OPT-STVIT: Video Recognition through Optimized Spatial-Temporal Video Vision Transformers

Dr. Divya Nimma[1], Arjun Uddagiri[2]

[1] *PhD in Computational Science, University of Southern Mississippi, Data Analyst in UMMC. nm.divya89@gmail.com*
[2] *Chief Executive Officer, Gloom Dev Pvt Ltd, Penamaluru, Vijayawada, Andhra Pradesh, India. arjunuddagiri@gmail.com*
Corresponding Author: Dr. Divya Nimma

KEYWORDS
Vision Transformer, Object Detection, Object Classification, Pyramid Vision Transformer, Adaptive Patch, Intelligent method

ABSTRACT
In this paper, we address the computational chal- lenges associated with video recognition tasks, where video transformers have shown impressive results but come with high computational costs. We introduce Opt-STViT, a token selection framework that dynamically chooses a subset of informative tokens in both temporal and spatial dimensions based on the input video samples. Specifically, we frame token selection as a ranking problem, leveraging a lightweight scorer network to estimate the importance of each token. Only tokens with top scores are retained for downstream processing. In the temporal dimension, we identify and keep the frames most relevant to the action categories, while in the spatial dimension, we pinpoint the most discriminative regions in feature maps without affecting the spatial context used hierarchically in most video transformers. To enable end-to-end training despite the non-differentiable nature of token selection, we employ a perturbed-maximum-based dif- ferentiable Top-K operator. Our extensive experiments, primar- ily conducted on the Kinetics-400 and something-something-V2 datasets using the recently introduced MViT video transformer backbone, demonstrate that our framework achieves similar results while requiring 20 percent less computational resources. We also establish the versatility of our approach across different transformer architectures and video datasets.

1. Introduction

Research in video recognition, which involves classifying video based content in actions and events, has been acceler- ated by the quick growth of online videos. Applications like content-based retrieval [1, 2] and recommendation systems [3- 6] are supported by this research. The model used for video recognition is spatial-temporal modeling which detects the movement of humans and moving objects for the interaction of each other over time. Vision transformers [7, 8] playing a vast role for the capturing long range dependencies in Natural Language processing (NLP) [9, 10]. But for video recognition these transformers [11-14] are computationally expensive. The issue arises from the fact that as the number of frames in a Identify applicable funding agency here. If none, delete this.

Should be handled sequentially, first focusing on salient frames throughout the whole time horizon, and then delving into those frames to find the most significant geographical region.

We shall overcome this limitation by introducing a novel ViT called Optimized Spatial Temporal Video Vision Transformer (Opt-STViT) plug-and-play ViT that may be used to learn how to video clip increases, the number of input tokens also increases linearly. This, in turn, leads to a quadratic cost in computing self-attention. Consequently, video transformers often require substantial computational resources, making them impractical for deployment in scenarios with limited computing capabil- ities. The previous studies of CNNs [15-24] for the video based recognition have many limitations and cannot performed well for the detection of video frameworks and architectures. The transformers take a picture and convert it into small patches that tokenize a sequence image processing. Self- attention layers are used to relate the modeled patches. Given that transformers have been demonstrated to be resistant to patch drop behaviors [25], a few fairly recent approaches have attempted to lower the computational cost of transformers in the picture domain by training them to drop redundant tokens [26, 27]. The transformer-based models were created for video categorization that are motivated by ViT and the fact that attention-based architectures are an obvious choice for representing long-range contextual interactions in video. The most effective models at the moment are built using deep 3D convolutional architectures [20, 21, 28], a logical progression from image classification CNNs [29, 30]. In order to better capture long-range relationships, these models have recently been improved by adding self-attention to their later layers [31-34] The idea of generalizing the transformers from image to video is non-trivial. For tokenization layer 3D cubes of the token are used for the video transformers which set all the tokens in a

sequence in the form of 3D vectors. Because of this, learning what to maintain in the sequence alone through sampling-based methods [26, 27]. always results in a set of spatially and temporally discontinuous tokens, obliterating the structural information in movies. The latest design of video transformers, which processes 3D tokens in a hierarchical fashion while maintaining both spatial and temporal context, likewise runs counter to this [12, 13]. We propose that the selection of spatial-temporal tokens in video transformers manage computational resources both physically and temporally in video transformers as shown in figure below.

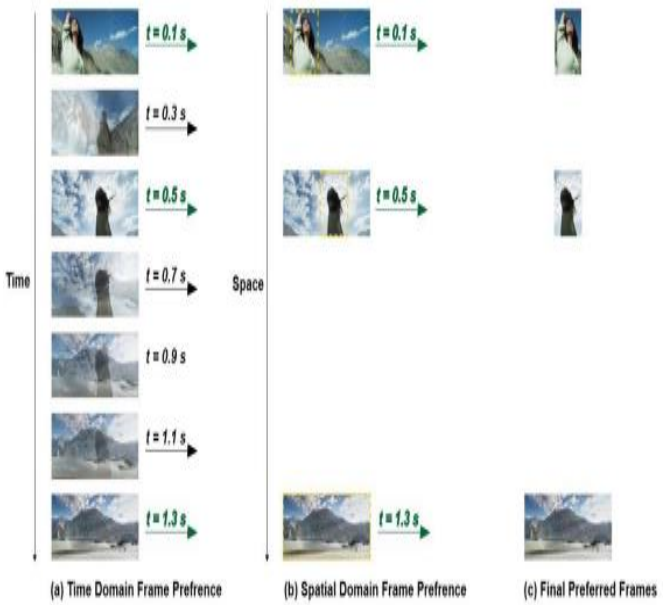

Figure 1: Fig a presents video frames at different temporal intervals for input. Of all the 07 frames only 03 frames are meaningful as far as recognition is concerned. Hence only 03 temporal frames are shortlisted in the first stage. Fig b performs spatial filtration over selected frames and keeps ROI based on anchors proposed. While fig c depicts the final recognized scene.

To employ the fewest number of tokens possible, Opt-STViT comprises of two networks: a temporal token selection network and a spatial token selection network. Each selection network is a multi-layer perceptron (MLP) that may be coupled to any point on a transformer model and forecasts the relevance score of each syllable. We select a few tokens with better values for further processing based on these scores. More specifically, Opt-STViT first chooses a few significant frames throughout the whole time horizon given a series of input tokens. We next divide the token sequences into anchors with regular forms for each frame and choose the single anchor that contributes the most to video recog- nition. It is important to note that choosing tokens with the highest scores is not differentiable, which creates difficulties for training. We use a recently suggested differentiable Top-K selection algorithm [35] to make selection end-to-end trainable using the perturbed maximum technique in order to address this issue. Additionally, this enables us to specify the number of tokens that are used. By presenting frame preference as a computational cost challenge, we would develop a frame preference technique to increase the inference effectiveness of video transformers. Each frame is designated with a significant score using a lightweight scorer network, and only those with top scores ratings are preserved for computation. We choose the most informative frames in the temporal dimension, while in the spatial dimension, the primarily employ the cutting- edge video transformer MViT [12] as our basic model and assess the performance of Opt-STViT using HPC. We shall be performing our experiments on Kinetics-400 [36] which is made up of 240k training films and 20k validation videos that fall into 400 different activity categories. After performing experiments on datasets by both methods, the results will demonstrate their accuracy by using Opt-STViT model can effectively improve the efficiency at the cost of only a slight loss of accuracy. Specifically, using MViT [12] as testing method, Opt-STViT reduces the computational cost by more than 33 Percent with a drop in accuracy of 0.7 Percent on Kinetics-400.

