



ARTICLE

Joint Smoothed Cross-Entropy and Soft-DTW Learning for TimeSformer-Based Video Action Recognition

Riski Nur Azizah¹ and Bong-Soo Sohn^{2,*}

¹Department of Computer Engineering, Chung-Ang University, Seoul, Republic of Korea

²School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea

*Corresponding Author: Bong-Soo Sohn. Email: bongbong@cau.ac.kr

Received: 05 June 2026; Accepted: 19 August 2026; Published: 22 October 2026

ABSTRACT: Video-based action recognition remains challenging because of variations in action execution speed and overfitting to static spatial cues. Although the TimeSformer architecture effectively captures long-range spatiotemporal dependencies, the standard cross-entropy loss lacks mechanisms to align dynamic temporal sequences. In this study, we propose a novel joint optimization framework that combines label-smoothed cross-entropy with Soft Dynamic Time Warping (Soft-DTW). Each input frame is divided into fixed-size patches that are embedded as tokens. Spatial attention captures structures and object appearance, while temporal attention learns how these patches evolve over time, enabling robust modeling of gesture dynamics. By factorizing the attention into separate space and time operations, TimeSformer efficiently captures subtle motion cues, and label smoothing reduces prediction overconfidence and reliance on static visual cues. Our primary contribution is the use of Soft-DTW as a temporal regularizer to reduce temporal misalignment between predicted temporal features and class-specific temporal prototypes. To evaluate the framework in a dynamic, domain-specific setting, we constructed a virtual reality (VR) gesture dataset comprising tapping and directional swiping actions. The evaluation results obtained using five-fold cross-validation demonstrate improved classification accuracy. We also conducted an extensive evaluation across standard action-recognition datasets such as Joint-annotated Human Motion Data Base (JHMDB), Human Motion Database 51 (HMDB51), University of Central Florida action recognition dataset (UCF101), and a 5% subset of the Kinetics human action video datasets (Kinetics-400/600). Experimental results show significant improvements in top-1 recognition accuracy on these public datasets. Furthermore, our proposed method demonstrates robust generalization in low-data regimes where TimeSformer typically underperforms.

KEYWORDS: Video action recognition; TimeSformer; joint loss optimization; temporal alignment loss; cross-entropy; label smoothing regularization; Soft-DTW

1 Introduction

Human Action Recognition (HAR) is a fundamental task in computer vision. Unlike static image classification, robust HAR in video sequences remains challenging. HAR requires a model that can interpret temporal dynamics and decode temporal movements across frames. Prior to the advent of deep learning, HAR relied heavily on hand-crafted features, such as dense trajectories [1], Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and Motion Boundary Histograms (MBH). Convolutional Neural Networks (CNNs) served as the dominant backbone during the early development of deep learning-based computer vision. Unlike traditional fully connected networks, CNNs are designed to process data with a known grid-like topology, such as images. In classic CNNs, initial layers employ learnable filters called

kernels to detect low-level geometric features, such as edges and curves, in their learning process [2]. Early video-recognition approaches based on 2D CNNs processed individual frames independently, discarding critical temporal evolution entirely unless combined with post-hoc aggregation methods [3]. Applying 2D CNNs to temporal data, such as videos or gesture sequences, also presents significant challenges. To address these limitations, 3D CNNs (C3D) were introduced, extending kernels into the temporal dimension to capture motion and appearance simultaneously [4]. However, these architectures remain constrained by their limited temporal receptive fields. The 3D convolutional kernel can only capture relationships within a small fixed window of adjacent frames. Architectures such as the Multi-stage linear-index dilated TCN (MSLID-TCN) [5] leverage multi-stage dilated convolutions to efficiently compress temporal features and align long-range sequences, avoiding the heavy computational burden of artificially stacking standard kernels.

Transformer models were originally designed for Natural Language Processing (NLP) [6], relying on a self-attention architecture. Motivated by the success of Transformers in NLP on text-based datasets, the Vision Transformer (ViT) [7] adapted this approach to computer vision. ViT challenged the dominance of CNN-based architectures by representing an image as a sequence of embedded patches. ViT enables more efficient training in the still-image domain compared to CNNs, with the possibility of building a larger learning capacity within the same computational budget [7–9]. Originally introduced for video understanding, TimeSformer [10] utilizes a Divided Space-Time Attention to capture long-range dependencies with better computational efficiency than traditional 3D CNNs. TimeSformer builds upon the ViT foundation while providing a scalable solution for the temporal dimension. The vast majority of video classification models, including the standard TimeSformer, are trained using cross-entropy (CE). However, standard CE optimizes categorical prediction accuracy without explicitly accounting for temporal alignment between video sequences. If the input gesture is slightly faster or slower than the training examples, the model may fail to align the features correctly, leading to misclassification.

To mitigate the overconfidence and overfitting issues inherent in standard CE loss, Label Smoothing (LS) offers robust regularization by smoothing the hard target distribution [11]. In standard CE optimization, target distributions are represented as one-hot vectors. This forces the network to adjust the logits of the predicted class toward infinity while adjusting other logits toward negative infinity. In the HAR field, human actions can contain overlapping visual and temporal characteristics, and CE may encourage the model to memorize overly specific spatial features, thereby limiting its ability to generalize. Label smoothing (LS) techniques were introduced to counter this problem by smoothing the hard target distribution [12]. Label smoothing can improve generalization by preventing the model from becoming excessively confident in the ground-truth class. With a smoothing parameter ϵ and K total classes, LS modifies the hard target distribution by assigning a probability of $1 - \epsilon + \frac{\epsilon}{K}$ to the ground-truth class, while the remaining probability mass is uniformly distributed such that each of the $K - 1$ incorrect classes receives $\frac{\epsilon}{K}$. Müller et al. [12] show how LS restricts the dominant logit spikes by distributing the penalty evenly to the non-target classes. In the spatiotemporal modeling of TimeSformer, label smoothing can improve generalization in low-data recognition tasks. However, even though LS is effective in mitigating overconfidence, it still fails to reflect the elasticity of human action across frames. To address this limitation, we incorporate Soft Dynamic Time Warping (Soft-DTW) as a differentiable temporal alignment objective. Classic Dynamic Time Warping (DTW) is a dynamic programming algorithm used to find the optimal alignment between two temporal sequences, regardless of speed [13]. Standard DTW is non-differentiable and therefore cannot be directly optimized through gradient-based learning, rendering it unusable for training deep neural networks via backpropagation. Its successor, Soft-DTW, provides a fully differentiable loss function that can compute the soft minimum of all alignment costs and penalize temporal misalignments [14].

