accuracy of the enhanced EgoViT model with gaze-hand-object features and gaze features are shown in Figure 4.4 and 4.5. The both training have a similar loss and accuracy curved line compared with the experiment orig_pretrain_HO in Figure 4.2. The three experiments have similarity loss converge rate and all reach the minimum loss after 15 epochs. The top-1, top-5 and mean class accuracy of the two experiments are shown in Table 4.3. The model with hand-object features achieves the highest top-1 accuracy of 0.517 and the highest mean class accuracy of 0.406, followed by the model with gaze-hand-object features, which achieves 0.514 and 0.400, respectively. However, the gaze-hand-object model has the highest top-5 accuracy of 0.767. The model with only gaze features has the lowest scores across all three metrics, with a top-1 accuracy of 0.489, a top-5 accuracy of 0.751, and a mean class accuracy of 0.377.

Table 4.3: Test Results on Models with Different Features

Experiment ID	Top-1 Acc.	Top-5 Acc.	Mean Class Acc.
Orig_HO	51.7	75.2	40.6
Enh_GHO	51.4	76.7	40.0
Enh_G	48.9	75.1	37.7

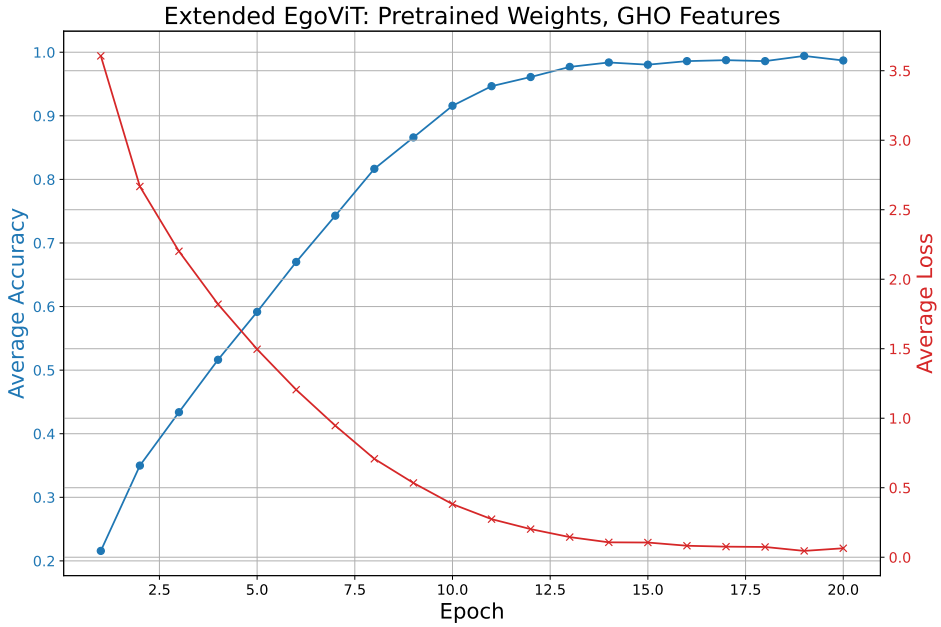


Figure 4.4: Training loss and accuracy of the enhanced EgoViT with gaze-hand-object features. Blue curve: training accuracy, red curve: training loss.

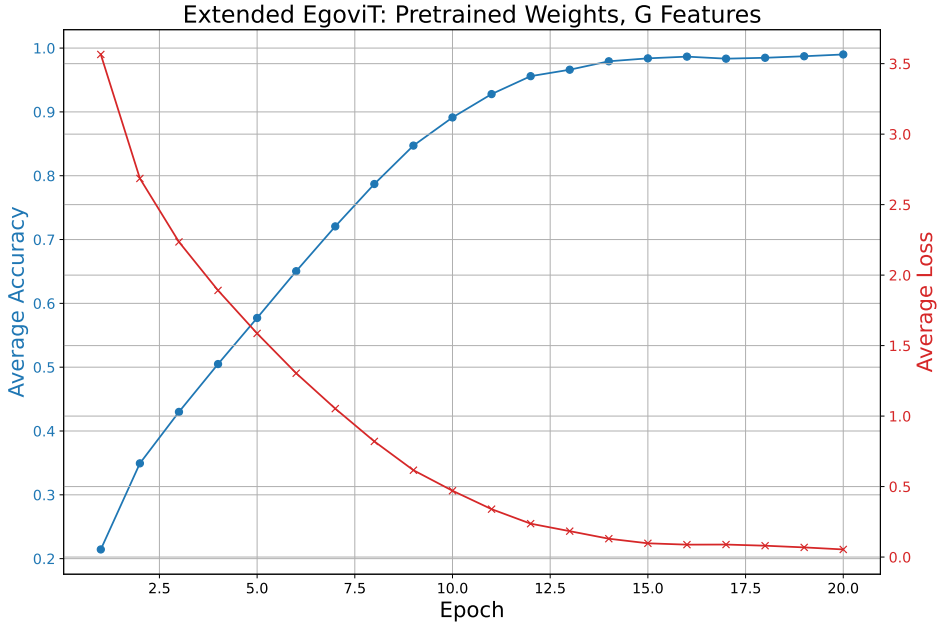


Figure 4.5: Training loss and accuracy of the enhanced EgoViT with gaze features. Blue curve: training accuracy, red curve: training loss.