2. Literature Review

Current plant disease classification systems primarily rely on traditional machine learning models and manual feature extraction techniques. These methods often face limitations in accuracy due to the challenges of data diversity and class imbalance. Many systems use basic convolutional neural networks (CNNs) which, while

effective, struggle to generalize across different plant species and disease variations. Additionally, existing systems often require extensive labeled datasets, which are not always available. This reliance on limited data and manual intervention results in slower and less precise disease detection, highlighting the need for more advanced and automated solutions.

2.1 Vision Transformers

Transformers [10] are being used as the backbone archi- tectures in place of convolutional neural networks (CNNs) in the computer vision community as a result of the excep- tional successes of Transformer models in the field of natural language processing (NLP) [7, 37-39]. Vision transformers have emerged as a result of this change, and they have demonstrated superior performance in a range of image-related tasks, including image classification [37], object detection [40], and semantic segmentation [41]. The use of a large amount of pre-training data in the image domain allowed for the achievement of these amazing results. have inspired the Computer vision community after the success of the NLPs and CNNs through the backbone architecture used in transformers. According to a number of methods, researchers have recently been looking at the use of vision transformers in the field of movies [11-13, 42-45]. For instance, TimeSformer [11] which concatenates patches from several frames along the temporal dimension, introduces modifications to the basic transformer design specifically for movies. To capture multiscale visual information necessary for video recognition, MViT [12] uses a hierarchical transformer design that gradually increases the channel size while decreasing the spatial resolution. A further method, VideoSwin [13], uses the natural bias towards locality in video data to apply the idea of window-based local self- attention [46] to video modelling.

2.2 Efficient Vision Transformers

Efficient Transformers are a class of neural network archi- tectures designed to optimize the computational efficiency of traditional Transformers [47]. They are particularly relevant to spatial-temporal video recognition, where they aim to reduce computational complexity and improve the efficiency of pro- cessing video data. By streamlining the architecture, efficient Transformers help address the computational challenges posed by video recognition tasks, including real-time processing, multi-modal data fusion, and scalability to handle longer video sequences and higher resolutions. In essence, they enable more effective and efficient analysis of video data, making them a crucial component in advancing the field of spatial-temporal video recognition.

2.3 Comparative Analysis

Video Vision Transformers (ViTs) represent a critical ad- vancement in spatial-temporal video recognition. Tailored for processing video data, ViTs excel at capturing temporal dy- namics, efficiently extracting spatial and temporal features, and facilitating multi-modal fusion, all crucial elements for recog- nizing complex patterns and actions in videos. Their scalability to longer video sequences, ability to leverage pre-training for transfer learning, and potential for real-time processing make them a valuable asset in various video recognition applica- tions[48]. With competitive performance and the capacity to model both spatial and temporal information effectively, ViTs have emerged as a pivotal tool in addressing the challengesof video understanding and recognition in computer vision research and applications.

2.4 Efficient Video Transformer

Convolutional neural networks (CNNs) have considerably increased video recognition since their adoption. Using 3D- CNN, such as C3D [49], I3D [28], and ResNet3D [50], that model both temporal and spatial information simultaneously, is one frequent strategy. In a separate strategy, frame-level features are extracted and aggregated at various time points using techniques such temporal averaging [51], LSTM net- works [52], or channel shifting [53]. Despite the success of these methods, their applicability may be constrained by the computational expense of CNNs, particularly 3D-CNNs. Recent studies have concentrated on developing lightweight architectures [22, 24, 53-56] or carrying out dynamic comput- ing on a per-video basis [16, 57-61] in order to increase the effectiveness of video recognition. With a focus on resolving spatial redundancy, our method is in line with the latter idea of decreasing intrinsic redundancy in video data. Convolutional neural networks (CNNs) and transformer models have been extensively studied in a number of recent studies on video recognition [20, 50, 62, 63] and [11-13]. However, because these models are frequently computationally intensive, effec- tive video recognition techniques have been developed [24, 25, 58, 64-66] to reduce the amount of time needed for inference. By compressing 3D CNNs, certain studies [20, 21, 24, 67, 68] have concentrated on developing compact models for video recognition. While using these methods results in significant memory

savings, the computational complexity is not reduced because each temporal clip of the incoming video must still be processed. The most pertinent temporal clips to feed into backbone models, on the other hand, are suggested by a few recent techniques [16, 23, 51, 65] allowing for resource-effective video recognition. These methods, however, largely focus on speeding up CNN-based video recognition algorithms. Our study is the first to examine effective recog- nition for video transformers, as far as we are aware. It is crucial to remember that the Optimized Spatial-Temporal Video Vision Transformers (Opt-STViT) method we present is distinct from and enhances current initiatives to build effective vision transformers [69, 70].

2.5 Temporal Spatial Video Vision Transformer

In recent years, there has been a noticeable trend in the field of object detection, particularly in the context of video object detection. This trend involves the use of context frames to improve the performance of object detection algorithms. However, existing methods in this domain typically aggregate features in a single step, which has limitations. One key drawback is the lack of spatial information from neighbor- ing frames, leading to insufficient feature aggregation. To address these issues, a novel approach has been proposed. This approach takes a progressive approach to integrate both temporal and spatial information, enhancing the overall object detection process. Temporal information is introduced through a specialized model called the Temporal Feature Aggregation Model (TFAM). TFAM incorporates an attention mechanism that focuses on the relationships between the context frames and the target frame, which is the frame where the object detection is performed. Additionally, a Spatial Transition Awareness Model (STAM) is employed in this approach. STAM's role is to capture and convey information about the spatial transitions between each context frame and the target frame. This is crucial for understanding how objects move or change positions between frames. This entire approach builds upon a transformer-based object detector known as DETR. Importantly, it maintains an end-to-end fashion, which means that it avoids the need for extensive post-processing steps. As a result of these innovations, this proposed method, referred to as PTSEFormer, achieves an impressive mean Average Precision (mAP) of 88.1 percent on the ImageNet VID dataset, demonstrating its effectiveness in video object detection. In summary, recent advancements in object detection, particularly in video detection, involve using context frames to improve performance. The proposed PTSEFormer introduces temporal information through TFAM and spatial information through STAM, working in conjunction with the DETR detector in an end-to-end manner, ultimately leading to substantial improve- ments in mAP on the ImageNet VID dataset.

3. Proposed Methodology

An overview of video transformers is provided in the first paragraph of this section (Sec. 3.1). Then, we discuss our token selection module, which adaptively selects a small number of significant tokens, and introduce a crucial method, the perturbed maximum method.in Section 3.2, "Model End-to- End Optimization," Finally, we go into detail regarding Create instances of the token selection modules in the temporal and spatial dimensions. (Sec. 3.3). Figure 2 and 3 displays the overarching framework.