In this study, we replace the standard cross-entropy loss of the original TimeSformer with a label-smoothed cross-entropy loss and complement it with a Soft-DTW objective. This approach helps the network learn video representations that can be temporally aligned with the learned class prototypes. LS softens the hard target distribution, while Soft-DTW penalizes the temporal misalignment between the predicted gesture and the ground-truth dynamics. We evaluate the proposed objective on multiple action-recognition benchmarks to assess its effectiveness in reducing classification uncertainty and temporal misalignment. Our main contribution is a joint objective that combines label smoothing for classification regularization with differentiable sequence alignment for temporal consistency. We further demonstrate that the combination of the loss formulations improves model generalization and data efficiency, with experimental results showing that the proposed objective outperforms standard cross-entropy training in low-data regimes. Finally, we introduce a custom Virtual Reality (VR) boxing gesture dataset to capture dynamic temporal warping and variable-speed human gestures. We show that the additional Soft-DTW in the classification system significantly enhances TimeSformer, outperforming the baseline models in recognizing rhythmic game input gestures, particularly in scenarios involving variable execution speeds common in VR rhythm games (i.e., tapping and swiping gestures with direction).

2 Related Work

Sequence modeling tasks were first addressed by Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Standard RNNs are susceptible to vanishing gradients, whereas LSTMs mitigate this problem through gated memory mechanisms to regulate temporal memory [15]. The introduction of the Transformer marked a major shift in sequence modeling, which allowed models to capture long-range dependencies by modeling pairwise dependencies among tokens in a sequence through self-attention [6]. Building upon this success, the Vision Transformer (ViT) challenged the dominance of Convolutional Neural Networks (CNNs) by splitting static images into sequences of fixed-size patches (e.g., 16×16 pixels), embedding them linearly, and processing them with a standard Transformer encoder. A pure Transformer could achieve state-of-the-art performance on image classification, but applying it naively to videos with stacked frames results in computational complexity, as the attention mechanism attempts to compare every patch in every frame [7]. Fig. 1 illustrates how ViT represents an image as a sequence of patch tokens for Transformer-based classification. ViT decomposes a static input image into a sequence of non-overlapping, fixed-size patches. The patches are linearly projected into a flat embedding and a learnable classification token (CLS). CLS serves to aggregate global information from the entire sequence. The key step is the application of the Transformer Encoder, which utilizes Multi-Head Self-Attention (MSA) to model the patches simultaneously, regardless of their spatial distance. This model computes relationships solely within the spatial domain of a single static frame. Lastly, the final classification prediction is derived by passing the CLS token state through a Multi-Layer Perceptron (MLP) head. ViT perceives the input as a static snapshot, making it inherently unable to capture the motion dynamics and temporal evolution required to distinguish action gestures.

Bertasius et al. proposed the Time-Space Transformer (TimeSformer), extending ViT to temporal data with Divided Space-Time Attention. This architecture factorizes the operation into two efficient steps: temporal attention and spatial attention, which allow the model to capture the complex temporal dynamics required for gesture recognition (such as the trajectory of a hand swiping or tapping over time) while maintaining the global spatial context established by the original ViT [10]. The application of standard cross-entropy (CE) loss in ViT often encourages the model to be overconfident in its predictions, causing overfitting in data-scarce regimes [7]. Later, Touvron et al. introduced Data-efficient Image Transformers (DeiT), which

utilized label smoothing and mixup augmentation. In this process, the loss function is modified to predict a weighted combination of classes [11,16].

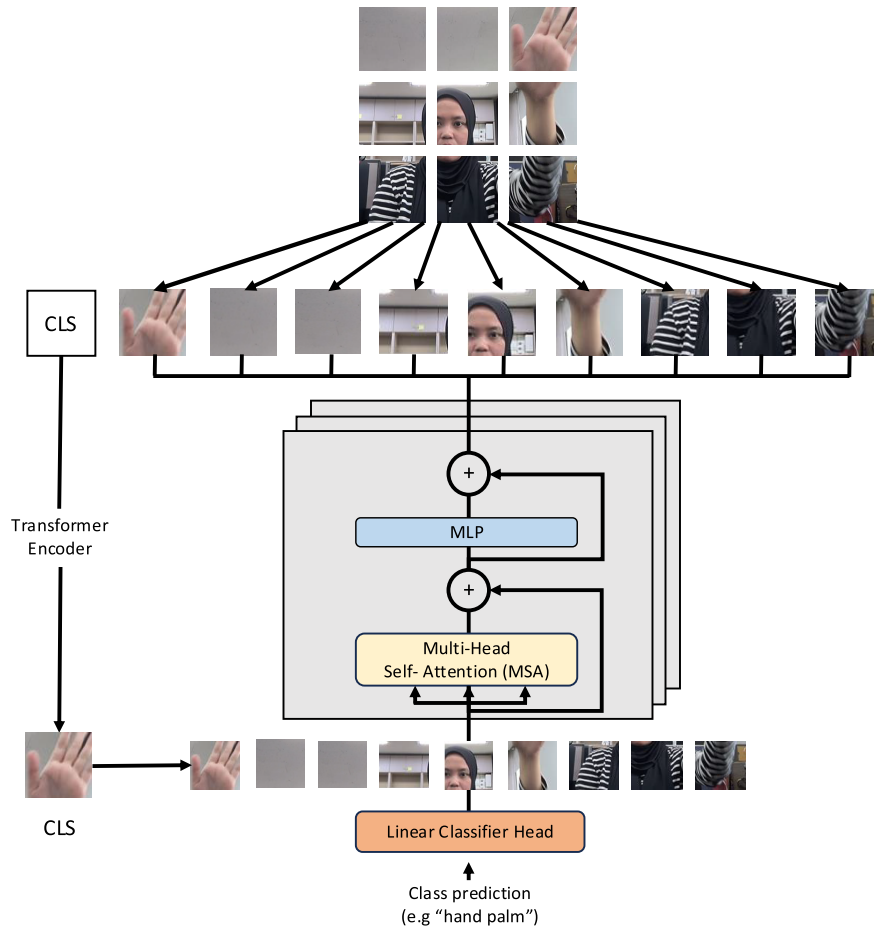


Figure 1: Application of the Transformer encoder and Multi-Head Self-Attention (MSA) in the Vision Transformer (ViT) architecture.

In dynamic gesture recognition, these distance metrics are insufficient because they fail to account for temporal warping when one gesture is performed faster than another. Soft-Dynamic Time Warping (Soft-DTW) has emerged as an alternative, providing a loss function that explicitly minimizes the alignment cost between sequences. Cai et al. [17] proposed DTWNet, which integrates learnable DTW kernels into CNNs to extract motion-invariant features. Ko et al. [18] recently utilized a differentiable alignment loss for video-text retrieval. This work demonstrates that soft alignment can bridge the gap between multi-modal temporal streams. More recently, large-scale video foundation models such as InternVideo [19] and its successor InternVideo3 [20], VideoMAE [21], and VideoMamba [22] have further improved temporal representation learning through self-supervised pretraining and efficient sequence modeling. These models have established state-of-the-art performance across multiple video benchmarks. Our work complements these advances by focusing not on designing a new backbone, but on improving the optimization objective through joint label smoothing and differentiable temporal alignment.