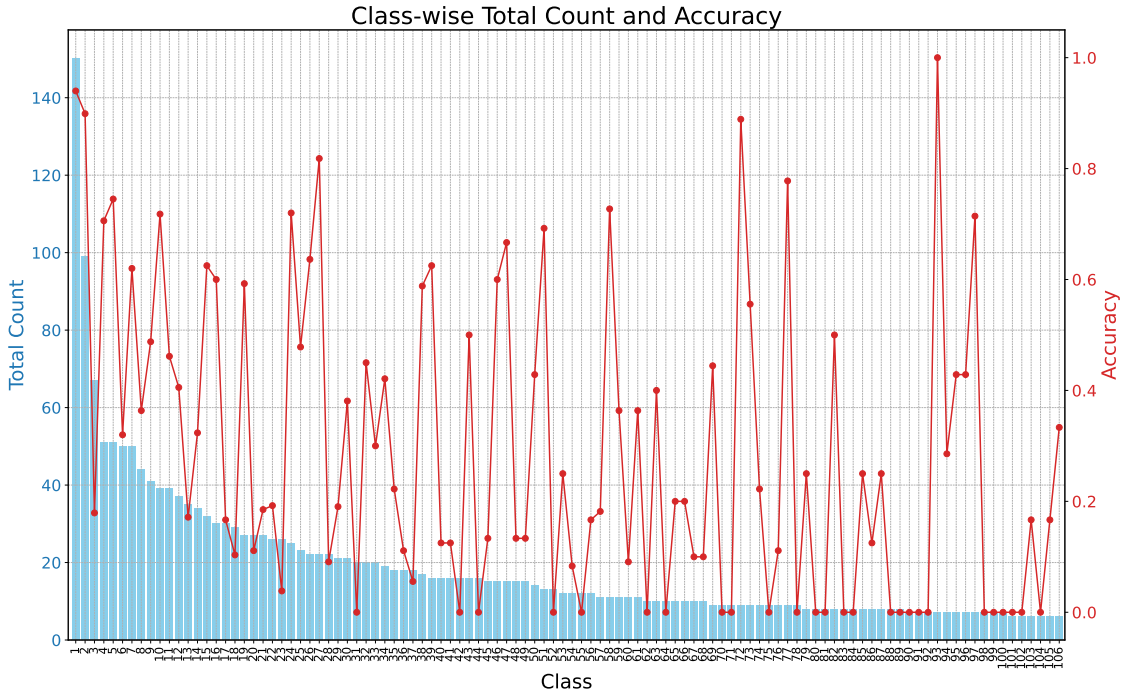


Figure 4.6: The class-wise total count (blue bars) and accuracy (red line) of the enhanced EgoViT model. The x-axis represents the different classes, while the left y-axis shows the total count of instances per class, and the right y-axis shows the accuracy for each class.

The number of samples for each action class (a total of 106 classes) in the test split1 is calculated and shown in Figure 4.6. The x-axis represents the action class, and the y-axis represents the number of samples in each class. The top-1 accuracy of the model with gaze-hand-object features is indicated by the red line in Figure 4.6. This figure shows the number of samples for each action class and the corresponding accuracy of the model with gaze-hand-object features. Class label 1 has the highest number of samples at 150, while the other classes have significantly fewer samples. From class label 31 to 106, the number of samples is under 20. The accuracy of action recognition for many classes with fewer samples is low, with some classes having an accuracy of 0. Since the distribution of samples in each class in train split1 is similar to that in test split1, the model is not able to effectively learn the features of classes with fewer samples.

Two experiments are conducted to explore the effect of gaze quality on the model. The enhanced EgoViT is trained with gaze-hand-object features and only gaze features, using gaze data extracted from gaze version 2. The hyperparameters for train-

ing remain the same as in the previous experiments. Table 4.4 shows the results and compares them with the results of the experiments Enh_GHO and Enh_G. The top-1 accuracy indicates that the model trained with gaze version 2 performs better than the model trained with gaze version 1. The experiment Enh_GHO_v2 achieves a top-1 accuracy of 0.520, which is 0.6% higher than the experiment Enh_GHO. The experiment Enh_G_v2 achieves a top-1 accuracy of 0.500, which is 1.1% higher than the experiment Enh_G and top-5 accuracy also improves by about 1.1%. These results demonstrate that the quality of gaze features significantly affects the model's performance. This effect is more noticeable when only gaze information is used.