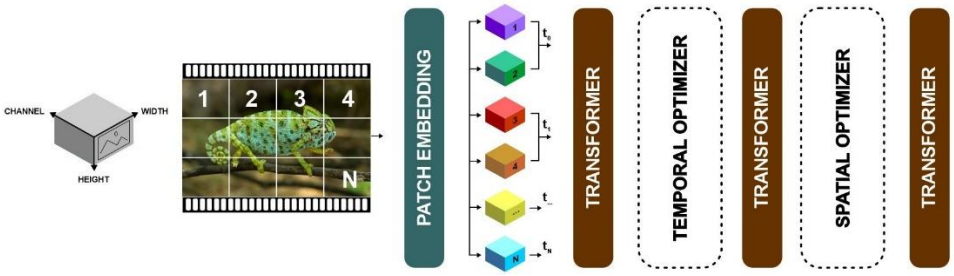

Fig 1: Overview of the Optimized Temporal Spatial Transformer

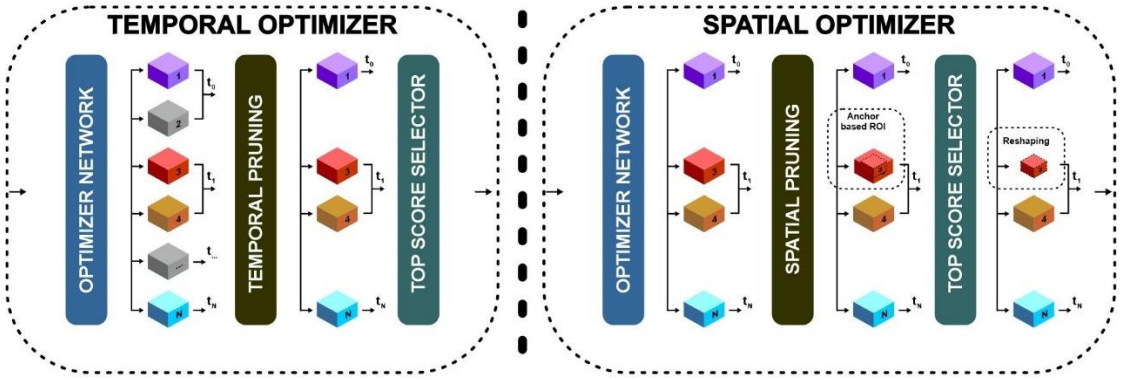

Fig 2: Expended view of the Temporal and Spatial Optimizers Respectively

3.1 Video Transformer

Let we have an input video of V∈R^(T×H×W×3) with size of H x W and T RGB frames video transformers typically employ one of two primary methods to convert these video frames into a sequence of patch embedding. The first method involves dividing the 2D patches within each frame independently using 2D convolutions and then concatenating all these patches along the time dimension. The second approach is to directly extract 3D tubes from the input videos and apply 3D convolutions to transform them into 3D embedding. In both of these approaches, the quantity of tokens generated changes directly to the temporal duration and spatial dimensions of the input video. We represent the resulting spatial-temporal patch embedding as x∈R^(H×W×d) where 'H' and 'W' are the lengths of the token sequence in the time and spatial dimensions, respectively, and 'd' denotes the embedding dimension. To introduce location information into these embedding, positional encodings are incorporated. To capture both the visual appearance and motion cues within videos, the patch embedding denoted as x are input into a series of transformer blocks. These transformer blocks perform spatial and temporal self-attention calculation as Let we have an input video of V∈

$\mathbb{R}^{\boldsymbol{T\times H\times W\times 3}}$ with size of H x W and T RGB frames video transformers typically employ one of two primary methods to convert these video frames into a sequence of patch embedding. The first method involves dividing the 2D patches within each frame independently using 2D convolutions and then concatenating all these patches along the time dimension. The second approach is to directly extract 3D tubes from the input videos and apply 3D convolutions to transform them into 3D embedding. In both of these approaches, the quantity of tokens generated changes directly to the temporal duration and spatial dimensions of the input video. We represent the resulting spatial-temporal patch embedding as $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{H\times W\times d}}$ where 'H' and 'W' are the lengths of the token sequence in the time and spatial dimensions, respectively, and 'd' denotes the embedding dimension. To introduce location information into these embedding, positional encodings are incorporated. To capture both the visual appearance and motion cues within videos, the patch embedding denoted as x are input into a series of transformer blocks. These transformer blocks perform spatial and temporal self-attention calculation as

$$\boldsymbol{Attention}(\boldsymbol{W_q}, \boldsymbol{W_k}, \boldsymbol{W_v}) = \boldsymbol{softmax}\left(\frac{\boldsymbol{W_q W_k^T}}{\sqrt{\boldsymbol{C}}}\right)\boldsymbol{W_v} \quad \textbf{(1)}$$

Here, $\boldsymbol{W_q}, \boldsymbol{W_k}, \boldsymbol{W_v}$represent the query, key, and value embedding derived from x, respectively. The softmax function is applied for normalization purposes, ensuring that the attention scores are appropriately scaled.

3.2 Algorithm for Video Vision Transformer

Algorithm-I for *Opt-STViT*: $\boldsymbol{P} \xleftarrow[\boldsymbol{function}]{} VideoVisionTransformer(\boldsymbol{V}, \boldsymbol{I}, \boldsymbol{z}, \boldsymbol{x}|\boldsymbol{\theta})$

Input: $\boldsymbol{V} \in \mathbb{R}^{T\times H\times W\times 3}, \boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}, \boldsymbol{z_{i,j}} \in \mathbb{R}^{H^2\times 3}$**;** $\boldsymbol{V}, \boldsymbol{I}, \boldsymbol{z}, \boldsymbol{x}$ are RGB input video, 3D frames, $(i,j)th$ patch and sequence of token IDs.

Output: $\boldsymbol{P} \in (\boldsymbol{0}, \boldsymbol{1})$**,** $\boldsymbol{P}$ is Binary Cross Entry Loss Function based conditional probability such that video object exist or not.

Hyperparameters: $l_{max}, L, H, d_E, d_{MLP}, d_f \in \mathbb{N}$

Parameters: $\boldsymbol{\theta}$ represents all the parameters as follows:

| $\boldsymbol{W_E} \in \mathbb{R}^{d_E \times N_V}$, token embedding matrix

For $l \in [L]$:

| $\boldsymbol{W^O} \in \mathbb{R}^{hd_v \times \boldsymbol{d_{model}}}$, multi-head attention

| $\boldsymbol{W_i^Q} \in \mathbb{R}^{\boldsymbol{d_{model}} \times d_Q}$, Query weight

| $\boldsymbol{W_i^K} \in \mathbb{R}^{\boldsymbol{d_{model}} \times d_K}$, Key weight

| $\boldsymbol{W_i^V} \in \mathbb{R}^{\boldsymbol{d_{model}} \times d_V}$, Value weight

| $\boldsymbol{\gamma^n}, \boldsymbol{\beta^m} \in \mathbb{R}^{d_E}$, sets of norm-layer

| $\boldsymbol{W_{MLP_k}^l} \in \mathbb{R}^{d_{MLP} \times \boldsymbol{d_E}}$, weight matrix for MLP layer

| $\boldsymbol{b_{MLP}} \in \mathbb{R}^{\boldsymbol{d_k}}$, bias matrix for MLP layer

| $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{d_E}}, \boldsymbol{W_u} \in \mathbb{R}^{N_V \times \boldsymbol{d_E}}$ final norm-layer & unembedding weight