The TimeSformer architecture was originally evaluated on large-scale video understanding benchmarks, such as the Kinetics human action video datasets 400 (Kinetics-400) [23,24] and Something-Something

V2 [10,11,16]. The Kinetics-400 dataset contains more than 300,000 video clips of human actions, such as playing musical instruments, participating in sports, and performing daily activities and has become a standard benchmark for learning generic motion representations. However, these datasets largely consist of third-person perspectives mined from YouTube videos. While effective for general action classification, they lack the fine-grained hand articulation details required for precise gesture interfaces. In smaller human action datasets, such as Human Motion Database 51 (HMDB51) [25] and University of Central Florida action recognition dataset (UCF101) [26], models can often achieve deceptively high accuracy by exploiting static spatial background cues rather than genuinely learning the complex temporal dynamics of the actions. A more fine-grained spatiotemporal dataset is Joint-annotated Human Motion Data Base (JHMDB), which provides 2D articulated tracking and focuses on complex human actions [27], making it a good benchmark to evaluate temporal inductive biases in low-data regimes.

3 Methodology

Fig. 2 presents an overview of the proposed TimeSformer-based framework. We propose a hybrid classification loss using label-smoothed cross entropy and Soft-DTW. Following the Vision Transformer (ViT) paradigm, this architecture is built to capture spatiotemporal dependencies.

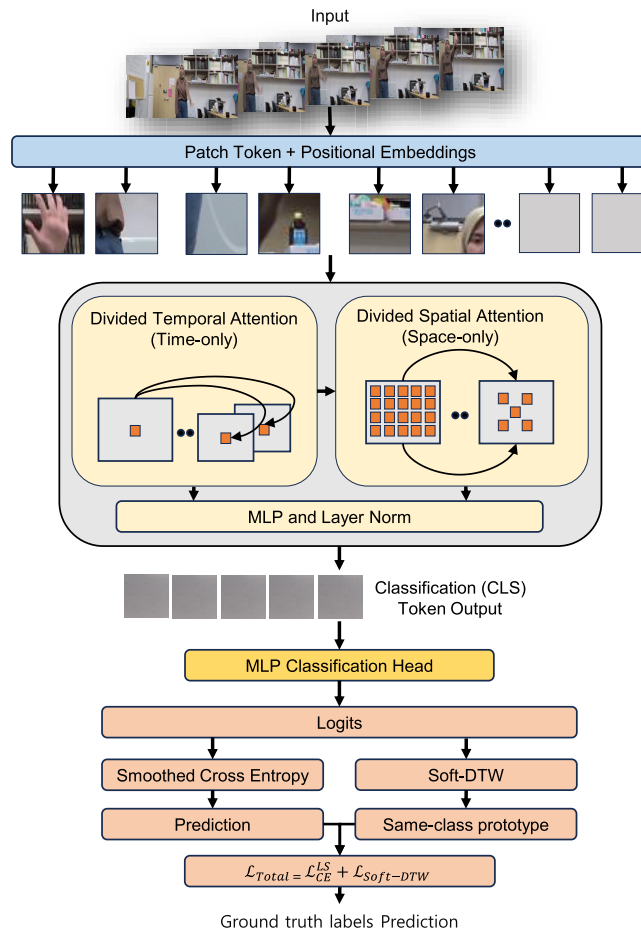


Figure 2: The architecture of the proposed method. We use the combined objective function of smoothed CE and Soft-DTW to maximize gesture recognition while minimizing temporal discrepancies.

The input is a video clip $X \in \mathbb{R}^{H \times W \times 3 \times F}$, which consists of F frames with a resolution of $H \times W$ and three color channels (RGB). To process it into a Transformer backbone, we decompose each frame into N non-overlapping patches of size $P \times P$, where each patch $x_{(p,t)}$ has dimension of $3P^2$. Each patch is projected into a D -dimensional embedding space with $\mathbf{x}_{(p,t)} \in \mathbb{R}^{3P^2}$, where $p = 1, \dots, N$ is a patch index and $t = 1, \dots, F$ is a temporal index. With a learnable weight matrix $E \in \mathbb{R}^{3P^2}$, an embedding vector $\mathbf{e}_{(p,t)}^{pos}$ is added to each patch. Based on Transformer architecture, a Classification token (CLS) serves to aggregate global video information across the layers for the final classification task. This token is denoted as $\mathbf{z}_{(0,0)}^0$ resulting in the input sequence $\mathbf{Z}^{(0)}$ to the Transformer encoder defined as:

$$\mathbf{Z}^{(0)} = [\mathbf{z}_{cls}, \dots, E\mathbf{x}_{(p,t)} + \mathbf{e}_{(p,t)}^{pos}, \dots] \quad (1)$$

where $p \in \{1, \dots, N\}$ and $t \in \{1, \dots, F\}$.

This results in an input tensor with $\mathbb{R}^{(NF+1) \times D}$ that is subsequently processed by Divided Space-Time Attention blocks. The first stage is the Temporal Attention block, which is designed to capture the motion dynamics between frames. The first encoded tokens $\mathbf{Z}^{(l-1)}$ are processed, and this temporal attention mechanism compares tokens at the same spatial location p across the frames F . Let $Q = Q_{(p,t)}$, $K = K_{(p,t)}$ and $V = V_{(p,t)}$ denote the query, key and value vectors from layers, respectively. The attention mechanism aligns tokens at the same spatial position as the output of each head, computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

where $\sqrt{d_k}$ is a scaling factor to prevent vanishing gradients. This scaling operation is an integral part of the scaled dot-product attention computation performed independently within each parallel head of the Multi-Head Attention (MHA) module.

The softmax function converts the similarity scores into a probability distribution, ensuring that the attention weights sum to one. This step assigns larger weights to temporally relevant tokens to selectively aggregate the temporal information. The next step is spatial attention, which captures spatial relationships among patches within each frame and is strictly confined to the spatial domain. This means that if we consider a specific patch in frame t , the system only attends to other patches within the same frame t . Spatial attention is computed using Eq. (2), where $(Q, K, V) \in \mathbb{R}^{N \times d}$ denote the tokens on the single frame.

The illustration of the Temporal and Spatial Attention architecture is detailed in Fig. 3. In the temporal attention block, the pink query path at spatial location p in frame t only compares the patch with the same location of p across the adjacent frames (blue key patches). This step refers to Eq. (2), where $Q_{(p,t)}$ attends to $K_{(p,t')}$ and $V_{(p,t')}$, for $t' \in \{1, \dots, F\}$ and forces the model to track the temporal data evolution without spatial distraction. Meanwhile, the spatial attention mechanism attends to all other patches within the frame. As illustrated, the blue patches in the spatial-attention module illustrate dense connections within a single frame. This represents the model learning relationships among different objects captured in a single frame and time (e.g., hand vs. head). This refers to Eq. (2), where the spatial attention mechanism is computed exclusively across the N spatial patches within a fixed time frame t .

