# Review of ViViT Paper (GNR 650)

Uddeshya Singh (22m2152) Sachin Giroh(22m2159)

## Motivation

The paper is driven by the aspiration to extend the success of transformer architectures, specifically the Vision Transformer (ViT), to video classification tasks. The inherent spatio-temporal nature of videos presents a unique set of challenges in adapting these architectures to efficiently handle video data. The authors aim to leverage the strengths of transformers to address the increased complexity of video data and to develop models that can efficiently process and classify videos.

## Novelties

The paper introduces several novel concepts and methodologies:

- **Video Tokenization:** Introduction of innovative methods for tokenizing video data, namely "Uniform Frame Sampling" and "Tubelet Embedding," designed to capture both spatial and temporal information from videos.
  - **Uniform Frame Sampling**
    One method to tokenize the input video is to uniformly sample $nt$ frames from the video clip. Each 2D frame is embedded independently using the same method as ViT. All these tokens are concatenated together.
  - **Tubelet Embedding**
    This method extends ViT's embedding to 3D and corresponds to a 3D convolution. For a tubelet of dimension $t \times h \times w$, $nt = \frac{t}{T}$, $nh = \frac{h}{H}$ and $nw = \frac{w}{W}$ tokens are extracted from the temporal, height, and width dimensions respectively. This method fuses spatio-temporal information during tokenization, contrasting the "Uniform frame sampling" where temporal information from different frames is fused by the transformer.
- **Factorized Architectures:** Proposal of transformer-based architectures that factorize the spatial and temporal dimensions of video data, allowing the models to handle the increased complexity of video data more efficiently.
- **Model Variants:** Presentation of four distinct model architectures, each introducing a different approach to handling spatiotemporal data, with increasing levels of factorization and efficiency.
  - **Model 1: Spatio-temporal Attention**
    **Architecture:** This model straightforwardly forwards all spatio-temporal tokens extracted from the video, through the transformer encoder.
    1. Each transformer layer in this model captures all pairwise interactions between all spatio-temporal tokens.

2. However, due to modeling all pairwise interactions, the Multi-Headed Self Attention (MSA) has a quadratic complexity with respect to the number of tokens. This complexity becomes significant for video data, as the number of tokens increases linearly with the number of input frames.

- **Model 2: Factorised Encoder**
  **Architecture:** This model consists of two separate transformer encoders.
  1. **Spatial Encoder:** Models interactions only between tokens extracted from the same temporal index. After processing through $L_s$ layers, a representation for each temporal index, $h_i \in R^d$, is obtained.
  2. **Temporal Encoder:** The frame-level representations, $h_i$, are concatenated into $H \in R^{nt \times d}$, and then forwarded through a temporal encoder consisting of $L_t$ transformer layers. This encoder models interactions between tokens from different temporal indices. The output token of this encoder is then finally classified.
- **Model 3: Factorised Self-Attention**
  **Architecture:** This model contains the same number of transformer layers as Model 1. However, the key difference lies in the computation of multi-headed self-attention.
  1. **Factorised Operation:** Instead of computing multi-headed self-attention across all pairs of tokens, $z$, at layer $l$, the operation is factorized to first compute self-attention spatially (among all tokens extracted from the same temporal index), and then temporally (among all tokens extracted from the same spatial index).
  2. **Efficiency:** Each self-attention block in the transformer models spatio-temporal interactions but does so more efficiently than Model 1 by factorizing the operation over two smaller sets of elements.
- **Model 4: Factorised Dot-Product Attention**
  **Architecture:** This model retains the same computational complexity as Models 2 and 3 but has the same number of parameters as the unfactorised Model 1.
  1. **Factorised Attention:** Attention weights for each token are computed separately over the spatial and temporal dimensions using different heads.
  2. **Dot-Product Attention:** The attention operation for each head is defined to factorize the multi-head dot-product attention over spatial and temporal dimensions.

# Major Contributions

- **Bridging the Gap Between Image and Video Processing**
  The paper successfully extends the principles behind the Vision Transformer (ViT), which was designed for images, to video data. It does so by introducing novel tokenization methods and architectures that cater to the spatiotemporal nature of videos.
- **Introduction of Video Tokenization Techniques**
  The paper introduces innovative methods to convert video data into a sequence of tokens suitable for transformer architectures.
- **Development of Factorized Transformer Architectures for Video**
  Recognizing the computational challenges of handling video data with transformers, the paper proposes architectures that factorize spatial and temporal dimensions, allowing for efficient processing.

# Critical Analysis

## Strengths

- **Versatility:** The approach is versatile, offering multiple model architectures to cater to different computational needs and dataset sizes.
- **Innovation:** The introduction of novel tokenization methods and factorized architectures showcases a deep understanding of the challenges posed by video data and offers innovative solutions.

## Limitations

- **Small Data:** ViViT doesn't give good results for small data; it requires image pretraining for satisfactory performance.
- **Classification Task:** The current ViViT model is developed primarily for classification tasks, potentially limiting its applicability to other video processing tasks without modifications.
- **Complexity:** While the factorized models offer computational efficiency, they introduce added complexity in terms of architecture design and understanding, which might pose challenges for practitioners unfamiliar with transformers.
- **Data Dependency:** The models seem to have an implicit dependency on having large datasets or using strong regularization methods to prevent overfitting.
- **Direct Comparisons:** A more detailed comparison between the introduced models in terms of performance, computational efficiency, and specific use-cases might have provided clearer guidance for practitioners and researchers.