3.3 Pseudo Code for Opt-STViT

1. ## Extraction of 3D Frames ***I*** RGB from input Video $\boldsymbol{V} \in \mathbb{R}^{T \times H \times W \times 3}$

2. ***for*** input video $\boldsymbol{V} \in \mathbb{R}^{T \times H \times W \times 3}$ ***do***

3. ***for*** kth second frames $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$ ***do***

4. $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3} \xleftarrow[\boldsymbol{function}]{} RGBFrameExtract(\boldsymbol{V} \in \mathbb{R}^{T \times H \times W \times 3} | \boldsymbol{\theta})$

5. ***end***

6. ***end***

7. ## Feed into ViTs

8. ***for*** 3D RGB frames $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$ ***do***

9. ***for*** $X_i \xleftarrow[\boldsymbol{function}]{} 3DConv(\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3} | \boldsymbol{\theta})$ ***do***

10. $FeatureMap\ \boldsymbol{X_{i+1}} \xleftarrow[\boldsymbol{function}]{} 3DConv(\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3} | \boldsymbol{X_i})$

11. ***for*** $FeatureMap\ \boldsymbol{X_{i+1}}$ ***do***

12. $\boldsymbol{E} \xleftarrow[\boldsymbol{function}]{} 3DPatchEmbedding(\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3} | \boldsymbol{X_{i+1}}, W_E)$

13. $\widetilde{\boldsymbol{X}}_{\boldsymbol{ij}} \boldsymbol{W_{ij}^O} \xleftarrow[\boldsymbol{function}]{} Attn(\boldsymbol{E}, \boldsymbol{X_{ij}} | \boldsymbol{W_{ij}^Q}, \boldsymbol{X_{ij}} \boldsymbol{W_{ij}^K}, \boldsymbol{X_{ij}} \boldsymbol{W_{ij}^V})$

14. $\widetilde{\boldsymbol{X}}_{\boldsymbol{i+1}} \xleftarrow[\boldsymbol{function}]{} Concat(\widetilde{\boldsymbol{X}}_{\boldsymbol{ij}} | \boldsymbol{W_{ij}^O})$

15. ***end***

16. ***for*** $\boldsymbol{X_{ij}}, \widetilde{\boldsymbol{X}}_{\boldsymbol{i(j-1)}}$ ***do***

17. $\boldsymbol{X'_{ij}} = \boldsymbol{X_{ij}} + \widetilde{\boldsymbol{X}}_{\boldsymbol{i(j-1)}}$ *while* $\boldsymbol{1 < j \leq h}$

18. ***end***

19. ***end***

20. ***end***

21. ***for*** $\boldsymbol{X'_{ij}}$ ***do***

22. $\boldsymbol{X_{ij}} \xleftarrow[\boldsymbol{function}]{} \boldsymbol{ReLu}(\boldsymbol{W_f X'_{ij}})$

23. ***end***

24. return $P(0,1) = SoftMax(W_u X_{ij}$

3.4 Dynamic Token Selection

As shown in Equation 1, the computational complexity of a video transformer increases quadratic ally with the number of tokens used in the self-attention blocks. Given the inherent spatial and temporal redundancies present in videos, a natural approach to mitigate this computational burden is to reduce the number of tokens. However, the challenge lies in determining which tokens should be retained and which ones can be discarded, and this is a non-trivial task. Reducing the number of tokens while preserving critical information is essential for efficient video processing. This involves striking a balance between computational efficiency and maintaining the representational power needed to address the target task effectively. Achieving this balance often requires innovative token selection methods that take into account the specific characteristics of the input data and the objectives of the task at hand. It is a decision closely tied to both the specific input sample and the target task at hand. Drawing inspiration from recent research on patch selection for high-resolution image recognition, we frame token selection as a ranking problem. Here's a detailed explanation of this approach.

3.4.1 Estimating Importance Scores

We begin by estimating importance scores for the input tokens. To accomplish this, we employ a lightweight scorer network. This network evaluates each token and assigns an importance score based on its relevance to the task.

3.4.2 Selecting Top Scorers

After the importance scores are calculated, we proceed to select the top scoring tokens. These tokens are considered the most informative and relevant for the downstream processing tasks. The described two-step process is used for both spatial and temporal token selection. In spatial token selection, it helps identify significant spatial patches within frames, while in temporal token selection, it assists in determining the most relevant frames over time. This approach is valuable for reducing computational complexity while preserving critical information for video analysis and understanding.

3.4.3 Scorer network.

The objective of the scorer network is to generate importance scores for each token in an input sequence represented as q, where q is a matrix with dimensions (L, C). Here, L signifies the length of the sequence when flattened, and ‘d’ represents the embedding dimension. To accomplish this, we employ a conventional two-layer fully connected (FC) neural network to compute these scores.

More specifically, we start by mapping the input tokens to a local representation denoted as f_l through a linear projection:

$$\boldsymbol{f}^{\boldsymbol{local}} \xleftarrow[\boldsymbol{function}]{} FullyConv(\boldsymbol{q}|\boldsymbol{\theta})$$

$$\boldsymbol{s} \xleftarrow[\boldsymbol{function}]{} FullyConv(\boldsymbol{f}_i|\boldsymbol{\theta})$$

where θ denotes the network weights and f^local is a vector of importance scores for all tokens, which has a dimension of ‘d’. These scores have been normalized using the min-max normalization technique.

3.5 Selection of Top Highest scorer

After obtaining the importance scores, denoted as "s," from the scorer network, we proceed to select the highest scores and extract the corresponding tokens. This operation is referred to as the "Top scorer" operator, and it returns the indices of the largest entries. In mathematical terms, it is represented as follows:

$$y = Top\ scorer(s) \in N\hat{}K.$$

To train the parameters of the scorer network through an end-to-end training process without the need for additional loss functions, we employ the perturbed maximum method. This allows us to create a differentiable version of the Top scorer operator, ensuring that gradients can be computed throughout the training process.

3.5.1 Forward

A smoothed variant of the Top scorer operation as described in below can be achieved by computing the expectation with respect to random perturbations. This is represented as follows:

Y= expect_argmax(rand| (y,s))

For testing, we apply the Top scorer algorithm for n iterations (typically set to 500 in all our experiments), and then calculate the expectation by averaging over the results of these n independent samples.

3.5.2 Backward

During the training process, we simultaneously train both the backbone models and the token selection networks using the cross-entropy loss in an end-to-end manner. However, during inference, we aim to further enhance efficiency by employing the hard Top scoring operation, where only a single Top scorer operation is performed (instead of n perturbed repetitions), and token selection is accomplished through tensor slicing.

However, it's important to note that using hard Top-scoring during inference can lead to a difference between the training and testing phases. To address this issue, we introduce a linear decay of the hyper-parameter d_E towards zero during the training process. It's worth emphasizing that when d_E equals zero, the differentiable Top scoring operation becomes equivalent to hard Top scorer, and the gradients flowing into the scorer network become negligible or vanish.