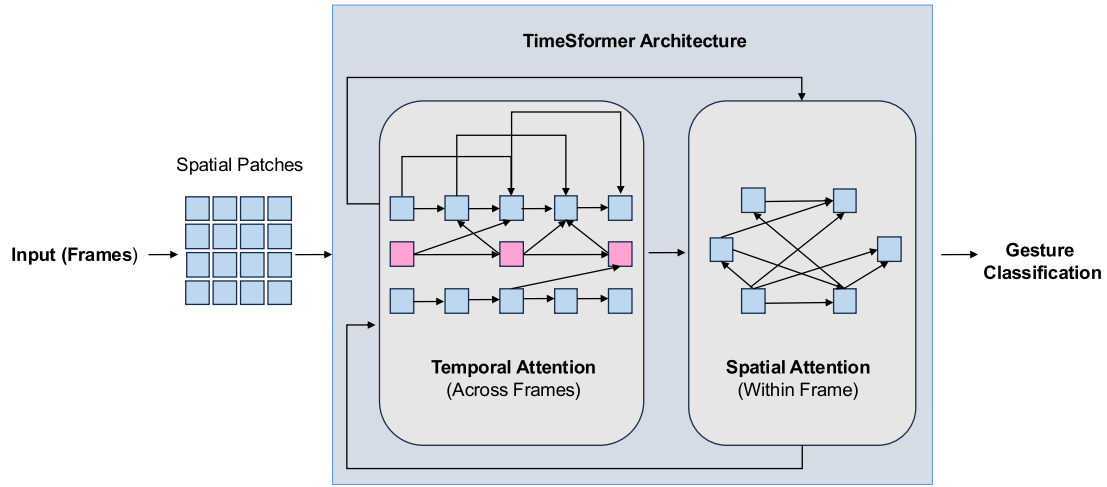


Figure 3: Overview of the divided space-time attention mechanism, consisting of temporal and spatial attention blocks. The temporal attention block captures the motion dynamics, and the Spatial Attention block captures the spatial attention.

The feature representations are processed by a standard Transformer encoding block with residual connection and layer normalization (LN). Residual connections help mitigate gradient degradation during deep network training. Meanwhile, Layer Normalization (LN) stabilizes intermediate feature distributions (e.g., zero mean and unit variance), preventing feature values from exploding or vanishing across deep layers. Define $\mathbf{x}_{in}$ as the input tensor and $\mathbf{x}_{attn}$ as the output of the divided space-time attention to calculate the residual connection:

$$\mathbf{x}_{attn} = \mathbf{x}_{in} + \text{Attention}(LN(\mathbf{x}_{in})) \quad (3)$$

We employ the Gaussian Error Linear Unit (GELU) activation function, with the final output $\mathbf{y}$ computed as:

$$\mathbf{y} = \mathbf{z}' + \text{MLP}(LN(\mathbf{z}')) \quad (4)$$

where MLP applies a point-wise transformation, expanding its dimension in the hidden layer before returning to the original embedding size. Lastly, to perform action recognition, we use a special classification token $\mathbf{x}_{cls}$ to aggregate the global information from the entire clip and pass the vector through the MLP head. This head maps the embedding vector to the target classes. The predicted probability distribution p is obtained as:

$$p = \text{softmax}(W\mathbf{x}_{cls} + b) \quad (5)$$

where p represents the vector of class probabilities, W is the learnable weight matrix of the classifier, and b is the bias term.

The standard cross-entropy (CE) loss can measure the divergence between p and the ground-truth distribution y . For a single video clip sample with C classes, the CE is obtained as:

$$\mathcal{L}_{CE} = - \sum_{c=1}^C y_c \log(p_c) \quad (6)$$

where y_c is a binary indicator to check if the class c is the correct classification and p_c is the predicted probability of c .

$$\mathcal{L}_{LS} = - \sum_{k=1}^K q_{\text{smooth}}(k) \log p(k) \quad (7)$$

Fig. 4 shows the computational flow of CE with label smoothing regularization. To reduce overconfidence and improve the calibration of the probability estimation, we apply label smoothing to the CE baseline. In this process, label smoothing acts as a regularizer that reduces overfitting, particularly in low-data regimes. The label-smoothed CE loss computes the weighted cross-entropy between the predicted distribution and the smoothed target distribution. The true class y is assigned as the separated loss to force the network probability $p(y)$ to increase while calculating the error of the correct class. Standard CE has a strict zero target distribution for incorrect classes, meaning that there is no penalty for leading the target probabilities to absolute zero. In the LS improvement, we set $k \neq y$ as the incorrect classes:

$$\mathcal{L}_{LS} = - \left(1 - \varepsilon + \frac{\varepsilon}{K}\right) \log p(y) - \sum_{k \neq y} \frac{\varepsilon}{K} \log p(k) \quad (8)$$

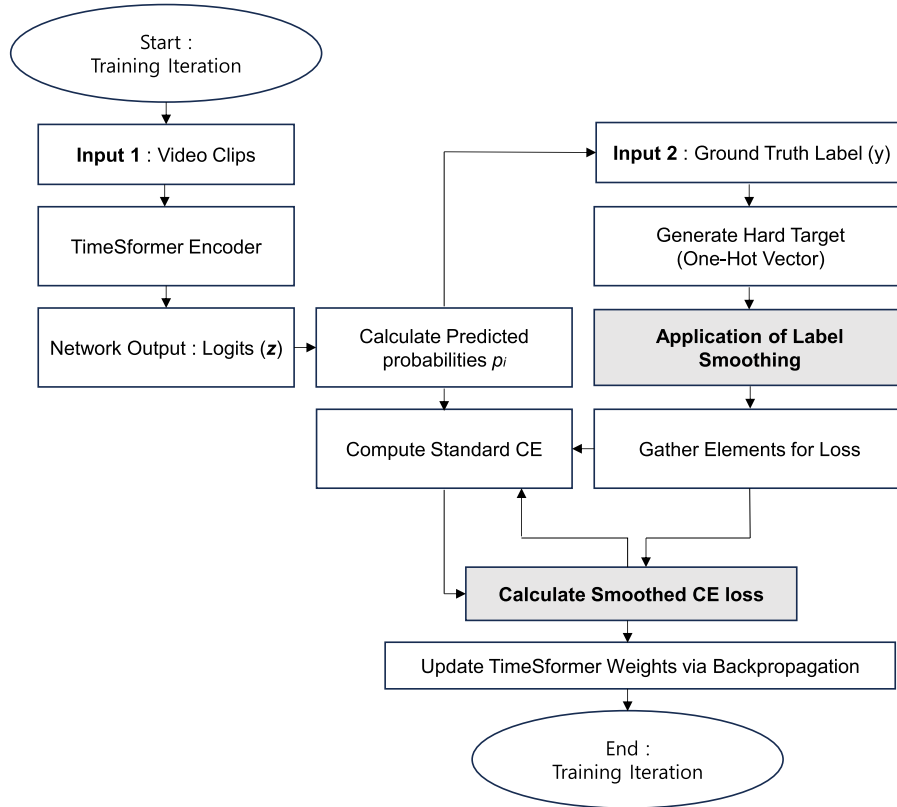


Figure 4: Computational flow of label-smoothed CE in the TimeSformer model.

Finally, we rearranged these terms as label-smoothed CE, with a smoothing parameter ε :

$$\mathcal{L}_{CE}^{LS} = (1 - \varepsilon) \mathcal{L}_{CE} + \varepsilon \mathcal{H}_{uniform} \quad (9)$$

where $\mathcal{L}_{CE} = -\log p(y)$ as the standard CE loss, and $\mathcal{H}_{uniform}$ is defined as:

$$\mathcal{H}_{uniform} = -\frac{1}{K} \sum_{k=1}^K \log p(k) \quad (10)$$

where $\mathcal{H}_{uniform}$ represents the loss between a uniform distribution and the model output probabilities.

