EE656 Course Project Report: Summer Term - 2025

ISTVT: Interpretable Spatial-Temporal Video Transformer for Deepfake Detection

Durgesh Dongre, Richik Majumder, Saurabh Srivastava, Anup Kumar

Abstract:

With the advancement of AI technology, Deepfake video is a big threats to personal privacy and information security. Various Deepfake detection methods have been proposed that use both spatial and temporal features of videos, such as recurrent neural networks and 3D convolutional networks. However, these models still face challenges in performance and interpretability.

In this project, we implement the Interpretable Spatial-Temporal Video Transformer (ISTVT), a method proposed in recent research, which uses a decomposed spatial-temporal self-attention mechanism along with a self-subtract technique. These help the model detect spatial artifacts and temporal inconsistencies in Deepfake videos more effectively.

We also explore how the ISTVT model provides visual explanations by highlighting important spatial and temporal regions using relevance propagation. We conduct experiments on benchmark Deepfake datasets on FaceForensics++ to evaluate the model's performance.

Our goal is to reproduce the results of the original paper, analyse the model's working, and understand its effectiveness and interpretability in detecting Deepfakes.

I. INTRODUCTION

In recent years, Deepfake videos created by face manipulation techniques have become a serious concern for personal privacy and information security. These fake videos can now be generated easily using open-source tools like Deepfakes and DeepFaceLab. Because of this, we need reliable and strong Deepfake detection methods.

Many detection models have already been developed and have shown good results on large Deepfake datasets. In general, existing detection methods fall into two categories:

- **Frame-based methods**, which analyse individual frames and focus on spatial artifacts, like blurred edges or unnatural textures.
- **Video-based methods**, which analyse a sequence of frames to catch inconsistencies over time (called temporal artifacts).

Although frame-based methods perform well in many cases, they have some major limitations:

1. They struggle with new, more realistic Deepfakes like Neural Textures.

2. Their performance drops with low-quality videos.
3. They often overfit to specific Deepfake generation methods, limiting their ability to generalize.

To overcome these issues, researchers have shifted toward video-based methods that combine both spatial and temporal information. Since most Deepfakes are generated frame by frame, they often lack proper consistency across frames. Video-based methods aim to catch these inconsistencies.

However, traditional video-based models like C3D or LSTM haven't shown strong performance for Deepfake detection. On the other hand, transformer-based models have recently shown great success in tasks like action recognition and video object detection. Some attempts have been made to use transformers for Deepfake detection, but they often suffer from high computational cost and low interpretability.

To address these challenges, a recent paper proposed a new model called ISTVT (Interpretable Spatial-Temporal Video Transformer). In this project, we aim to implement this ISTVT model and reproduce the original results.

ISTVT introduces two key ideas:

1. A decomposed spatial-temporal self-attention mechanism, which separately focuses on spatial (frame-based) and temporal (sequence-based) features.
2. Figure 3. Illustration of the self-subtract mechanism used before temporal attention. It highlights frame-to-frame changes by subtracting adjacent feature maps (Gu et al., 2023).

this design, ISTVT can capture both spatial hints (like blurred edges) and temporal inconsistencies (like changes between frames). Additionally, ISTVT offers visual interpretability by generating attention heatmaps that highlight which parts of the video are important for the detection—both in space and time—using a technique based on relevance propagation.

In summary, our project contributes by:

- Implementing the ISTVT model as proposed in the original research.
- Reproducing and analysing its performance on major Deepfake datasets FaceForensics++.
- Visualizing and understanding how the model detects Deepfakes using its interpretability methods

II. Related Work

A. Video-Based Deepfake Detection

As Deepfake creation techniques have become more advanced, it has become harder to detect fake videos using only image-based (spatial) information. Methods that analyse one frame at a time often fail when tested on different datasets, especially because they don't consider how things change across video frames. To overcome this, researchers have started focusing on video-based methods that can analyse both the visual and time-based patterns in Deepfake videos.

