ISTVT Based Deepfake Detection

Aryan Gautam (220222)
Anuj Gaur (220181)
Naman Patidar (230679)
Khushal Karadiya (230558)

EE656 Course Project – July 2025
Mentor: Prof. Nishchal K. Verma

Abstract

This project explores a transformer-based method for detecting deepfake videos using the Image-Spatial and Temporal Vision Transformer (ISTVT) architecture. Unlike traditional CNNs that work on individual frames, our model captures both spatial and temporal patterns by combining CNN-based tokenization with decomposed spatial-temporal attention. We preprocess videos using MTCNN to extract aligned face frames and train the model on a subset of the FaceForensics++ dataset. The results show that our approach performs well in distinguishing real from fake videos, making it suitable for practical deepfake detection tasks.

1 Introduction

Deepfakes are AI-generated videos that can manipulate facial expressions and speech with high realism, raising concerns around misinformation and digital trust. As these techniques become more advanced, effective detection methods are increasingly important.

Traditional frame-based CNN approaches often miss subtle temporal inconsistencies present in deepfakes. Transformer-based models, which capture long-range dependencies, offer a better fit for video-level analysis.

In this project, we explore the ISTVT (Image-Spatial and Temporal Vision Transformer) architecture, which models both spatial and temporal patterns across video frames to improve deepfake detection.

2 Methodology

Our deepfake detection pipeline takes a sequence of video frames as input, with shape $T \times C \times H \times W$, where T is the number of frames and C, H, W are the channel, height, and width.

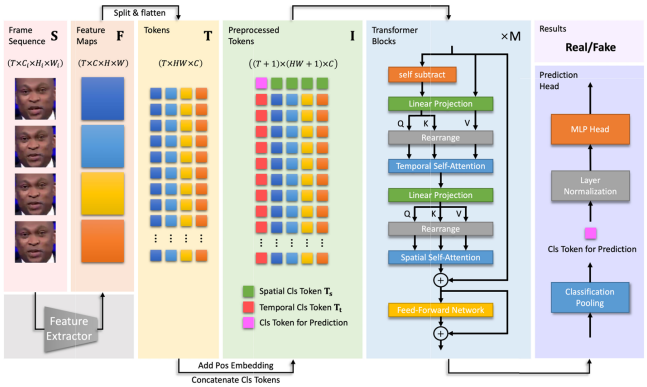

Figure 1: The overall architecture of the ISTVT framework, which combines Xception-based spatial tokenization with decomposed spatial-temporal transformer blocks for effective deepfake detection.

Since deepfakes are often generated frame-by-frame, they may contain subtle temporal artifacts. To capture both spatial and temporal cues, we use the Interpretable Spatial-Temporal Video Transformer (ISTVT) architecture, which includes:

- A CNN-based tokenizer (Xception) to extract spatial features from each frame.
- Decomposed transformer blocks for separate spatial and temporal self-attention.
- A multilayer perceptron (MLP) for binary classification.

The spatial transformer captures texture details within frames, while the temporal transformer learns inconsistencies across time. This helps the model detect deepfakes more effectively and interpretably.

2.1 Network Architecture

The ISTVT framework is designed to capture subtle textural and temporal artifacts in deepfake videos. The overall architecture is illustrated in Figure 1.

To focus on fine-grained manipulation cues, we use the entry flow of the Xception network as a lightweight CNN-based tokenizer. Given an input sequence of face frames $S \in \mathbb{R}^{T \times C_i \times H_i \times W_i}$, the model processes each frame through the Xception layers to extract feature maps $F \in \mathbb{R}^{T \times C \times H \times W}$, where C, H, and W denote the number of channels, height, and width of the output feature maps, respectively.

Each feature map is divided into 1×1 patches and flattened to form a sequence of tokens $T \in \mathbb{R}^{T \times HW \times C}$. These tokens are then augmented with two types of learnable classification tokens:

- **Spatial classification token** $T_s \in \mathbb{R}^{T \times 1 \times C}$
- **Temporal classification token** $T_t \in \mathbb{R}^{1 \times (HW+1) \times C}$

These are concatenated to form the complete input tensor $I \in \mathbb{R}^{(T+1) \times (HW+1) \times C}$, followed by the addition of positional embeddings to preserve spatial and temporal order.