Soft-DTW [14] provides a differentiable measure of discrepancy between temporal sequences, allowing temporal alignment to be optimized through backpropagation. The Soft-DTW kernel is explained in Algorithm 1. Let $D \in \mathbb{R}^{N \times M}$ denote the pairwise cost matrix where each $D_{(i,j)}$ is the pairwise distance of the temporal features. The squared Euclidean distance is obtained as:

$$D_{(i,j)} = \|x_i - y_j\|^2 \quad (11)$$

Algorithm 1: Soft-DTW forward accumulation kernel

Require: Cost matrix $\Delta \in \mathbb{R}^{T \times T}$, smoothing parameter $\gamma > 0$

Ensure: Soft-DTW alignment cost $sdw_{\gamma}(\mathbf{seq1}, \mathbf{seq2})$

```

1: Initialize  $R \in \mathbb{R}^{(T+1) \times (T+1)}$ 
2:  $R[0, 0] \leftarrow 0$ 
3: for  $i = 1$  to  $T$  do
4:    $R[i, 0] \leftarrow +\infty$ 
5: end for
6: for  $j = 1$  to  $T$  do
7:    $R[0, j] \leftarrow +\infty$ 
8: end for
9: for  $i = 1$  to  $T$  do
10:  for  $j = 1$  to  $T$  do
11:    // Costs from insertion, deletion, and match paths
12:     $a \leftarrow R[i-1, j]$ 
13:     $b \leftarrow R[i, j-1]$ 
14:     $c \leftarrow R[i-1, j-1]$ 
15:     $R[i, j] \leftarrow \Delta_{i,j} + \text{soft\_min}_{\gamma}(a, b, c)$ 
16:  end for
17: end for
18: return  $R[T, T]$ 

```

Following the spatial aggregation of tokens from Divided Space-Time Attention blocks, the extracted temporal features form the predicted sequence, defined as $S_{pred} \in \mathbb{R}^{F \times D}$. The corresponding same-class prototype, which acts as the alignment target, is maintained in the same latent space, $S_{target} \in \mathbb{R}^{F \times D}$. This enables the framework to construct a valid cost matrix $\Delta \in \mathbb{R}^{F \times F}$ for the Soft-DTW kernel, ensuring meaningful regularization of the temporal dynamics.

For each class $c \in \{1, \dots, K\}$, the temporal prototype P_c is initialized during the first training epoch using the predicted temporal sequence S_{pred} extracted from the first encountered training sample of that specific class:

$$P_c^{(0)} = S_{pred} \quad (12)$$

When a new video sequence of class c is processed during subsequent training iterations t , its extracted temporal feature S_{pred} is used to iteratively update the corresponding class prototype P_c via a momentum-based moving average:

$$P_c^{(t)} = \beta P_c^{(t-1)} + (1 - \beta) S_{pred} \quad (13)$$

where $\beta \in [0, 1)$ is the momentum parameter that controls the update rate. During the joint loss computation, the Soft-DTW kernel aligns the current prediction against this stabilized prototype:

$$S_{target} = P_c^{(t-1)} \quad (14)$$

With our joint learning strategy, the proposed Soft-DTW works as an auxiliary regularizer that enforces temporal consistency among sequences. To establish reliable target alignments for the Soft-DTW branch, we employ a momentum-based global prototype construction. Rather than strictly per-batch or static per-class constructions, we maintain a continuous global prototype for each gesture class.

While conventional DTW minimizes the accumulated distance along the alignment path, Soft-DTW replaces the minimum operator with a soft-min operator using a smoothing parameter γ :

$$\min \gamma(z) - \gamma \log \sum_{(i)} \exp\left(\frac{-z_i}{\gamma}\right) \quad (15)$$

Under the Gibbs distribution, the Soft-DTW loss is defined as:

$$\mathcal{L}_{SDTW} = \min \gamma \left(\sum_{(i,j) \in \pi} D_{(i,j)} \right) \quad (16)$$

This loss function allows the network to learn temporal alignments via standard backpropagation, providing a smooth, differentiable approximation of the DTW discrepancy. We use a combined objective function to jointly optimize classification and temporal alignment. By optimizing these components, our model learns to maximize gesture recognition accuracy while minimizing temporal discrepancies. The resulting objective is defined as:

$$\mathcal{L} = \mathcal{L}_{CE}^{LS} + \lambda \mathcal{L}_{SDTW} \quad (17)$$

where λ is a hyperparameter that controls the importance of the temporal alignment constraint with a small positive value (e.g., 0.1) to prevent the auxiliary Soft-DTW term from dominating the primary classification objective. The network parameters are optimized through backpropagation by minimizing $\mathcal{L}$ over the training distribution. Algorithm 2 summarizes the joint optimization of smoothed CE and Soft-DTW in the TimeSformer backbone.

Algorithm 2: Joint optimization of smoothed CE with Soft-DTW in TimeSformer

Input: Video clip V , Hard label Y , S_{target} , Classes K , Smoothing parameter α , Soft-DTW weight λ , Learning rate η

Output: Optimized TimeSformer network weights θ

- 1: **while** Network has not converged **do**
 - 2: **1. Forward Pass (Feature Extraction)**
 - 3: Extract spatiotemporal tokens: $Z \leftarrow \text{TimeSformer}(V; \theta)$
-

(Continued)

Algorithm 2 (continued)

```

4: Predict class probabilities:  $P \leftarrow \text{Softmax}(\text{ClassificationHead}(Z))$ 
5: Extract temporal sequence features:  $S_{pred} \leftarrow \text{TemporalTokens}(Z)$ 
6: 2. Label Smoothing Regularization
7: for  $k = 1$  to  $K$  do
8:    $Y_{LS}[k] \leftarrow (1 - \alpha)Y[k] + \frac{\alpha}{K}$ 
9: end for
10: 3. Objective Function Computation
11: Compute Smoothed CE Loss:
12:    $\mathcal{L}_{CE}^{LS} \leftarrow -\sum_{k=1}^K Y_{LS}[k] \log(P[k])$ 
13: Compute Temporal Alignment Loss:
14:    $\mathcal{L}_{\text{Soft-DTW}} \leftarrow \text{Soft-DTW}(S_{pred}, S_{target})$ 
15: Compute Joint Loss:
16:    $\mathcal{L}_{\text{Total}} \leftarrow \mathcal{L}_{CE}^{LS} + \lambda \mathcal{L}_{\text{Soft-DTW}}$ 
17: 4. Backpropagation & Update
18: Compute gradients:  $g \leftarrow \nabla_{\theta} \mathcal{L}_{\text{Total}}$ 
19: Update weights:  $\theta \leftarrow \theta - \eta \cdot g$ 
20: end while
21: return  $\theta$ 

```

For feature preparation, we employ a strict preprocessing pipeline, which first removes the classification (CLS) token, mean-pools the remaining spatial patches, and normalizes the resulting vectors. With max-subtraction, we use the log-sum-exp trick for stable calculations.