3.6 Temporal and Spatial Token Selection

The unique characteristics of appearance and motion information in videos, we adopt a two-step approach for token selection. First, we focus on selecting salient frames, and then we further refine our attention to identify the most crucial spatial regions within those frames. This approach involves separating the processing of spatial and temporal information, attending to significant frames initially and subsequently delving into these frames to pinpoint the most important spatial regions.

3.6.1 Temporal selection

Given the input tokens *x* with dimensions *W×H×d,* the objective of temporal selection is to choose Top scoring matrix out of the W frames and disregard the remaining ones. Here's how this process is carried out: Initially, we perform average pooling on *x* along the spatial dimension. This results in a sequence of frame-based tokens denoted as *xt* with dimensions *W×d.* Next, we pass *xt* through the scorer network and the Top Scorer operator to generate an indicator matrix *Yt.* This matrix identifies the frames with the top *K* highest scores and has dimensions *H×Top* Scorer. Afterward, we reshape the input x back to its original form, so it becomes *x* with dimensions *W×(H×d).* Finally, we extract the selected top scoring frames from *x* using the indicator matrix *Yt* by performing a matrix multiplication*: z = Yt^T x,* resulting in a matrix *z* with dimensions *Top scoring matrix×(H×d).* The selected tokens are then reshaped to form ` with dimensions *Top scoring matrix ×H×d* for further processing in downstream tasks.

3.6.2 Spatial selection

Unlike temporal selection, spatial selection is conducted independently for each frame, with the goal of selecting best scoring token out of N tokens for each frame. To clarify, we first input the tokens of a given frame, denoted as x_m and residing in an H x d matrix, into a scorer network to generate importance scores s_m, both of which vary across frames but are described here without frame subscripts for simplicity. To identify the top Scoring spatial tokens, one might initially consider directly applying the Top Scoring operator to the token-based scores s. However, this approach disrupts the spatial arrangement of input tokens, which is particularly problematic for spatial selection in video transformers for two key reasons.

First, modern video transformers, employ a hierarchical architecture that progressively reduces spatial resolutions through multiple stages. Abruptly removing tokens disrupts the spatial structure, which is detrimental to local operations like convolutions and pooling used for spatial down-sampling. Second, the misalignment of spatial tokens along the temporal dimension complicates temporal modeling significantly. Instead of relying on token-based selection, we introduce an innovative anchor-based approach for spatial selection. Here's how it works:

Once we have obtained importance scores, denoted as *'s,'* for each frame, we start by reshaping these scores into a *2D* score map, *'s_s,'* which takes the shape of *s_s ∈ R √N x √N.* Next, we partition this score map into a grid of overlapping anchors, referred to as *'e_s_s,'* which forms a matrix of size *e_s_s ∈ R G x K.* Each anchor covers a region of *K* tokens. The parameter G represents the total number of anchors, and it is calculated as *G = (√N-√Kα+ 1)^2,* where α denotes the stride between anchors. After the anchors are defined, we proceed to aggregate the importance scores within each anchor using average pooling. This aggregation results in anchor-based

scores, denoted as *'s_a,'* which exist in the space *R G*. This transformation effectively transforms the original problem of selecting the top scoring tokens into a simpler task of selecting the token with the highest score within each anchor, essentially becoming a Top-1 selection problem. To identify the highest-scoring token within each anchor,

We once again make use of the Top Scoring operator, setting *K* to *1*. This process generates an indicator matrix and allows us to extract the anchor with the highest score, achieving our goal of spatial selection.

3.7 Algorithm for Temporal and Spatial Token Selection

Algorithm-II for *Opt-STViT*: $\boldsymbol{P} \xleftarrow[\boldsymbol{function}]{} VideoVisionTransformer(\boldsymbol{V}, \boldsymbol{I}, \boldsymbol{z}, \boldsymbol{x}|\boldsymbol{\theta})$

Input: $\boldsymbol{q} \in \mathbb{R}^{L\times C}, \boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}, \boldsymbol{z_{i,j}} \in \mathbb{R}^{H^2\times 3}$**;** $\boldsymbol{V}, \boldsymbol{I}, \boldsymbol{z}, \boldsymbol{x}$ are RGB input video, 3D frames, $(i,j)th$ patch and sequence of token IDs.

Output: $\boldsymbol{P} \in (\boldsymbol{0}, \boldsymbol{1})$**,** $\boldsymbol{P}$ is Binray Cross Entry Loss Function based conditional probability such that video object exist or not.

Hyperparameters: $l_{max}, L, H, d_E, d_{MLP}, d_f \in \mathbb{N}$

Parameters: $\boldsymbol{\theta}$ represents all the parameters as follows:

| $\boldsymbol{W_E} \in \mathbb{R}^{d_E\times N_V}$, token embedding matrix

For $l \in [L]$:

| $\boldsymbol{W^O} \in \mathbb{R}^{hd_v\times \boldsymbol{d_{model}}}$**,** multi-head attention

| $\boldsymbol{W_i^Q} \in \mathbb{R}^{\boldsymbol{d_{model}}\times d_Q}$**,** Query weight

| $\boldsymbol{W_i^K} \in \mathbb{R}^{\boldsymbol{d_{model}}\times d_K}$**,** Key weight

| $\boldsymbol{W_i^V} \in \mathbb{R}^{\boldsymbol{d_{model}}\times d_V}$**,** Value weight

| $\boldsymbol{\gamma^n}, \boldsymbol{\beta^m} \in \mathbb{R}^{d_E}$**,** sets of norm-layer

| $\boldsymbol{W_{MLP_k}^l} \in \mathbb{R}^{d_{MLP}\times \boldsymbol{d_E}}$**,** weight matrix for MLP layer

| $\boldsymbol{b_{MLP}} \in \mathbb{R}^{\boldsymbol{d_k}}$**,** bias matrix for MLP layer

| $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{d_E}}, \boldsymbol{W_u} \in \mathbb{R}^{N_V\times \boldsymbol{d_E}}$ final norm-layer & unembedding weight

25. ## Token Score-Ranking of input token $\boldsymbol{q} \in \mathbb{R}^{L\times C}$

26. ***for*** input video $\boldsymbol{q} \in \mathbb{R}^{L\times C}$ ***do***

27. ***for*** $\boldsymbol{f^{local}} \xleftarrow[\boldsymbol{function}]{} FullyConv(\boldsymbol{q}|\boldsymbol{\theta}_{local})$ ***do***

28. ***for*** $\boldsymbol{f^{Global}} \xleftarrow[\boldsymbol{function}]{} Avg(\boldsymbol{f_i^{local}})$ ***do***

29. ***for*** $\boldsymbol{f}_i \xleftarrow[\boldsymbol{function}]{} Concat(\boldsymbol{f^{local}}, \boldsymbol{f^{Golbal}})$ ***do***

30. $\boldsymbol{s} \xleftarrow[\boldsymbol{function}]{} FullyConv(\boldsymbol{f}_i|\boldsymbol{\theta})$

31. ***end***

32. ***end***

33. ***end***

34. ***end***

35. ## Top-Scored token selection

36. ***for*** Top-Scored token $\boldsymbol{s}$ ***do***

37. ***for*** $\boldsymbol{y} \xleftarrow[\boldsymbol{function}]{} TopScore(\boldsymbol{s})$ ***do***

38. $\boldsymbol{z} \xleftarrow[\boldsymbol{function}]{} TempSelect(\boldsymbol{y})$

39. ***end***

40. ***end***

41.