The enriched token sequence is passed through 2 stacked spatial-temporal transformer blocks with decomposed self-attention. The token at position (0, 0) in the output serves as the joint spatial-temporal representation for final prediction.

2.2 Spatial-Temporal Transformer Block

Deepfake generation methods typically manipulate each video frame independently, introducing artifacts (e.g., blurred textures, abnormal facial structures) that are temporally inconsistent. To effectively capture these cues, ISTVT decomposes the attention mechanism into separate *spatial* and *temporal* self-attention operations.

The input to each transformer block is projected into query (Q), key (K), and value (V) tensors using linear layers. These tensors are then split across multiple heads, reshaped to the form $(T+1) \times (HW+1) \times N \times D$, where T is the number of frames, HW is the number of spatial patches, N is the number of heads, and $D = \frac{C}{N}$ is the dimension per head.

Temporal Self-Attention: For each spatial patch j, attention is applied across the temporal dimension to model changes over time. The operation is defined as:

$$O_t(:, j, :, :) = \text{softmax}\left(\frac{Q(:, j, :, :)K(:, j, :, :)^T}{\sqrt{D}}\right)V(:, j, :, :) \tag{1}$$

Spatial Self-Attention: Similarly, for each frame k, attention is applied over all spatial patches:

$$O_s(k, :, :, :) = \text{softmax}\left(\frac{Q(k, :, :, :)K(k, :, :, :)^T}{\sqrt{D}}\right)V(k, :, :, :) \tag{2}$$

These operations are performed efficiently by reshaping the tensors to allow matrix multiplication across the desired dimensions. This decomposition reduces the computational complexity from $\mathcal{O}(T^2H^2W^2)$ in vanilla self-attention to $\mathcal{O}(T^2 + H^2W^2)$, making the model more scalable for video input.

The output of both spatial and temporal attentions is fused and passed to subsequent transformer blocks, enabling the network to jointly learn spatial details and temporal consistency essential for deepfake detection.

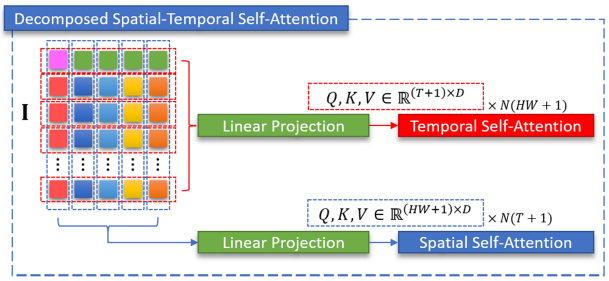

Figure 2: Spatial-temporal decomposition of self-attention in ISTVT architecture.

2.3 Self-Subtract Mechanism

To enhance the discriminative ability of the temporal self-attention, ISTVT introduces a **self-subtract mechanism** that emphasizes inter-frame inconsistencies and suppresses redundant static features. This operation computes frame-wise differences to highlight temporal distortions commonly introduced in deepfake videos.

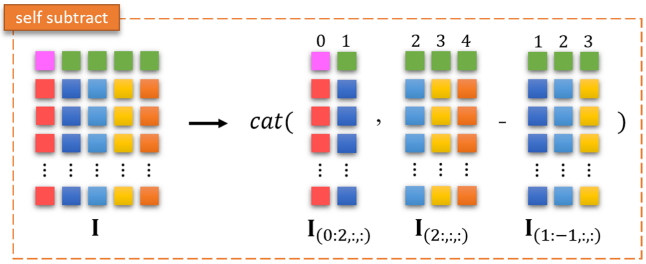

Figure 3: Self Subtract Mechanism

Given an input tensor $I \in \mathbb{R}^{(T+1) \times (HW+1) \times C}$, we compute a residual tensor I' by subtracting adjacent frames along the temporal axis:

$$I' = \text{cat}(I[0:2, :, :],\ I[2:, :, :] - I[1:-1, :, :],\ \text{dim} = 0) \tag{3}$$

Here, the classification token is preserved unaltered for the first two positions, while subsequent tokens represent residual changes. This residual tensor I' is used to compute the *queries* and *keys* in the temporal self-attention, while the *values* are derived from the original input I. This design helps the model to

focus more on dynamic inconsistencies rather than repetitive content.

Standard transformer components such as residual connections, layer normalization, and a feed-forward network are also applied. The complete spatial-temporal transformer module consists of 2 such blocks, combining both attention types with the self-subtract mechanism to enhance robustness against subtle manipulations.