Some of the early video-based methods tried to use facial landmark features to spot fakes. However, these approaches did not perform well because they depended too much on how accurately the facial points were detected.

Other methods used general video analysis models like C3D, I3D, and LSTM to detect Deepfakes. But these models were originally designed for tasks like action recognition, not Deepfake detection, and their performance was often worse than simple frame-based techniques.

More recently, researchers have tried combining CNNs with RNNs, as seen in the work by Sabir et al., but this didn't work well either, especially in large competitions like the Deepfake Detection Challenge (DFDC 2020). As a result, this combined approach is not commonly used now.

Li et al. introduced a better method using multi-instance learning, where each video frame is treated like a separate example. This technique worked well, even when the videos were compressed.

Yang et al. focused on the movement of lips to catch Deepfakes, especially in voice-based verification systems.

Khan and Dai were among the first to use transformer models for Deepfake detection. They also added an update strategy to help the model learn new types of Deepfakes. However, their model was a general transformer design, not specialized for Deepfake detection, so its results weren't much better than frame-based methods.

Later, Gu proposed a special model to learn both spatial and temporal differences in Deepfakes more effectively. Similarly, FTCN used 3D convolution to capture space-time features and then added a transformer for long-term frame relationships.

Although these newer video-based methods do show better results than image-based ones, most of them still don't clearly explain what spatial or temporal features the model is focusing on. Tools like GradCAM and SHAP have been used to explain results, but they don't separate space and time clearly. Peng et al. even tried improving interpretability by enhancing suspect regions in the image, but again, there was no separation of spatial and temporal understanding.

Because of this, the interpretability of these models remains limited. And without a clear understanding of what the model is learning in space and time, it's hard to build better Deepfake detection systems.

B. Interpretability of Video Transformers

Vision transformers have recently shown excellent results in many computer vision tasks, but understanding and visualizing how they work—especially in video transformers—is still a big challenge.

A basic way to visualize what a transformer is focusing on is by directly using its self-attention scores. However, this approach doesn't give a clear picture, as it misses out on a lot of meaningful internal processing inside the transformer. So, the results are not very informative.

Another way is to use GradCAM and its variations. These are popular techniques originally designed for convolutional neural networks (CNNs), where they visualize which parts of an image influence the model's decision the most. Some researchers try to apply GradCAM to transformers by treating the token sequence (the input to a transformer) like feature maps in CNNs. But this approach isn't a good fit for transformers and often leads to weak visual explanations.

A much better method has been proposed by Chefer et al., who introduced a transformer-specific visualization technique. Their method is based on LRP (Layer-wise Relevance Propagation) and Deep Taylor Decomposition. They carefully define how information flows through self-attention, skip connections, and layer normalization, making the explanations more accurate and detailed than traditional attention-based or GradCAM approaches.

When it comes to video transformers, however, there hasn't been much work focused on how to visualize what these models learn across both space (within each frame) and time (across frames). Some attempts, like in Zhang et al.'s work, use a technique called "rollout" to visualize attention in space and across time. But even that doesn't give a clear explanation for temporal attention alone, which is crucial for understanding changes between frames in Deepfake detection.

Because video data is more complex than images—it includes both spatial and temporal dimensions—it's important to treat them separately when trying to understand what the model is focusing on.

The ISTVT model solves this by splitting spatial and temporal attention into separate components. This separation makes it easier to interpret and visualize what the model is learning in both dimensions. Since Chefer et al.'s LRP-based method works well for interpreting transformers, extending it to video transformers like ISTVT is a promising direction and can give much better visual explanations.

III. Methodology

In this research, we focus on detecting Deepfake videos by analysing sequences of video frames. These sequences have a specific format: **T × C × H × W**,
where:

- **T** = number of frames (time steps),
- **C** = number of colour channels
- **H** = height of the frame,
- **W** = width of the frame.