42. ## Top-Scored token selection

43. ***for*** Top-Scored token ***s do***

44. ***for*** $\boldsymbol{y} \xleftarrow[\boldsymbol{function}]{} TopScore(\boldsymbol{s})$ ***do***

45. $\boldsymbol{z} \xleftarrow[\boldsymbol{function}]{} SpatialSelect(\boldsymbol{y})$

46. ***end***

47. ***End***

4. Results and Discussion

In this section, we assess the performance of Opt-STViT through a comprehensive series of experiments carried out on two substantial video recognition datasets. We employ two modern video transformer backbones for this evaluation. Our experimental setup is detailed in Section 4.1, followed by the presentation of key results in Section 4.2. Additionally, we conduct ablation studies in Section 4.3 to get the significance of various components used in our research.

4.1 Experimental Setup

Our primary base model for evaluation is the *MViT-B16*, which represents a state-of-the-art video transformer architecture. We assess the effectiveness of *Opt-STViT* using the Kinetics-400 dataset [36]. However, it's worth noting that our approach is designed to be versatile and applicable to various transformer architectures and datasets. To demonstrate this versatility, we also conduct experiments with the Video Swin Transformer on both the Kinetics-400 dataset and the Something-SomethingV2 (SSV2) dataset [71].

Different *Opt-STViT* variations are represented by the notation *B-T_R^L-S_R^L* where B denotes the backbone network and T and S denote the token selection operations carried out along the temporal and spatial dimensions, respectively. The positions of the token selection modules and the related ratios of the chosen tokens are indicated by the letters L and R. For instance, *MViT-B16-T^0-S^4* specifies using *MViT-B16* as the base model and doing temporal token selection before the 0th self-attention block with a selection ratio of 0.4 and spatial token selection before the 4th block with a ratio of 0.6.

4.2 Implementation

In our experiments, we fine-tuned pre-trained video transformer models with our *Opt-STViT* modules, initializing *Opt-STViT* module parameters randomly and setting σ in Equation 7 to 0.1. The learning rate for backbone layers was 0.01 times that of *Opt-STViT* modules. For *MViT* models on Kinetics-400, we used 16-frame video clips with a temporal stride of 4, spatial size 224x224, AdamW optimizer for 20 epochs (with 3 epochs of linear warmup), initial learning rates of 1e-4 for *Opt-STViT* and 1e-6 for the backbone, and a mini-batch size of 16. A cosine learning rate schedule was applied. For Video Swin Transformer, we used 32-frame clips with a temporal stride of 2, the learning rate for selection networks was 3e-4, and the backbone model had a learning rate 0.01 times smaller, with a batch size of 64. Training protocols varied for Kinetics-400 and Something-Something-V2, with AdamW optimizer and warm-up epochs. During inference, we followed original backbone model testing strategies for equitable performance comparison.

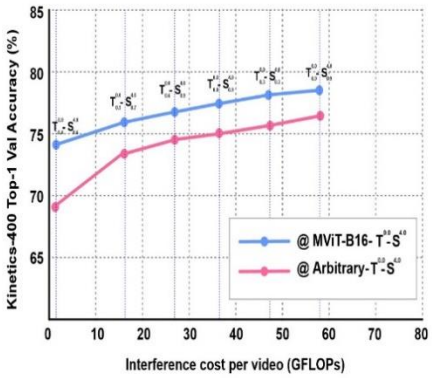

Fig 2: Comparison between Arbitrary and MVit-B16 Token Selection

4.3 The efficiency of Opt-STViT

In our initial comparison, we evaluate Opt-STViT against common token selection baselines, including:

Random (Rand.): This baseline randomly selects K tokens from the input, disregarding their visual content. It represents a simplistic, non-contextual token selection approach.

Gumbel-Softmax (GS): GS employs a Gumbel-softmax trick for token selection. It's important to note that GS cannot be applied to spatial token selection due to the presence of spatial down sampling in recent video transformers.

In Table 1, we provide a summary of results for temporal-only, spatial-only, and joint token selection. In cases where baseline settings are unfeasible, denoted as N/A, we present this information accordingly. Our observations indicate that Opt-STViT consistently achieves the highest accuracy when compared to all baseline methods, despite operating within a similar computational budget. Particularly noteworthy is Opt-STViT significant performance advantage over GS, even though both methods employ the same scorer network design. These results underscore the efficiency of our differentiable Top Token Selector operator for dynamic token selection, which contributes to Opt-STViT superior performance.

Table I: Comparison of Opt-STVit with Random Tokens

Configuration		Random Token	Gumbel Softmax	Opt-ST ViT
Temporal (Only)	T 0.5^0	74.1	74.1	75.7
Spatial (Only)	S 0.9^4	74.9	N/A	76.3
Temporal and Spatial	T 0.8^0-S 0.9^4	75	N/A	77.1
Temporal and Spatial	T 0.9^0-S 0.9^4	76.2	N/A	78.8

In our further comparison, we compared Opt-STViT with the Rand. baseline under various computational budgets, as depicted in Figure 4. The results consistently demonstrate the superior performance of *Opt-STViT*, especially in settings with significant reductions in computation. For example, in the case of $\mathrm{MViT-B16-T}^{0}_{0.8}-\mathrm{S}^{4}_{0.9}$, *Opt-STViT* outperforms Rand. by an impressive margin of 2.7% while using a similar computational budget of 12 GFLOPs. This confirms that our dynamic token selection modules effectively preserve informative tokens. Additionally, it's worth noting that the computational overhead introduced by our token selection module is negligible. In fact, the parameters and FLOPs of the scorer network in *Opt-STViT* constitute only 1.0% and 0.7% of those in the original MViT-B16 backbone, underscoring the efficiency of our approach.

4.4 Results Comparison

In Table 2, we present a comparative analysis between *Opt-STViT* and state-of-the-art video recognition models using the Kinetics-400 dataset. This comparison encompasses a range of models, including both CNN-based and Transformer-based architectures. Our goal is to illustrate the efficacy and competitiveness of our approach alongside these top-performing models. In this paper, we provide the computational cost for inference, which is calculated as the cost for a single view multiplied by the number of views in both spatial and temporal

dimensions. This cost is expressed in Giga-FLOPs (GFLOPs). To facilitate a clear comparison, we categorize the models into two groups and specifically compare *Opt-STViT* with models having similar GFLOPs. It's worth noting that our default *Opt-STViT* settings are based on MViT-T0-S4, which presents a particularly challenging scenario for spatial as well as temporal selection

Table II: Result Comparison of Kinetics-400

Models	Pre-Train	GFLOPS	Top-1
X3D-L	-	0.81	78.2
TimeSformer	IN-21K	0.62	78.6
MViT-B16	-	0.38	79.3
MViT-B16-T 0.8^0-S 0.9^4	-	0.29	80.5
MViT-B16-T 0.8^0-S 0.9^4	-	0.35	79.4

The table 3 presents a comparison of four video interference detection models: TSM, STM, TEA, and MVIT-B16, evaluated on the IN-1K video interference detection dataset. MVIT-B16 emerges as the top performer in terms of Top-1 Val Accuracy (%), achieving an impressive 68.6%, followed by the TEA model at 66.1%, the STM model at 65.3%, and the TSM model at 64.2%. Regarding computational demands measured in GFLOPS (Giga-Floating Point Operations Per Second), MVIT-B16 ranks highest with 2.41 GFLOPS, followed by TEA at 2.24 GFLOPS, STM at 0.65 GFLOPS, and TSM at 0.41 GFLOPS. GFLOPS serves as an indicator of the computational resources needed to execute a model. These results highlight a trade-off between model accuracy and computational cost. MVIT-B16 excels in accuracy but demands the most computational resources. TEA and STM, while slightly less accurate, are more resource-efficient options. On the other hand, the TSM model offers the lowest accuracy among the four but is the most resource-efficient.