2.4 Model Interpretability

Understanding model decisions is crucial in deepfake detection, where synthetic content often looks real. ISTVT enhances interpretability by using decomposed spatial-temporal attention to highlight key spatial and temporal regions influencing predictions. Inspired by transformer explainability methods like Layer-wise Relevance Propagation, we adapt them from image- to video-level to attribute decisions across both space and time.

In each of the 2 transformer blocks, relevance scores are computed for both temporal and spatial attention modules. These scores quantify how much each token contributes to the classification decision (specifically for the 'Fake' class). For each spatial or temporal position i, attention relevance is computed via:

$$\bar{A}_d^{(m)}(i) = I + \max\left(\mathbb{E}_h\left[R_d^{(m)}(:,i,:) \circ \nabla A_d^{(m)}(:,i,:)\right], 0\right) \tag{4}$$

$$U_d(i) = \bar{A}_d^{(1)}(i) \cdot \bar{A}_d^{(2)}(i) \cdot \cdots \cdot \bar{A}_d^{(M)}(i) \tag{5}$$

where $\nabla A_d^{(m)}$ is the gradient of the attention scores, $R_d^{(m)}$ denotes relevance, and $\circ$ is the element-wise (Hadamard) product.

The outputs U_t and U_s represent temporal and spatial attention relevance respectively. For visualization, only the relevance maps associated with the classification token are retained (excluding self-relevance). These maps are then aggregated, reshaped to a tensor of size $T \times H \times W$, and upscaled to the input frame resolution $T \times H_i \times W_i$ using bilinear interpolation to produce interpretable heatmaps.

These visualizations provide insights into which spatial and temporal regions the model attends to when identifying deepfakes, offering a window into the inner workings of ISTVT's decision process.

2.5 Dataset

We used a balanced subset of the FaceForensics++ dataset, a standard benchmark for deepfake detection featuring real and synthetically manipulated videos (e.g., Deepfakes, FaceSwap, Face2Face, NeuralTextures). Specifically, we selected 200 real and 200 fake videos. Aligned face frames were extracted using MTCNN and served as inputs to the ISTVT model, enabling evaluation of spatial-temporal modeling effectiveness on a diverse and representative sample..

2.6 Implementation Details

2.6.1 Data Preprocessing

We use the FaceForensics++ dataset and extract aligned face frames from both real and fake videos. For each video, up to 10 frames are uniformly sampled using OpenCV. Each selected frame is passed through the **MTCNN** (Multi-task Cascaded Convolutional Networks) face detector with a margin of 20 and resized to 224×224. The aligned facial crops are saved as tensors in structured directories, labeled by video class (real or fake).

- Frame sampling is performed at fixed intervals across the video.
- MTCNN processes each frame to detect and align a single face.
- Cropped faces are saved as `.pt` tensors with consistent naming.

2.6.2 Model Training Settings

The model is implemented in PyTorch and trained using the following configuration:

- **Optimizer:** Adam optimizer.
- **Loss Function:** Binary Cross-Entropy (BCE) with logits.
- **Learning Rate:** Fixed at 1×10^{-4}.
- **Epochs:** Trained for 15 epochs.
- **Batch Size:** 4.
- **Early Stopping:** Not explicitly applied, but the best model is selected based on minimum validation loss.

The model is trained on GPU using a custom training loop with dynamic learning rate scheduling. For each epoch, accuracy and loss are computed on both training and validation sets. The best-performing model (based on lowest validation loss) is saved automatically during training.

2.7 Detection Performance

The ISTVT-based deepfake detection model was trained for 15 epochs on the FaceForensics++ dataset. Training and validation accuracy, along with loss metrics, were tracked to evaluate convergence and generalization. Key observations include:

- The model showed rapid improvement in early epochs, reaching a validation accuracy of **91.25%** by epoch 4.
- Peak performance occurred at **epoch 13**, with a validation accuracy of **88.75%** and the lowest validation loss of **0.2214**.
- Final training accuracy reached **98.12%**, indicating strong learning of spatial-temporal patterns.
- Validation accuracy remained stable between **85%–90%** in later epochs, suggesting good generalization with minimal overfitting.

These results highlight ISTVT's effectiveness in capturing subtle spatial and temporal manipulations. The decomposed attention blocks and Xception-based tokenization contribute significantly to its classification performance.