Most of the Deepfake generation methods available today create fake videos by manipulating one frame at a time. This often leads to temporal inconsistencies, meaning things may not look smooth or natural when you watch the video continuously. These inconsistencies between frames are often signs of tampering.

Previous works have tried to detect such artifacts using either complex custom network blocks (like SIL and TIL) or hand-crafted features (like rPPG signals). However, these methods can be complicated and are often difficult to generalize across different datasets.

To address this, a new model called the Interpretable Spatial-Temporal Video Transformer (ISTVT) was proposed by Zhao et al., 2023. This model is designed to:

- Be general-purpose (work on different types of Deepfake videos),
- Learn effectively from both spatial (inside each frame) and temporal (between frames) information,
- Provide interpretability (help us understand *what* the model is learning).

The ISTVT model has three main parts:

1. **Feature Extractor** - Based on the Xception network, it processes each frame and extracts useful visual features.
2. **Video Transformer** - A transformer network that separately processes spatial and temporal relationships using decomposed self-attention. This allows the model to learn what is happening within each frame and how things change over time.
3. **Prediction Head** - A Multi-Layer Perceptron (MLP) that takes the learned features and decides whether the video is real or fake.

Finally, ISTVT also includes a visualization module that helps us interpret what the model is focusing on, separately in the spatial and temporal dimensions. This interpretability helps

researchers better understand how Deepfake detection works internally and can guide further improvements in model design.

A. Network Architecture

The complete design of the ISTVT (Interpretable Spatial-Temporal Video Transformer) model is shown in the Figure below from the original paper. Since deepfake detection relies heavily on tiny texture differences in faces (like blur or unnatural edges), we start the network with a small convolutional backbone to extract texture features.