Table III: Result Comparison of Something-Something-400

Models	Pre-Train	GFLOPS	Top-1
TSM	K400	0.41	64.2
STM	IN-1K	2.24	65.3
TEA	IN-1K	2.41	66.1
MViT-B16	K400	0.65	68.6

4.5 Result Discussion

4.5.1 Different Token Selection

The flexibility of our Opt-STViT module allows for various token selection configurations to achieve a comparable reduction in computational load. To illustrate, in the case of reducing the computational demands of MViT-B16 by approximately 50%, one can adopt one of the following strategies:

(1) implementing spatial-only or temporal-only token selection at earlier stages with a higher selection ratio (e.g., -S0.6 or -T0.6);

(2) applying spatial-only or temporal-only token selection at later stages with a lower selection ratio (e.g., -S40.3 or -T40.3);

(3) conducting joint token selection (e.g., -T0.6-S40.9),

In this section, we research into a comprehensive analysis of these choices, using MViT-B16 as an example and assessing their performance on the Kinetics-400 dataset. The inference cost is reported for a single view. It's evident from our observations that temporal selection significantly outperforms spatial selection, underscoring the greater importance of mitigating temporal redundancy in videos compared to spatial redundancy. Furthermore, the most favorable outcome is achieved with joint token selection, where temporal token selection occurs early in the model, and spatial token selection occurs in the deeper layers (i.e., -T0.6-S40.9). We also examine the scenario of conducting token selection in a multi-step manner, involving multiple token selections

at various layers of a transformer network, each with a higher selection ratio.

We present visualizations of the outcomes of temporal-only, spatial-only, and joint token selection in Figure 5 and 6, showcasing the frames and regions that are discarded by Opt-STViT. Our observations reveal that Opt-STViT excels in not only accurately identifying the most informative frames within a video clip but also pinpointing the discriminative regions within each frame. By effectively pruning tokens in both the temporal and spatial dimensions, Opt-STViT retains only those tokens crucial for accurate action recognition. These selected tokens are subsequently fed into the subsequent video transformers, resulting in a reduced computational cost while maintaining minimal loss in classification accuracy. This demonstrates the efficiency and effectiveness of Opt-STViT in optimizing token selection for video analysis tasks.

Fig 1: Girl exploring the Beauty of Nature

Fig 2: Chameleon Changing

5. Conclusion and Future Work

In this research paper, we introduced Opt-STViT, a dynamic spatio-temporal token selection framework designed to alleviate both temporal and spatial redundancies within video transformers, enhancing the efficiency of video recognition. We framed token selection as a leading problem, using a lightweight selection network to predict token importance, and preserving only those tokens with top scores for further processing. In the temporal dimension, we selected a subset of frames most relevant to the action category, while in the spatial dimension, we retained the most discriminative regions within each frame to maintain structural information. To facilitate end-to-end training of the backbone model with the token selection module, we incorporated a perturbed-maximum-based differentiable Top Scoring operator. Our extensive experiments across various video recognition benchmarks confirmed that Opt-STViT achieves competitive efficiency-accuracy trade-offs, highlighting its practical utility in video analysis tasks.

References:

[1] Dong, J., et al. Dual encoding for zero-example video retrieval. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[2] Yuan, L., et al. Central similarity quantization for efficient image and video retrieval. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[3] Davidson, J., et al. The YouTube video recommendation system. in Proceedings of the fourth ACM conference on Recommender systems. 2010.

[4] Mei, T., et al., Contextual video recommendation by multimodal relevance and user feedback. ACM Transactions on Information Systems (TOIS), 2011. 29(2): p. 1-24.

[5] Lee, J. and S. Abu-El-Haija. Large-scale content-only video recommendation. in Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017.

[6] T.Yuvanth Sai,Sk. Areef,V. Sri Harsha,Gubbala Satya Sai Deepak,Dr. K. Amarendra,Pachipala Yellamma,"A Study on Cloud and IoT based Accident Detection & Prevention Systems",2023.

[7] Vasamsetti Sri Harsha,Tirumalasetti Yuvanth Sai,Gubbala Satya Sai Deepak,Shaik Areef,Amarendra K.,Pachipala Yellamma,"Image Demorpher Using Machine Learning: Removing Fake Layers and Restoring Original Images",2024.

[8] Arjun Uddagiri,Pragada Eswar,Tummu Vineetha,"Enhancing Mobile security with Automated sim slot ejection system and authentication mechanism",2023

[9] Devlin, J., et al., Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint

arXiv:1810.04805, 2018.

[10] Vaswani, A., et al., Attention is all you need. Advances in neural information processing systems, 2017. 30.

[11] Bertasius, G., H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? in ICML. 2021.

[12] Fan, H., et al. Multiscale vision transformers. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[13] Liu, Z., et al. Video swin transformer. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[14] Wang, R., et al. Bevt: Bert pretraining of video transformers. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[15] T.Pranoom, Y. Vamsi , K Rohit . ," Crop Classification analysis using machine learning",2024.

[16] Korbar, B., D. Tran, and L. Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[17] Wang, Y., et al. Adaptive focus for efficient video recognition. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[18] Wang, Y., et al. Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022. IEEE.

[19] Sun, Z., et al., Human action recognition from various data modalities: A review. IEEE transactions on pattern analysis and machine intelligence, 2022.

[20] Feichtenhofer, C., et al. Slowfast networks for video recognition. in Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[21] Feichtenhofer, C. X3d: Expanding architectures for efficient video recognition. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[22] Tran, D., et al. A closer look at spatiotemporal convolutions for action recognition. in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018.

[23] Wu, Z., et al. Adaframe: Adaptive frame selection for fast video recognition. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[24] Zolfaghari, M., K. Singh, and T. Brox. Eco: Efficient convolutional network for online video understanding. in Proceedings of the European conference on computer vision (ECCV). 2018.

[25] Naseer, M.M., et al., Intriguing properties of vision transformers. Advances in Neural Information Processing Systems, 2021. 34: p. 23296-23308.

[26] Pan, B., et al., IA-RED $^ 2$: Interpretability-Aware Redundancy Reduction for Vision Transformers. Advances in Neural Information Processing Systems, 2021. 34: p. 24898-24911.

[27] Rao, Y., et al., Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems, 2021. 34: p. 13937-13949.

[28] Carreira, J. and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[29] He, K., et al., Deep residual learning for image recognition. CVPR. 2016. arXiv preprint arXiv:1512.03385, 2016.