Table 4.4: Test Results on Gaze Version 1 and Gaze Version 2

Experiment ID	Top-1 Acc.(%)	Top-5 Acc.(%)	Mean Class Acc.(%)
Enh_GHO	51.4	76.7	40.0
Enh_G	48.9	75.1	37.7
Enh_GHO_v2	52.0	76.3	38.7
Enh_G_v2	50.0	76.2	38.0

4.4 Training Variants of the Enhanced EgoViT

In this section, two variants of the enhanced EgoViT model are trained and evaluated. The first variant, referred to as enhanced EgoViT version 2, has a modified DCTG and PADM module. The second variant, referred to as enhanced EgoViT version 3, has a modified head. Both variants are trained with the same hyperparameters as the previous experiments, and gaze version 2 is utilized.

The DCTG module of enhanced EgoViT version 2 features a different feature merging mechanism. The gaze features and hand-object features are concatenated, and average pooling is not applied to the concatenated features. Thus, the output, i.e., the class token, has a shape of [32, 2, 2048]. This means that the gaze features and hand-object features are not merged into a single feature but are kept separate and processed independently in subsequent layers. In the PADM module, only the gaze features from the G groups are merged. At the end, only the gaze features are used as the class token for classification, while the hand-object features are included in the information exchange in the Video Swin Transformer.

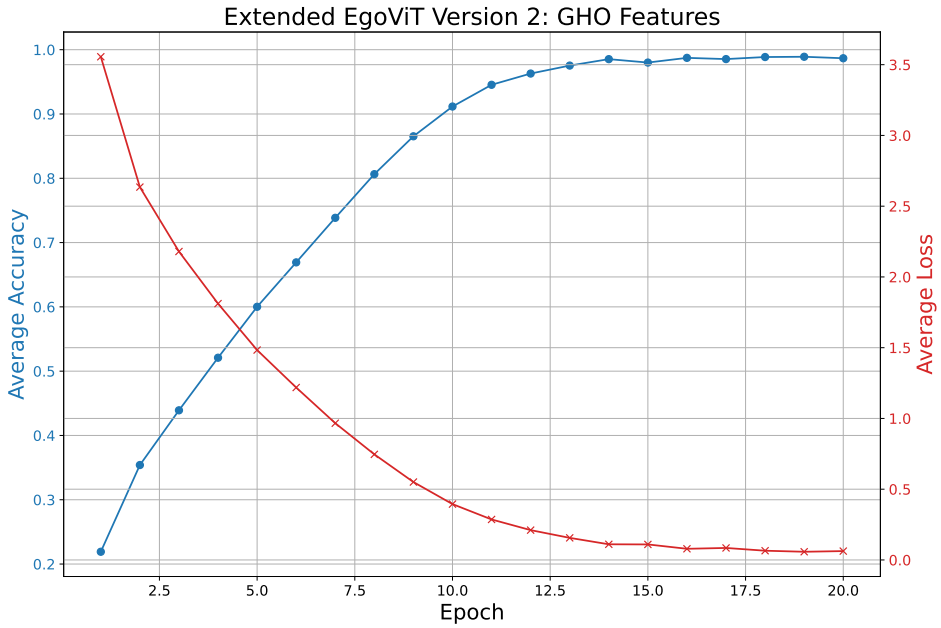


Figure 4.7: Training loss and accuracy of the enhanced EgoViT version 2 with gaze-hand-object features. Blue curve: training accuracy, red curve: training loss.

The training loss and accuracy of the enhanced EgoViT version 2 with gaze-hand-object features are shown in Figure 4.7. Comparing with Figure 4.4 the loss and accuracy curves of the two experiments are similar. They both reach the minimum loss after 15 epochs and have the kinks points at epoch 2 and 15. This training result indicates that the gaze features and hand-object features can be used as class tokens in the model and processed independently. This modification does not significantly affect the model's training performance.

The testing results of the enhanced EgoViT version 2 with gaze-hand-object features are shown in Table 4.5. The model achieves a top-1 accuracy of 50.2%, a top-5 accuracy of 75.8%, and a mean class accuracy of 38.3%. These results are lower than those of the experiment Enh_GHO_v2. This indicates that the modified DCTG and PADM modules do not improve the model's performance.

Table 4.5: Comparison of Test Results on EgoViT with Gaze Information

Training Method	Top-1 Acc.(%)	Top-5 Acc.(%)	Mean Class Acc.(%)
Enh_GHO_v2	52.0	76.3	38.7
Enh_v2_GHO_v2	50.4	75.8	38.3

The second variant, enhanced EgoViT version 3, has a different input configuration for the head. Both the weighted class token and the normal token from the long-term stage are directly fed into the head without separation. To evaluate the effect of this input configuration, the variant enhanced EgoViT version 3 is trained with hand-object features, gaze features, and gaze-hand-object features. The training loss and accuracy of the enhanced EgoViT version 3 with hand-object features, gaze-hand-object features, and only gaze features are shown in Figure 4.8, 4.9, and 4.10 respectively. They exhibit similar loss convergence curves, reaching the minimum loss after 15 epochs.