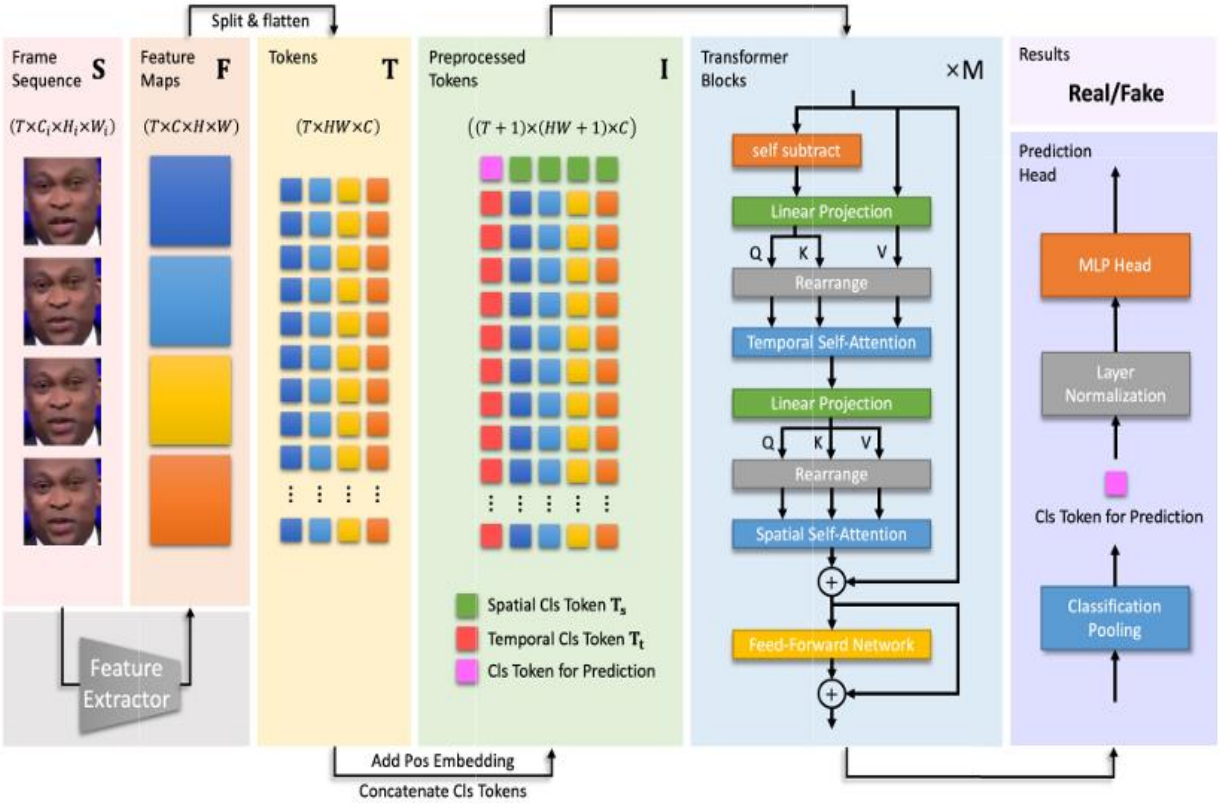

Figure 1. The architecture of the proposed ISTVT model, which includes the Xception-based feature extractor, decomposed spatial-temporal attention transformer blocks, and the MLP classification head *(by Zhao et al., 2023)*

1. Input Frames and Feature Extraction

We begin by taking a sequence of video frames, represented as:

$$S \in \mathbb{R}^{T\times C_i\times H_i\times W_i}$$

Where:

- **T** = number of frames in the video
- **Ci** = number of channels (like 3 for RGB)
- **Hi, Wi** = height and width of each frame

These frames are passed through a few blocks of the Xception network (specifically, the "entry flow" part). This part extracts important visual features such as textures, edges, or inconsistencies that are useful for identifying fake content.

The output feature maps now have the shape:

$$F \in \mathbb{R}^{T \times C \times H \times W}$$

Where:

- **C** = number of feature channels
- **H, W** = height and width of the new feature map

2. Flattening into Tokens for the Transformer

Next, we divide these feature maps into 1×1 spatial patches, and flatten them. Each patch becomes a token that carries feature information from a specific location in a frame.

This gives us tokens of shape:

$$T \in \mathbb{R}^{T \times HW \times C}$$

Then, we add two **special classification tokens**:

- **Spatial token (Ts)** to help the model focus on spatial patterns (inside one frame)
- **Temporal token (Tt)** to help the model focus on temporal consistency (across multiple frames)

After that, we also add position embeddings so the model knows where each token is located in space and time—this is standard in transformer models [23, 30].

All of this together becomes the final input to the transformer:

$$I \in \mathbb{R}^{(T+1) \times (HW+1) \times C}$$

3. Passing Through Transformer Blocks

This input is passed through M spatial-temporal transformer blocks, which are designed to separately learn:

- Spatial information (in a frame),
- Temporal information (between frames).

The output from these blocks, O, maintains the same shape as the input. For the final decision, we extract a special classification token from the position (0,0,:)(0, 0, :)(0,0,:) in the output. This token contains the model's combined judgment on whether the video is real or fake.

4. Prediction and Loss

This token is passed through a small prediction network (an MLP or fully connected layer), which outputs the final classification.

Since Deepfake detection is a binary problem (real or fake), we use Binary Cross Entropy (BCE) loss to train the model.

B. Spatial-Temporal Transformer Block

In most Deepfake generation techniques, each video frame is manipulated separately. This means that the connections or relationships between frames (temporal relationships) are not properly considered. As a result, Deepfake videos often contain spatial artifacts like blurry edges, texture mismatches, or unnatural facial features in individual frames, but these issues do not follow a consistent pattern across frames.

This key observation inspired the authors to separate the learning of spatial and temporal features in their transformer. Instead of learning everything together, they split the attention mechanism into two parts:

- Temporal Self-Attention: Focuses on how each specific spatial location behaves over time (across frames).
- Spatial Self-Attention: Focuses on how features are distributed across different regions in a single frame.