4 Experimental Results

Training is conducted in two stages: linear probing and fine-tuning. During linear probing, only the classification head and, optionally, the LayerNorm parameters are optimized using label-smoothed CE. This stage establishes stable class separation before the temporal-alignment objective is introduced. During the fine-tuning stage, we activate the joint objective, in which the Soft-DTW weight λ is increased during the first epoch to prevent optimization instability. The smoothing parameter γ in Eq. (15) is linearly warmed up to the target value, which ensures smooth gradient propagation during training. Soft-DTW loss is not used during inference, resulting in runtime and memory complexity identical to those of the standard TimeSformer architecture.

To evaluate our proposed method, we constructed a custom video dataset for dynamic gesture recognition in a VR rhythm-game environment. The raw video clip sequences were recorded using a built-in laptop webcam at 30 fps and a resolution of 1920×1080 pixels. Frames were subsequently resized to 224×224 pixels with a spatial down-sampling step. This resizing reduces memory consumption and computational cost during training. The dataset consists of eight gesture classes, as detailed in Table 1. The small dataset was used to explicitly demonstrate our framework's robustness. To prevent identity leakage, our 5-fold cross-validation is strictly subject-independent. Subjects included in the training folds are completely excluded from the testing folds. This ensures that the network learns subject-independent temporal patterns rather than memorizing spatial identities. Our dataset consists of 794 video clip samples, with 15% used as the validation set. We sampled $F = 8$ frames for the input sequence $X \in \mathbb{R}^{224 \times 224 \times 3 \times 8}$.

Table 1: The gesture classes of our VR rhythm-game dataset.

Class Number	Class Name	Gesture Details
Class 1	swipe_bot_left	Swiping to the bottom direction with left hand
Class 2	swipe_bot_right	Swiping to the bottom direction with right hand
Class 3	swipe_left	Swiping to the left direction with left hand
Class 4	swipe_right	Swiping to the right direction with right hand
Class 5	swipe_top_left	Swiping to the top direction with left hand
Class 6	swipe_top_right	Swiping to the top direction with right hand
Class 7	tapping_left	Tapping using left hand
Class 8	tapping_right	Tapping using right hand

The experiments were conducted using the PyTorch deep learning framework and executed on an AMD Ryzen 5 5600X 6-Core processor and NVIDIA GeForce RTX 3060 Ti GPU with 8 GB VRAM. For the software environment, we utilized Python 3.12.7 and CUDA 12.9 (V12.9.41) to accelerate the GPU computation. We employed the Stochastic Gradient Descent (SGD) optimizer for training and implemented a standard learning rate schedule with a warmup phase, followed by cosine annealing decay. The hyperparameters used to train the proposed method are summarized in [Table 2](#).

Table 2: The hyperparameters for TimeSformer with label-smoothed CE and Soft-DTW loss in our VR rhythm game dataset.

Hyperparameter	Value	Description
Sequence Length (F)	8	The number of frames sampled per video clip
Image Size ($H \times W$)	224	Spatial resolution to which input frames are resized
Batch Size	4	The number of video samples in parallel during a single training
Weight Decay	0.05	L2 regularization coefficient to prevent overfitting
Head Learning Rate	1×10^{-4}	Learning rate applied to MLP
Fine-tuning LR	5×10^{-5}	Lower learning rate applied during fine-tuning
Freeze Epoch	5	The number of epochs for linear probing
Fine-tuning Epoch	10	The number of epochs for fine-tuning
Warm-up Epochs	1	Initial period during fine-tuning
λ Soft-DTW	0.1	The weight (λ) controlling the contribution of the Soft-DTW loss
Number of Folds	5	Number of cross-validation folds

We employed a confusion matrix M where M_{ij} is the number of samples belonging to class i predicted as class j . Overall accuracy is:

$$\text{ACC} = \frac{\sum_i M_{ii}}{\sum_i \sum_j M_{ij}}. \quad (18)$$

Given the limited size of our VR rhythm-game dataset, we employed five-fold cross-validation to evaluate the classification accuracy. The dataset is partitioned into five subsets, with four subsets used for training and one used for testing. This procedure is repeated until each subset has served as the test set, and the result is reported as the mean top-1 accuracy across the five folds. As shown in [Table 3](#), the baseline CE model achieved an accuracy of 84.62%, whereas the proposed method reached 87.69%. Note that Soft-DTW

is utilized exclusively during training and introduces no architectural modifications during inference. [Table 4](#) reports model complexity and inference speed.

Table 3: The classification accuracy (%) of CE, Smoothed CE and Smoothed CE+Soft-DTW integration applied in public benchmark datasets and our VR rhythm dataset.

Dataset	CE	Smoothed CE	Smoothed CE + SoftDTW (Proposed)	Improvement from CE (Percentage Points)
JHMDB	87.69%	88.43%	89.23%	+1.54%
HMDB51	69.34%	69.74%	69.67%	+0.33%
K-400	72.78%	73.13%	73.41%	+0.63%
5% K-400/600 subset	89.8%	90.20%	91.11%	+1.31%
UCF101	96.7%	98.28%	98.33%	+1.63%
VR Custom Dataset	84.62%	84.62%	87.69%	+3.07%

Table 4: Model complexity and inference performance on our VR custom dataset.

Metric	Value
Parameters	121.1M
FLOPs (per forward pass)	190.1 GFLOPs
Test Set Size	13 clips
Total Inference Time	27.9 s
Total Forward Pass	8
Throughput	0.40 clips/s
Latency	2142 ms/clip
Peak VRAM Usage	600–750 MB

[Fig. 5](#) illustrates the design of our VR boxing rhythm game. We integrated the trained model into a custom VR Boxing Rhythm game to generate rhythmic notes based on predicted gestures. A score is awarded when the player performs the correct gesture within the specified temporal window as shown in [Fig. 6](#). This design was developed to simulate an interaction where gesture timing and classification are critical for the user to score a point. The model will predict the gesture class and generate an obstacle based on the predicted note, while the player needs to hit the notes with Hitting tools (boxing glove animation on the left-bottom side of [Fig. 5](#)). Finally, [Fig. 6](#) shows that the score is registered only if the correct gesture is performed within a valid temporal window upon the note's arrival, and successful hits increment the player's score, providing immediate feedback to the player.

We conducted an ablation study comparing standard TimeSformer with CE, TimeSformer with label-smoothed CE, and the combination of label-smoothed CE and Soft-DTW loss to validate the efficacy of the proposed method with public benchmark datasets. This evaluation was performed across five global action datasets: JHMDB [27], HMDB51 [25], Kinetics-400 [23], a 5% subset of Kinetics 400/600, and UCF101 [26] using the official training and validation splits provided by the datasets' authors to ensure our results are directly comparable. The first dataset is JHMDB, which is a standard computer vision dataset used for spatio-temporal action recognition. This dataset contains 960 annotated video sequences and 21 human action categories. The JHMDB dataset is a subset of HMDB51. HMDB51 consists of 6766 video clips across 51 action categories, with a minimum of 101 video clips in each class. The Kinetics-400 dataset is a

large-scale human action dataset collected from YouTube video clips, consisting of approximately 300,000 video clips with an average duration of 10 s and 400 classes. The Kinetics-400/600 5% subset is a smaller version collected from Kinetics-400 and Kinetics-600 datasets, used to enable efficient experimentation under limited computational resources. The UCF101 dataset has 101 classes and contains over 13,000 video clips collected from YouTube. With the smaller size of our dataset, we employed a five-fold cross-validation to measure accuracy. The public datasets were processed using the exact same frame sampling strategy ($F = 8$ frames) and spatial resizing (224×224 pixels) as our custom VR dataset. The fine-tuning hyperparameters remained identical to the configuration detailed in Table 2. The only methodological difference was the evaluation on our VR dataset that utilized 5-Fold Cross-Validation, and all public datasets were evaluated using their official training and validation splits to ensure direct comparability with prior literature. Table 3 shows the top-1 classification accuracy for the public datasets and five-fold cross-validation accuracy for our own VR rhythm game dataset.