Additionally, the training loss of experiments Enh_GHO, Enh_GHO_v2, and Enh_GHO_v3 are compared in Figure 4.11. The loss curves of Enh_GHO and Enh_GHO_v2 represented by lines with cross markers and lines with circle markers, exhibit a similar curve trend. The line with square markers represents Enh_GHO_v3, which converges more slowly between epochs 4 and 15. This result indicates that the input configuration of the head and independently processing gaze features and hand-object features in the short-term and long-term stages do not significantly affect the model's training performance.

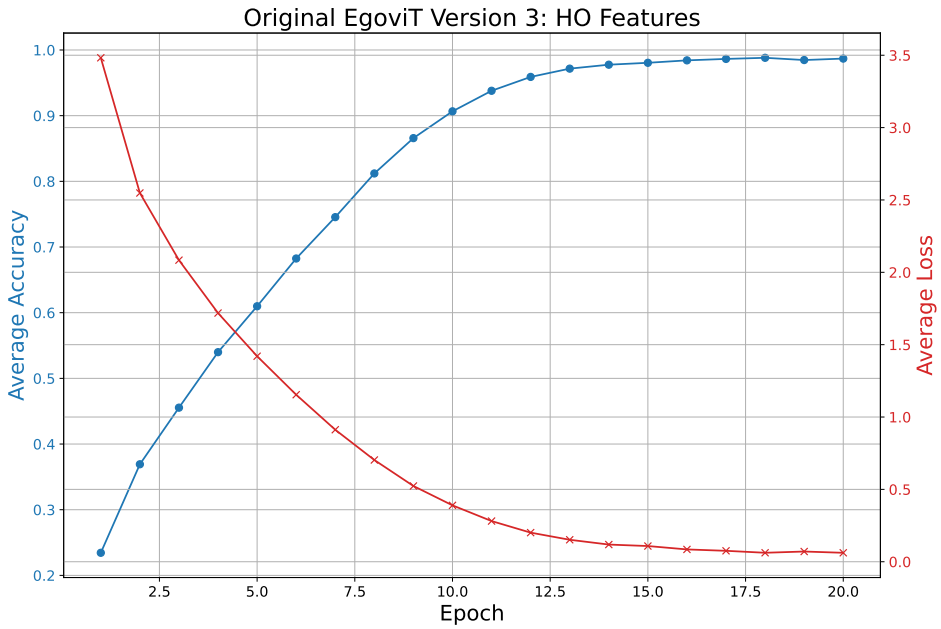


Figure 4.8: Training loss and accuracy of the original EgoViT version 3 with hand-object features. Blue curve: training accuracy, red curve: training loss.

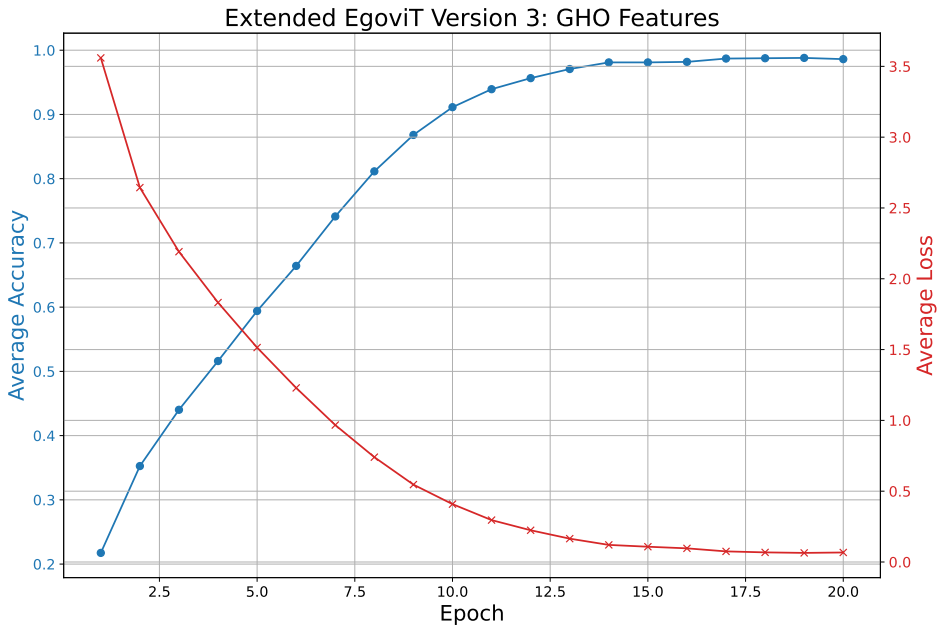


Figure 4.9: Training loss and accuracy of the enhanced EgoViT version 3 with gaze-hand-object features. Blue curve: training accuracy, red curve: training loss.