[30] Sermanet, W.L.Y.J.P., S.R.D.A.D. Erhan, and V.V. Szegedy. Christian and Andrew Rabinovich. Going deeper with convolutions. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[31] Wu, C.-Y., et al. Long-term feature banks for detailed video understanding. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[32] Wang, X., et al. Non-local neural networks. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[33] Girdhar, R., et al. Video action transformer network. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[34] Arnab, A., C. Sun, and C. Schmid. Unified graph structured models for video understanding. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[35] Berthet, Q., et al., Learning with differentiable pertubed optimizers. Advances in neural information processing systems, 2020. 33: p. 9508-9519.

[36] Kay, W., et al., The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[37] Zhang, D., et al. Feature pyramid transformer. in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16. 2020. Springer.

[38] Touvron, H., et al. Training data-efficient image transformers & distillation through attention. in International conference

on machine learning. 2021. PMLR.

[39] Heo, B., et al. Rethinking spatial dimensions of vision transformers. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[40] Zhu, X., et al., Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020.

[41] Zheng, S., et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[42] Arnab, A., et al. Vivit: A video vision transformer. in Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[43] Gabeur, V., et al. Multi-modal transformer for video retrieval. in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. 2020. Springer.

[44] Ryoo, M., et al., Tokenlearner: Adaptive space-time tokenization for videos. Advances in Neural Information Processing Systems, 2021. 34: p. 12786-12797.

[45] Wang, Y., et al. End-to-end video instance segmentation with transformers. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[46] Liu, Z., et al. Swin transformer: Hierarchical vision transformer using shifted windows. in Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[47] Chen, S., et al., Adaptformer: Adapting vision transformers for scalable visual recognition. Advances in Neural Information Processing Systems, 2022. 35: p. 16664-16678.

[48] Huang, Z., et al., Towards training stronger video vision transformers for epic-kitchens-100 action recognition. arXiv preprint arXiv:2106.05058, 2021.

[49] Wang, Y., et al., Implicit semantic data augmentation for deep networks. Advances in Neural Information Processing Systems, 2019. 32.

[50] Hara, K., H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018.

[51] Wu, Z., et al., A dynamic frame selection framework for fast video recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. 44(4): p. 1699-1711.

[52] Donahue, J., et al. Long-term recurrent convolutional networks for visual recognition and description. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[53] Lin, J., C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. in Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[54] Xie, S., et al. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. in Proceedings of the European conference on computer vision (ECCV). 2018.

[55] Pan, B., et al., VA-RED $^ 2$: Video Adaptive Redundancy Reduction. arXiv preprint arXiv:2102.07887, 2021.

[56] Tran, D., et al. Video classification with channel-separated convolutional networks. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[57] Wu, Z., et al., Liteeval: A coarse-to-fine framework for resource efficient video recognition. Advances in neural information processing systems, 2019. 32.

[58] Yeung, S., et al. End-to-end learning of action detection from frame glimpses in videos. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[59] Meng, Y., et al., Adafuse: Adaptive temporal fusion network for efficient action recognition. arXiv preprint arXiv:2102. 05775, 2021.

[60] Zhu, S., et al., A3d: Adaptive 3d networks for video action recognition. arXiv preprint arXiv:2011.12384, 2020.

[61] Li, H., et al. 2d or not 2d? adaptive 3d convolution selection for efficient video recognition. in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[62] Feichtenhofer, C., A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[63] Xu, L., H. Huang, and J. Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[64] Wang, H., et al. Video modeling with correlation networks. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[65] Bhardwaj, S., M. Srinivasan, and M.M. Khapra. Efficient video classification using fewer frames. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[66] Zheng, Y.-D., et al., Dynamic sampling networks for efficient action recognition in videos. IEEE transactions on image processing, 2020. 29: p. 7970-7983.

[67] Wu, C.-Y., et al. Compressed video action recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[68] Kondratyuk, D., et al. Movinets: Mobile video networks for efficient video recognition. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[69] Wang, S., et al., Self-attention with linear complexity. arXiv preprint arXiv:2006.04768, 2020. 8.

[70] Kitaev, N., Ł. Kaiser, and A. Levskaya, Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451, 2020.

[71] Goyal, R., et al. The" something something" video database for learning and evaluating visual common sense. in Proceedings of the IEEE international conference on computer vision. 2017.

[72] Arjun Uddagiri,Pragada Eswar,Tummu Vineetha,"Enhancing Mobile security with Automated sim slot ejection system and authentication mechanism",2023

[73] Divya Nimma, Rajendar Nimma, Arjun Uddagiri," Advanced Image Forensics: Detecting and reconstructing Manipulated Images with Deep Learning.",2024

[74] Nimma, D., Zhou, Z. Correction to: IntelPVT: intelligent patch-based pyramid vision transformers for object detection and classification. Int. J. Mach. Learn. & Cyber.(2023).

[75] Nimma, Divya & Zhou, Zhaoxian. (2023). IntelPVT: intelligent patch-based pyramid vision transformers for object detection and classification. International Journal of Machine Learning and Cybernetics. 1-12. 10.1007/s13042-023-01996-2.

[76] Nimma, D., Zhou, Z. Correction to IntelPVT: intelligent patch-based pyramid vision transformers for object detection and classification. Int. J. Mach. Learn. & Cyber. 15, 3057 (2024). https://doi.org/10.1007/s13042-023-02052-9

[77] Divya Nimma, "Advanced Image Forensics: Detecting and reconstructing Manipulated Images with Deep Learning. ", Int J Intell Syst Appl Eng, vol. 12, no. 4, pp. 283 –, Jun. 2024.

[78] Mithun DSouza, Divya Nimma, Kiran Sree Pokkuluri, Janjhyam Venkata Naga Ramesh, Suresh Babu Kondaveeti and Lavanya Kongala, "Multiclass Osteoporosis Detection: Enhancing Accuracy with Woodpecker-Optimized CNN-XGBoost" International Journal of Advanced Computer Science and Applications(IJACSA), 15(8), 2024. http://dx.doi.org/10.14569/IJACSA.2024.0150889

[79] Wael Ahmad AlZoubi, Girish Bhagwant Desale, Sweety Bakyarani E, Uma Kumari C R, Divya Nimma, K Swetha and B Kiran Bala, "Attention-Based Deep Learning Approach for Pedestrian Detection in Self-Driving Cars" International Journal of Advanced Computer Science and Applications(IJACSA), 15(8), 2024. http://dx.doi.org/10.14569/IJACSA.2024.0150891

[80] Divya Nimma, "Deep Learning Techniques for Image Recognition and Classification", IJRITCC, vol. 12, no. 2, pp. 467–474, Apr. 2024.

[81] Divya Nimma, "Image Processing in Augmented Reality (AR) and Virtual Reality (VR)", IJRITCC, vol. 12, no. 2, pp. 475–482, Apr. 2024.

[82] Divya Nimma and Arjun Uddagiri, "Advancements in Deep Learning Architectures for Image Recognition and Semantic Segmentation" International Journal of Advanced Computer Science and Applications(IJACSA), 15(8), 2024. http://dx.doi.org/10.14569/IJACSA.2024.01508114