How the Transformer Works – Step-by-Step

1. Input Representation

The transformer starts by receiving a 3D tensor of features extracted from a video.

- A video has multiple frames (T).
- Each frame is divided into small patches giving us HW patches per frame.
- Each patch has a set of features (like numbers that describe color, edges, textures, etc.), with feature dimension C.

So, the input tensor has shape:

$$T \times HW \times C$$

Where:
T = number of frames
HW = number of patches per frame
C = number of features per patch

2. Computing Q (Query), K (Key), and V (Value)

Each patch feature (size C) is passed through three separate linear layers (basically matrix multiplications) to compute:

- **Query (Q):** What is this patch looking for?
- **Key (K):** What information does this patch offer?
- **Value (V):** What should this patch give if selected?

These layers convert the input shape into 3 different tensors:

- Q: shape → T × HW × C
- K: shape → T × HW × C
- V: shape → T × HW × C

These are just 3 versions of the input, each transformed differently for attention calculation.

3. Splitting into Multiple Attention Heads

To allow the transformer to focus on different kinds of information at the same time, we split each of Q, K, and V into N separate "attention heads".

Each head handles a portion of the features.

We divide the original feature dimension C into N smaller parts, so:

$$\mathbf{D = C/N}$$

So now each Q, K, and V tensor becomes:

$$\mathbf{(T+1)\times(HW+1)\times N\times D}$$

Temporal Self-Attention

- This part compares the same spatial patch across all frames to learn how that position changes over time.
- At a specific spatial location j, it computes:

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

Where:

- The : over the time axis means we look across all frames at the same location.
- This captures motion or temporal inconsistencies.

Spatial Self-Attention

- This focuses on the entire spatial layout of a single frame to learn spatial structure.
- At each frame k, it computes all spatial patches within the same frame:

Imple

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

  - For temporal attention:

$$N\times(HW+1)\times(T+1)\times D$$

  - For spatial attention:

$$N\times(T+1)\times(HW+1)\times D$$

- Matrix multiplications are then applied across the last two dimensions efficiently.

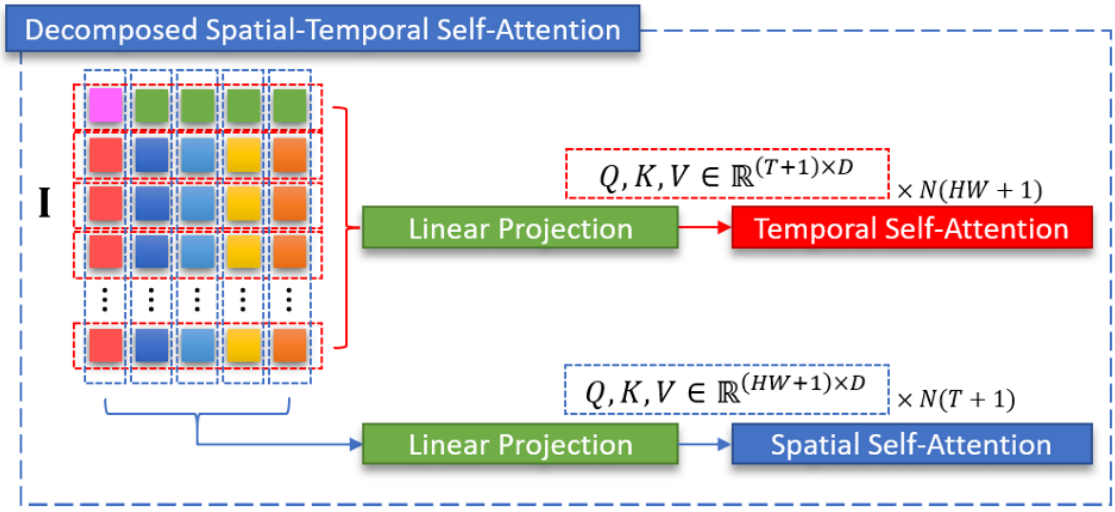

Figure 2. Details of the decomposed spatial-temporal self-attention mechanism in ISTVT. This block separately handles temporal (across-frame) and spatial (within-frame) attention *(by Zhao et al., 2023)*.