Figure 5: VR boxing rhythm game design used to implement VR custom dataset.

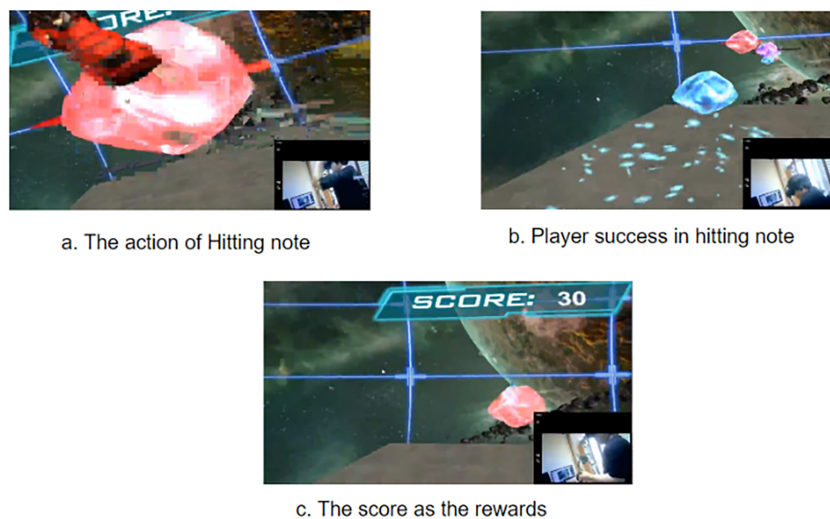


Figure 6: VR boxing rhythm gameplay interface.

Table 3 shows the highest performance improvement in our VR game dataset, with our method outperforming the original cross-entropy by 3.07%. The original TimeSformer with CE architecture reported a top-1 accuracy of 77.9% [10] on the Kinetics-400 benchmark. Because this large dataset is distributed via YouTube video IDs, a subset of the original video corpus is no longer accessible due to deletions or bitstream corruption during download. This has led to a slightly reduced training distribution and a reduction in the number of effective training samples compared to the previous study. In this case, our approach improves classification accuracy by 0.63%. In the smaller datasets, we use 5% of the combination of the Kinetics400 and Kinetics600 datasets with 1.31% improvement on the result. Other datasets, such as HMDB51, show only modest improvements, but the smaller JHMDB dataset exhibits a larger improvement of 1.54 percentage points.

The integration of label smoothing in CE provides immediate and consistent accuracy improvements over the standard CE. For instance, classification accuracy on the JHMDB dataset improved from 87.69% to 88.43%, and UCF101 saw a substantial gain of 1.58%. These results suggest that label smoothing helps mitigate network overconfidence and promotes more robust feature representations. Adding Soft-DTW as an auxiliary loss further improves performance, supporting its effectiveness as a temporal regularizer. Larger gains were observed on datasets where temporal alignment plays a more critical role—for instance, reaching 91.11% accuracy on the 5% K-400/600 subset. While performance on highly saturated benchmarks like HMDB51 remains stable when adding Soft-DTW, the auxiliary loss preserves overall accuracy while encouraging temporally consistent representations.

We observed a substantial improvement on the UCF101 dataset with +1.63% top-1 accuracy improvement. The proposed method achieved a top-1 accuracy of 98.33% on UCF101, which was the highest accuracy among the evaluated datasets. Motivated by this result, we further compared our method with recent Transformer-based approaches evaluated on UCF101, as summarized in **Table 5**.

Table 5: Comparison of the proposed TimeSformer with SmoothedCE+SoftDTW loss against recent Transformer baselines on the UCF101 benchmark.

Year	Method	Backbone	Frames	Pretrain	UCF top-1
2021	TimeSformer Base [10]	ViT-B (div.space-time)	8	K-400	96.7%
2022	VideoMAE [21]	ViT-B (masked autoencoder)	16	K-400	96.1%
2023	MGM [28]	ViT-B (motion guided MAE)	16	K-400	94.2%
2024	LS-ViT [29]	ViT-B (long-short temporal diff.)	16	K-400	97.1%
2026	2STT [30]	MViTV2-S	16	K-400	93.54%
2026	Ours	ViT-B (div.space-time)	8	K-400	98.33%

To ensure a fair and comparative analysis, all models listed in **Table 5** were pre-trained on the Kinetics-400 dataset and used transformer-based backbones for the UCF101 dataset. Recent state-of-the-art methods typically rely on computationally heavy architectural modifications, such as masked autoencoders (VideoMAE) or motion-guided masking (MGM), and require 16 input frames to effectively model temporal dynamics. In contrast, our method retains the standard TimeSformer (ViT-B) architecture with Divided Space-Time Attention and requires only 8 input frames. The results show that our method achieves the highest top-1 accuracy while using the smallest number of input frames. Our proposed method introduces no additional inference-time computational cost, because the model is identical to the TimeSformer-Base architecture, matching the TimeSformer-Base in FLOPs, parameter count, and latency. These results suggest

that standard CE training is more susceptible to performance degradation in low-data regimes, and Soft-DTW loss enables the model to understand temporal dynamics as an inductive bias. Label smoothing also helps to prevent early memorization and reduce overfitting in low-data settings, resulting in robust performance when large-scale training data is unavailable. These results suggest that our proposed loss function is particularly effective when actions exhibit high intra-class variation in execution speed, with the Soft-DTW loss successfully aligning these sequences.

5 Discussion

We acknowledge a limitation in our current use of a fixed hyperparameter ($\lambda = 0.1$) to balance the objective functions. As observed in our experiments, the degree of temporal variation differs across datasets, with highly dynamic gestures exhibiting greater variation in execution speed. In future work, we plan to investigate an adaptive λ tuning mechanism. By dynamically scaling the Soft-DTW contribution based on batch-wise temporal variance, the training objective could adaptively adjust the degree of temporal regularization, potentially improving feature alignment across diverse action scenarios.

Although the proposed joint loss framework achieves high accuracy, the current inference latency of 2142 ms/clip limits immediate real-time deployment. Future work will transition from theoretical loss optimization to practical system implementation, focusing on model compression and predictive buffering to meet the stringent throughput demands of interactive VR environments.