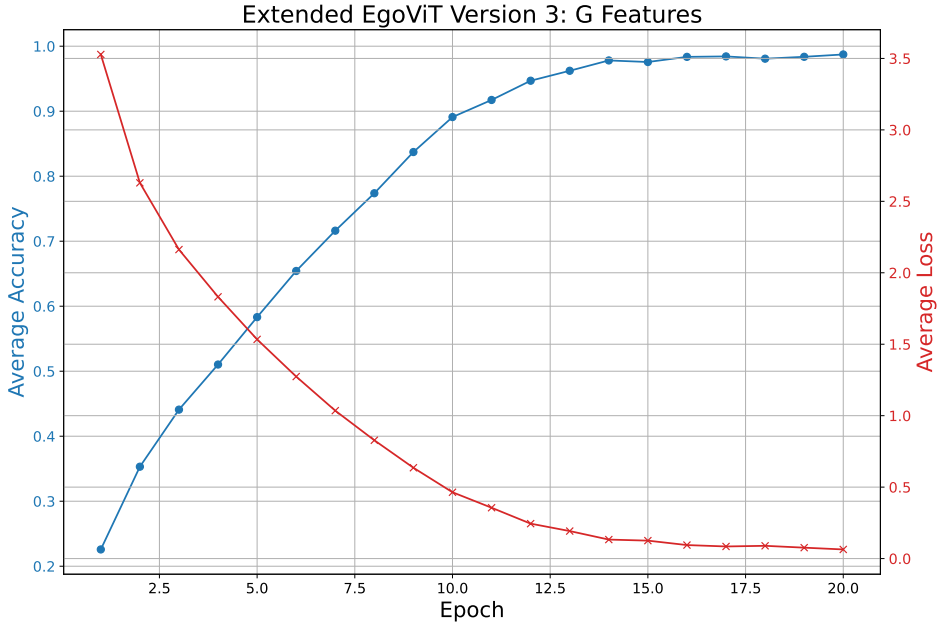


Figure 4.10: Training loss and accuracy of the enhanced EgoViT version 3 with gaze features. Blue curve: training accuracy, red curve: training loss.

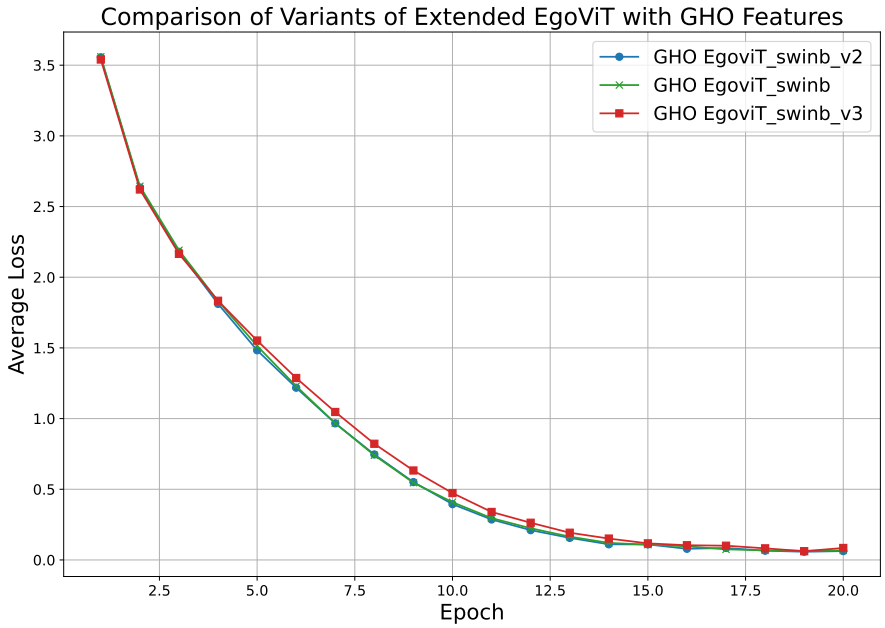


Figure 4.11: Comparison of training loss of the variants of enhanced EgoViT with gaze-hand-object features.

Table 4.6: Test Results on EgoViT Version 3 Trained on Different Features

Experiment ID	Top-1 Acc.(%)	Top-5 Acc.(%)	Mean Class Acc.(%)
Orig_v3_HO	51.1	77.4	40.5
Enh_v3_GHO_v2	51.2	77.9	39.2
Enh_v3_G_v2	50.8	75.9	37.8

The testing results of the enhanced EgoViT version 3 with hand-object features, gaze-hand-object features, and only gaze features are shown in Table 4.6. The experiment orig_v3_GHO achieves the highest top-1 and top-5 accuracy of 51.2% and 77.9%, respectively. However, However, the results are almost the same as those of the experimen Orig_v3_HO, with only 0.1% and 0.5% higher top-1 and top-5 accuracy, respectively. The experiment Enh_v3_G_v2 achieves the loswest accuracy across all three metrics. This result indicates that the input configuration of the head does not significantly affect the EAR performance.