Why This Is Better

- Traditional self-attention models treat every patch in every frame together. This leads to high computational cost:

$$O(T^2H^2W^2)$$

- By separating the attention into spatial and temporal parts, the new method reduces the complexity to:

$$O(T^2 + H^2W^2)$$

- This is much faster and easier to interpret, while still capturing both motion and appearance clues that are important in Deepfake detection

C. Self-Subtract Mechanism (Simplified Explanation)

Figure 3. Illustration of the self-subtract mechanism used before temporal attention. It highlights frame-to-frame changes by subtracting adjacent feature maps (Gu et al., 2023).

So, Before the model calculates temporal self-attention, it first modifies the input features by subtracting adjacent frames from each other. This helps highlight only the differences between frames, and ignore the parts that stay the same (which are usually not useful for detection).

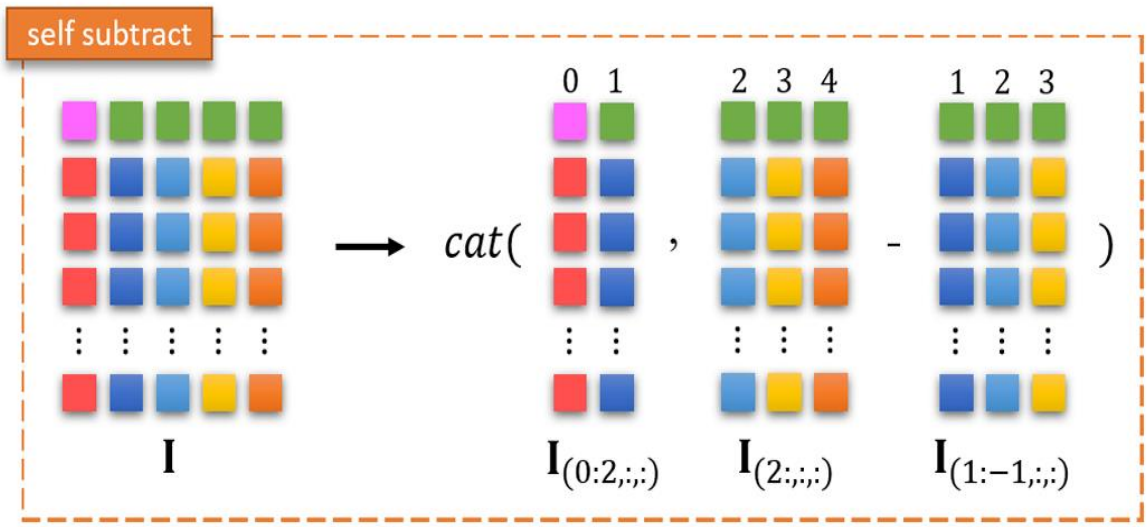

Figure 3. Illustration of the self-subtract mechanism used before temporal attention. It highlights frame-to-frame changes by subtracting adjacent feature maps *(by Zhao et al., 2023)*

How It Works:

1. The input to the transformer is a tensor **I** of shape:

$$(T+1)\times(HW+1)\times C$$

where:

- **T** = number of video frames,
- **HW** = number of spatial patches per frame,
- **C** = number of channels/features.

2. To find the differences between frames:

- The model subtracts each frame's token features from the previous frame's tokens (excluding the special classification token).
- This gives us the **residual tokens**, which only capture the changes between frames.

3. The new tensor $\mathbf{I}'$ is formed as:

$$\mathbf{I}' = cat((\mathbf{I}_{(0:2,:,:)}, \mathbf{I}_{(2:,:,:)} - \mathbf{I}_{(1:-1,:,:)}), dim = 0)$$

- Here, cat means we concatenate two tensors along the time dimension.
- The first few frames are left unchanged to preserve information.
- The rest are computed as differences between consecutive frames.