While the primary focus of this study is algorithmic accuracy and loss function optimization of the gesture recognition model, the practical deployment in interactive systems necessitates broader system-level considerations. The end-to-end pipeline required to deploy this framework encompasses continuous video capture, network transmission, and data storage, which inherently introduces vulnerabilities regarding data integrity and user privacy. Integrating encryption protocols and privacy-preserving machine learning techniques at the application layer will be a critical prerequisite for securely deploying these models without compromising data integrity.

6 Conclusion

This study addressed the challenges of human action recognition in data-scarce environments. Although TimeSformer performs well when trained on large-scale datasets, its performance degrades when applied to limited datasets. To address this limitation, we propose a joint-objective loss function that integrates label-smoothed CE and Soft-DTW regularization for the TimeSformer backbone. We introduced a custom VR rhythm-game dataset to evaluate the proposed method in a practical, low-data setting. Experimental results show an improvement of 3.07 percentage points using five-fold cross-validation in this VR custom dataset. Furthermore, we achieved improved top-1 accuracy on public datasets such as JHMDB, HMDB51, Kinetics-400, and UCF101. The proposed method achieved its highest top-1 accuracy of 98.33% on UCF101 using only eight input frames, which indicates that our loss function improves the performance of TimeSformer with low computational overhead.

Acknowledgement: Not applicable.

Funding Statement: This research was supported by Chung-Ang University Young Scientist Scholarship (CAYSS) in 2020.

Author Contributions: The authors confirm their contribution to the paper as follows: Conceptualization, Riski Nur Azizah and Bong-Soo Sohn; methodology, Riski Nur Azizah; software, Riski Nur Azizah; validation, Riski Nur Azizah; formal analysis, Riski Nur Azizah and Bong-Soo Sohn; investigation, Riski Nur Azizah and Bong-Soo Sohn; resources,

Riski Nur Azizah; data curation, Riski Nur Azizah; writing—original draft preparation, Riski Nur Azizah; writing—review and editing, Riski Nur Azizah and Bong-Soo Sohn; visualization, Riski Nur Azizah; supervision, Bong-Soo Sohn; project administration, Riski Nur Azizah and Bong-Soo Sohn. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the first author, Riski Nur Azizah (riskinurazizah22@gmail.com), upon reasonable request.

Ethics Approval: The custom gesture dataset utilized in this study was generated exclusively by the author recording her own movements. No external human participants were recruited, recorded, or involved in any capacity. Therefore, institutional ethics committee approval and informed consent were not required.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang H, Kläser A, Schmid C, Liu CL. Dense trajectories and motion boundary descriptors for action recognition. *Int J Comput Vis.* 2013;103(1):60–79. doi:10.1007/s11263-012-0594-8.
2. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE.* 1998;86(11):2278–324. doi:10.1109/5.726791.
3. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Li FF. Large-scale video classification with convolutional neural networks. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*; 2014 Jun 23–28; Columbus, OH, USA. p. 1725–32.
4. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3D convolutional networks. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*; 2015 Dec 7–13; Santiago, Chile. p. 4489–97.
5. Gao S, Wu R, Liu S, Erkan U, Toktas A, Liu J, et al. MSLID-TCN: multi-stage linear-index dilated temporal convolutional network for temporal action segmentation. *Int J Mach Learn Cybern.* 2024;16(1):567–81.
6. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Adv Neural Inf Process Syst.* 2017;30:15. doi:10.65215/mdcm8z23.
7. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16×16 words: transformers for image recognition at scale. In: *Proceedings of ICLR*; 2021 May 3–7; Online.
8. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-End object detection with transformers. In: Vedaldi A, Bischof H, Brox T, Frahm JM, editors. *Computer vision—ECCV 2020*. Berlin/Heidelberg, Germany: Springer; 2020. p. 213–29.
9. Zhao H, Jia J, Koltun V. Exploring self-attention for image recognition. In: *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2020 Jun 14–19; Seattle, WA, USA. p. 10073–82.
10. Bertasius G, Wang H, Torresani L. Is space-time attention all you need for video understanding?. In: *Proceedings of ICML*; 2021 Jun 18–24; Online.
11. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2016 Jun 27–30; Las Vegas, NV, USA. p. 2818–26.
12. Müller R, Kornblith S, Hinton GE. When does label smoothing help? In: *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*; 2019 Dec 8–14; Vancouver, BC, Canada.
13. Muller M. *Dynamic time warping*. Berlin/Heidelberg, Germany: Springer; 2007. p. 69–84.
14. Cuturi M, Blondel M. Soft-DTW: a differentiable loss function for time-series. In: *Proceedings of the 34th International Conference on Machine Learning*; 2017 Aug 6–11; Sydney, Australia. p. 894–903.
15. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.

16. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H. Training data-efficient image transformers & distillation through attention. In: Proceedings of the 38th International Conference on Machine Learning; 2021 Jul 18–24; Virtual. p. 10347–57.
17. Cai X, Xu T, Yi J, Huang J, Rajasekaran S. DTWNet: a dynamic time warping network. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019); 2019 Dec 8–14; Vancouver, BC, Canada.
18. Ko D, Choi J, Ko J, Noh S, On KW, Kim ES, et al. Video-text representation learning via differentiable weak temporal alignment. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 5006–15.
19. Wang Y, Li K, Li Y, He Y, Huang B, Zhao Z, et al. InternVideo: general video foundation models via generative and discriminative learning. arXiv:2212.03191. 2022.
20. Yan Z, Xia S, Yu J, Wu Y, Jiang T, Li S, et al. InternVideo3: agentify foundation models with multimodal contextual reasoning. arXiv:2606.12195. 2026.
21. Tong Z, Song Y, Wang J, Wang L. VideoMAE: masked autoencoders are data-efficient learners for self-supervised video pre-training. Adv Neural Inf Process Syst. 2022;35:10078–93. doi:10.52202/068431-0732.
22. Li K, Li X, Wang Y, He Y, Wang Y, Wang L, et al. Videomamba: state space model for efficient video understanding. In: European Conference on Computer Vision. Berlin/Heidelberg, Germany: Springer; 2024. p. 237–55.
23. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, et al. The kinetics human action video dataset. arXiv:1705.06950. 2017.
24. Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 6299–308.
25. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. HMDB: a large video database for human motion recognition. In: Proceedings of the 2011 International Conference on Computer Vision; 2011 Nov 6–13; Barcelona, Spain. p. 2556–63.
26. Soomro K, Zamir AR, Shah M. UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402. 2012.
27. Jhuang H, Gall J, Zuffi S, Schmid C, Black MJ. Towards understanding action recognition. In: Proceedings of the 2013 IEEE International Conference on Computer Vision; 2013 Dec 1–8; Sydney, Australia. p. 3192–9.
28. Fan D, Wang J, Liao S, Zhu Y, Bhat V, Santos-Villalobos H, et al. Motion-guided masking for spatiotemporal representation learning. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Dec 4–6; Paris, France. p. 5596–606.
29. Chen D, Wu P, Chen M, Wu M, Zhang T, Li C. LS-ViT: vision transformer for action recognition based on long and short-term temporal difference. Front Neurorobot. 2024;18:1457843.
30. Kurpukdee N, Bors AG. Two-stream temporal transformer for video action classification. arXiv:2601.14086. 2026.