4.5 Disscussion

The results of all experiments are analyzed and discussed in this section. A comparison of the three metrics (top-1 accuracy, top-5 accuracy, and mean class accuracy) for all experiments is shown in Table 4.7, providing an overview of the inference results.

The experiments conducted in section 4.2 show that the original EgoViT performs better with pretrained weights from the Video Swin Transformer. Although the pretrained weights are trained on an image dataset and the architecture of the Video Swin Transformer is modified, the model with a fixed learning rate of 1e−5 achieves better accuracy than the model with a scheduler learning rate as followed by [7]. The configuration of the learning rate for different layers in the model has the potential to affect model accuracy, thus a new configuration could be explored in future work.

In experiments Enh_GHO, Enh_G, Enh_GHO_v2, and Enh_G_v2, two versions of gaze data are used for training. The results show that the model trained with gaze version 2 performs better than the model trained with gaze version 1. The quality of collected gaze tracking data significantly affects the model's performance, especially when only gaze information is used. The two additional experiments on the variants of the enhanced EgoViT show that the modified DCTG and PADM modules, as well as the input configuration of the head, do not improve the model's performance. One reason could be that the information in the class token, i.e., the gaze and hand-object features, are already exchanged with the normal token in the 3D Shifted MSA in the Video Swin Transformer. Therefore, even the gaze and hand-object features are processed independently in the short-term and long-term stages, the class token has the surfficient information for classification.

A noticeable obeservation is all experiments have a mean accuracy under 41%, and the top-1 accuracy is only around 50%. The sesults achieves lower performance than the Video Swin Transformer trained on other datasets. A plausible reason for this is that the EGTEA Gaze+ dataset is more challenging than other datasets. This dataset has an imbalanced distribution of samples in each class, with some classes having fewer than 40 samples. While class label 1 has about 600 samples. The model may not able to effectively learn the features of classes with fewer samples.

Although the dataset is challenging, the results of the experiment Enh_GHO_v2 shows a promising improvement in the performance of EAR. The enhanced EgoViT model trained with gaze-hand-object features achieves a top-1 accuracy of 52.0% and a top-5 accuracy of 76.3%. These results are higher than those of the original EgoViT model trained with hand-object features. This demonstrates that integrating additional gaze information can potentially improve EAR performance. The work in this thesis provides a foundation for future research on the use of gaze information in EAR.

Table 4.7: Test Results of Various Experiments

Experiment ID	Top-1 Acc.(%)	Top-5 Acc.(%)	Mean Class Acc.(%)
Orig_HO_no_pretrain	51.5	78.5	38.8
Orig_HO	51.7	75.2	40.6
Orig_HO_sched. LR	48.4	74.4	35.8
Enh_ GHO	51.4	76.7	40.0
Enh_G	48.9	75.1	37.7
Enh_GHO_v2	52.0	76.3	38.7
Enh_G_v2	50.0	76.2	38.0
Enh_v2_GHO_v2	50.4	75.8	38.3
Enh_v3_HO	51.1	77.4	40.5
Enh_v3_GHO_v2	51.2	77.9	39.2
Enh_ v3_G_v2	50.8	75.9	37.8

Chapter 5

Summary

5.1 Conclusion

In this thesis, a Gaze-Enhanced DCTG module for the EgoViT model was proposed to improve action recognition performance in egocentric videos. The Gaze-Enhanced DCTG is designed to focus on additional regions of interest in the video frames, identified by gaze tracking points and detected hands and objects. Compared to the original EgoViT, the proposed Gaze-Enhanced EgoViT incorporates gaze tracking points as an additional input to the model. The Gaze-Enhanced DCTG module extracts gaze features from the gaze tracking points and input frames, merging them with hand-object features from a modified HOD module to generate dynamic class tokens. These dynamic class tokens guide the model to focus on regions enriched with gaze and hand-object information.

The proposed Gaze-Enhanced EgoViT model retains a similar architecture to the original EgoViT model, consisting of G groups in the short-term stage, a class token Merging module, and a long-term stage. This architecture effectively handles temporal relationships between short-term phases. To process local relationships along the spatial dimension, a modified Video Swin Transformer is used as the backbone of the proposed model.