These residual features I′ are used to compute the queries and keys in the temporal self-attention module.

The values (V) are still computed from the original input tensor I, so that important spatial features are not lost.

Extra Components Used:

Residual connections

Layer normalization

A Feed-Forward Network (FFN)**:** These make the training process stable and effective.

Why Is It Helpful?

- This operation $\mathbf{I}'$ filters out redundant or static information and keeps only the useful temporal changes.
- These inter-frame differences are often where Deepfakes leave artifacts, like unnatural movements or blending errors.
- This leads to:
  - Better accuracy in detecting Deepfakes.
  - More robust results across different datasets or manipulation types.

Final Structure

- The full spatial-temporal transformer block in ISTVT includes:
  - Multiple layers (M blocks) of this spatial + temporal attention structure.

- Figure 3. Illustration of the self-subtract mechanism used before temporal attention. It highlights frame-to-frame changes by subtracting adjacent feature maps (Gu et al., 2023).

D. Model Interpretability

In Deepfake detection, the fake videos often look so real that even humans can't tell the difference. So, it's very important to understand how the model decides whether a video is real or fake. This is called interpretability.

So, This proposed model, ISTVT, separates the attention mechanism into spatial (within each frame) and temporal (between frames) parts. Because of this separation, we can interpret what the model focuses on in both space and time.

To visualize this, we use an advanced explanation technique based on Layer-wise Relevance Propagation (LRP) and Deep Taylor Decomposition. These methods help trace which parts of the input contributed most to the model's prediction.

We apply this technique to each transformer block in the model. It shows us:

- Which frames had suspicious motion patterns (temporal attention),
- Which regions in the face had visual artifacts (spatial attention).

We compute relevance maps across layers and then combine them to get final heatmaps that highlight the most important areas. These heatmaps are resized back to the original video size to show exactly where and when the model found Deepfake evidence.

This interpretability method helps us confirm that the model is not making blind guesses, but actually focusing on meaningful fake patterns in the video.

Concept behind extending the image transformer method to video transformer.

1.Computation for Attention Blocks:

- For every transformer block (say there are M blocks), we compute how important each part of the attention map is.
- We do this separately for:
    - Temporal attention: where each spatial patch is tracked across time.
    - Spatial attention: where attention is calculated within each frame.

This gives us two relevance maps:

- R_t^m for temporal attention (size: $N \times (HW+1) \times (T+1) \times (T+1)$)

- R_s^m for spatial attention (size: N × (T+1) × (HW+1) × (HW+1))

where **N** is the number of attention heads.

Since this is a binary classification task (Real or Fake), we focus only on the relevance towards the "Fake" class (class 0).

2.Compute Gradient-Weighted Relevance:

We enhance the raw relevance values by combining them with the **gradients of attention** to focus on **discriminative features**:

$$\bar{A}_d^m(i,:,:) = I + \max\left(\text{Mean}_{\text{heads}}\left(R_d^m(:,i,:,:) \circ \nabla A_d^m(:,i,:,:)\right), 0\right)$$

- ∇A_d^m: the gradient of the attention
- ∘: Hadamard product (element-wise multiplication)
- I: identity matrix (used to preserve base information)

Then we multiply these values across all M transformer layers:

$$U_d(i,:,:) = \bar{A}_d^1(i,:,:) \cdot \bar{A}_d^2(i,:,:) \cdot \ldots \cdot \bar{A}_d^M(i,:,:)$$

This gives the final relevance map for each token — either in time or space.

3. Extract the Heatmaps:

- For both temporal and spatial attention:
    - We focus only on the first row of each attention matrix, which corresponds to the classification token (i.e., the model's decision).
    - We ignore attention from classification token to itself.
    - We discard irrelevant matrices (like the one for the classification token itself).