The proposed Gaze-Enhanced EgoViT model was evaluated on the EGTEA Gaze+ dataset, a comprehensive dataset for first-person view actions with gaze tracking points. The experimental results demonstrate that the proposed Gaze-Enhanced EgoViT model achieves a top-1 accuracy of 52.0% and a top-5 accuracy of 76.3%, representing an improvement over the baseline EgoViT model. The model was trained and tested on two versions of gaze data. Gaze version 2, with better

gaze feature quality, yielded better performance, suggesting that the quality of gaze features significantly impacts the model's performance. Those results suggest that the quality of gaze features plays a significant role in the model's performance. The proposed Gaze-Enhanced EgoViT model is a promising approach to improving action recognition in egocentric videos. Additionally, experiments conducted on the model with only gaze features showed that it achieves a top-1 accuracy above 50%, indicating that gaze information is useful for action recognition in egocentric videos. Further investigation into the impact of gaze information on EAR is warranted.

5.2 Future Works and Outlook

Futere work: The experimental results indicate that the proposed Gaze-Enhanced EgoViT model is a promising approach for improving action recognition performance in egocentric videos. To significantly enhance the model's performance, improving the quality of gaze features should be explored. There are three possibilities for this:

1. Training and evaluating the model on a dataset with more accurate collected gaze tracking points.

2. Studying the impact of gaze region size on EAR. Conducte experiments with different gaze-box sizes to determine the optimal size for the model.

3. Using pretrained networks, such as autoencoders, to extract gaze features. An autoencoder can be trained to extract features from images and reconstruct the images, accuiring higher quality extracted features.

Outlook in the field of battery production: The rapid growth of electric vehicles has increased the demand for batteries, making their production a key technology for value creation. As the demand and production of electric vehicles rise, so does the need for high-quality batteries. Gigafactories are being built to meet this demand, optimizing production quality, performance, and cost. Artificial intelligence technologies, such as computer vision and machine learning, are increasingly used in battery production to enhance quality. Niri et al. [40] combined machine learning to create predictive models for battery performance, linking lab-scale manufacturing to pilot-line production and supporting smarter and cleaner electrode production for high-quality Li-ion batteries. Smart factories are equipped with many industrial robotic arms, and ensuring the safety of interactions between

workers and robotic arms is crucial. Suh et al. [41] presented a novel wearable sensing prototype integrating IMU and body capacitance sensors to recognize worker activities in manufacturing.

These studies show that artificial intelligence technologies have broad prospects in battery production. The proposed Gaze-Enhanced EgoViT model has the potential to be applied in various ways to improve production quality. For example, it could be used to monitor robotic arm activities to ensure process accuracy and avoid collisions with workers. Since the model can handle egocentric videos, the monitoring camera could be directly mounted on the moving part of the robotic arm, providing a first-person view and enhancing the environmental perception of the robotic arm. Another application could be wearable equipment for workers, such as glasses with cameras and gaze tracking functions. The proposed Gaze-Enhanced EgoViT model can recognize worker activities and identify which part of the production process the worker is engaged in, improving assembly quality by detecting anomalies in real-time. The Transformer-based model is suitable for handling varying sequence video lengths, making it ideal for real-time monitoring of different production processes performed by workers. The model has the potential to understand the relationships between different production processes and provide feedback to workers in real-time, thereby improving production efficiency and quality.

Acronyms

CNNs convolutional neural networks. 7

DCT Dynamic Class Token. 16

DCTG dynamic class token generator. 1

EAR egocentric action recognition. 3

FFN feed-forward network. 13

FPV first person view. 28

HOD Hand and Object Detector. 5

HRI Human-Robot Interaction. 3

LN Layer Normalization. 13

LSTM long short-term memory. 20

MSA multi-Head Self-Attention. 13

NLP natural language processing. 9

PADM Pyramid Architecture with a Dynamic Merging. 3

TSM termed temporal shift module. 8

TSN Temporal Segment Networks. 8

ViT Vision Transformer. 2

ViViT Video Vision Transformer. 12

VR Virtual Reality. 3

List of Tables

List of Figures

Bibliography

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016), pp. 770–778.

[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, *Densely Connected Convolutional Networks*, arXiv:1608.06993 [cs], Jan. 2018.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, arXiv:2010.11929 [cs], June 2021.

[4] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, *ViViT: A Video Vision Transformer*, arXiv:2103.15691 [cs], Nov. 2021.

[5] G. Bertasius, H. Wang, and L. Torresani, *Is Space-Time Attention All You Need for Video Understanding?* arXiv:2102.05095 [cs], June 2021.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*, arXiv:2103.14030 [cs], Aug. 2021.

[7] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, *Video Swin Transformer*, arXiv:2106.13230 [cs], June 2021.

[8] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, "Understanding Human Hands in Contact at Internet Scale," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020), pp. 9866–9875.

[9] C. Pan, Z. Zhang, S. Velipasalar, and Y. Xu, *EgoViT: Pyramid Video Transformer for Egocentric Action Recognition*, arXiv:2303.08920 [cs], Mar. 2023.

[10] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," en, Trends in Cognitive Sciences **9**, 188–194 (2005).

[11] M. Land, N. Mennie, and J. Rusted, "The Roles of Vision and Eye Movements in the Control of Activities of Daily Living," en, Perception **28**, 1311–1328 (1999).

[12] Y. Huang, M. Cai, Z. Li, and Y. Sato, *Predicting Gaze in Egocentric Video by Learning Task-dependent Attention Transition*, arXiv:1803.09125 [cs], Dec. 2018.

[13] H. R. Tavakoli, E. Rahtu, J. Kannala, and A. Borji, "Digging Deeper Into Egocentric Gaze Prediction," in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (Jan. 2019), pp. 273–282.

[14] B. Lai, M. Liu, F. Ryan, and J. M. Rehg, *In the Eye of Transformer: Global-Local Correlation for Egocentric Gaze Estimation*, arXiv:2208.04464 [cs], Aug. 2022.

[15] Y. Li, M. Liu, and J. M. Rehg, *In the Eye of the Beholder: Gaze and Actions in First Person Video*, arXiv:2006.00626 [cs], Oct. 2020.

[16] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, C. Fuegen, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, *Ego4D: Around the World in 3,000 Hours of Egocentric Video*, arXiv:2110.07058 [cs], Mar. 2022.

[17] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li, *A comprehensive study of deep video action recognition*, 2020.

[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks,"

in 2014 IEEE Conference on Computer Vision and Pattern Recognition (June 2014), pp. 1725–1732.

[19] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: towards good practices for deep action recognition," CoRR **abs/1608.00859** (2016).

[20] J. Lin, C. Gan, and S. Han, *TSM: Temporal Shift Module for Efficient Video Understanding*, arXiv:1811.08383 [cs], Aug. 2019.

[21] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines," IEEE Transactions on Pattern Analysis and Machine Intelligence **43**, 4125–4141 (2021).

[22] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, *Charades-Ego: A Large-Scale Dataset of Paired Third and First Person Videos*, arXiv:1804.09626 [cs], Apr. 2018.

[23] R. Herzig, E. Ben-Avraham, K. Mangalam, A. Bar, G. Chechik, A. Rohrbach, T. Darrell, and A. Globerson, *Object-Region Video Transformers*, arXiv:2110.06915 [cs], June 2022.

[24] X. Wang, Y. Wu, L. Zhu, and Y. Yang, "Symbiotic Attention with Privileged Information for Egocentric Action Recognition," Proceedings of the AAAI Conference on Artificial Intelligence **34**, 12249–12256 (2020).

[25] Y. Huang, M. Cai, and Y. Sato, "An Ego-Vision System for Discovering Human Joint Attention," IEEE Transactions on Human-Machine Systems **50**, 306–316 (2020).

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention Is All You Need*, arXiv:1706.03762 [cs], Aug. 2023.

[27] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," CoRR **abs/1711.07971** (2017).

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," CoRR **abs/2005.12872** (2020).

[29] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, *Video Transformer Network*, arXiv:2102.00719 [cs], Aug. 2021.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, arXiv:1409.0575 [cs], Jan. 2015.

[31] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, *The Kinetics Human Action Video Dataset*, arXiv:1705.06950 [cs], May 2017.

[32] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in 2017 ieee international conference on computer vision (iccv) (2017), pp. 843–852.

[33] K. Kato, Y. Li, and A. Gupta, "Compositional Learning for Human Object Interaction," en, in *Computer Vision – ECCV 2018*, Vol. 11218, edited by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Series Title: Lecture Notes in Computer Science (Springer International Publishing, Cham, 2018), pp. 247–264.

[34] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, *Object Level Visual Reasoning in Videos*, arXiv:1806.06157 [cs], Sept. 2018.

[35] B. Xu, Y. Wong, J. Li, Q. Zhao, and M. S. Kankanhalli, "Learning to Detect Human-Object Interactions With Knowledge," en, in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019), pp. 2019–2028.

[36] N. Shvetsova, B. Chen, A. Rouditchenko, S. Thomas, B. Kingsbury, R. Feris, D. Harwath, J. Glass, and H. Kuehne, "Everything at Once – Multi-modal Fusion Transformer for Video Retrieval," en, in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2022), pp. 19988–19997.

[37] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, arXiv:1506.01497 [cs], Jan. 2016.

[38] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: the epic-kitchens dataset," in European conference on computer vision (eccv) (2018).

[39] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, arXiv:1711.05101 [cs, math], Jan. 2019.

[40] M. F. Niri, K. Liu, G. Apachitei, L. R. Ramirez, M. Lain, D. Widanage, and J. Marco, "Machine learning for optimised and clean li-ion battery manufacturing: revealing the dependency between electrode and cell characteristics," Journal of Cleaner Production **324**, 129272 (2021).

[41] S. Suh, V. F. Rey, S. Bian, Y.-C. Huang, J. M. Rožanec, H. T. Ghinani, B. Zhou, and P. Lukowicz, *Worker activity recognition in manufacturing line using near-body electric field*, 2023.