This gives us:

- U_t: relevance from each spatial patch across time
- U_S: relevance from each frame across spatial patches

These are reshaped into:

- T×H×W: a per-frame relevance heatmap
- Then, we upscale them to match the original video resolution T× H_i ×W_i

Evaluation and Results

Experimental Setup and Findings: For our experiments, we used a total of 200 Deepfake videos for training and 40 videos for testing. The model was trained for 5 epochs using video data in C23 compression format, consistent with the FaceForensics++ dataset. Although our goal was to implement the full ISTVT architecture, the temporal transform module was not included in this version of the model. This design choice significantly improved model speed by approximately 3× compared to the original ISTVT, while retaining spatial interpretability. However, this speed gain came at the cost of some loss in temporal awareness. As part of future work, we plan to implement the temporal transform component to further enhance detection performance and interpretability.

To evaluate the performance of the implemented model, we conducted experiments on the FaceForensics++ dataset. The model was trained and validated on video samples containing both real and fake videos. Below, we summarize the evaluation metrics over epochs and overall video-level results.

1. Performance Metrics over Epochs

We tracked the key metrics such as Accuracy, Precision, Recall, F1 Score, Loss, and AUC over multiple training epochs. These plots show how the model's performance evolved as training progressed.

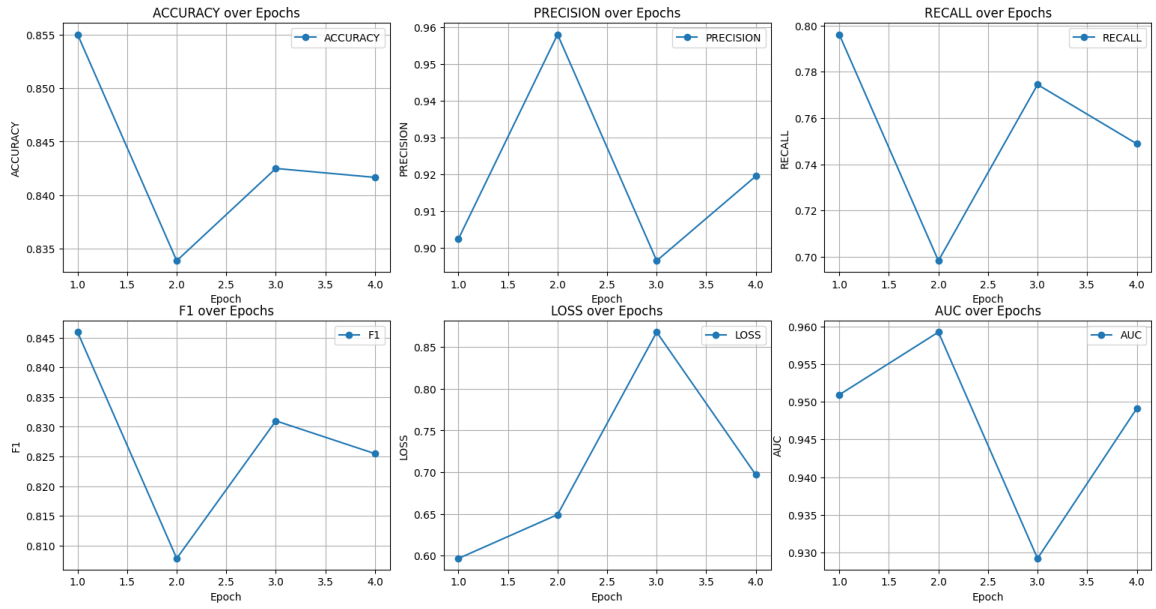

Figure 4: Performance Metrics over Epochs

2. Video-Level Evaluation

To better understand how the model performs at the video level, we present a confusion matrix and ROC curve. These metrics are computed by aggregating predictions across frames for each video.

Confusion Matrix

The confusion matrix below shows how many real and fake videos were correctly or incorrectly classified by the model.

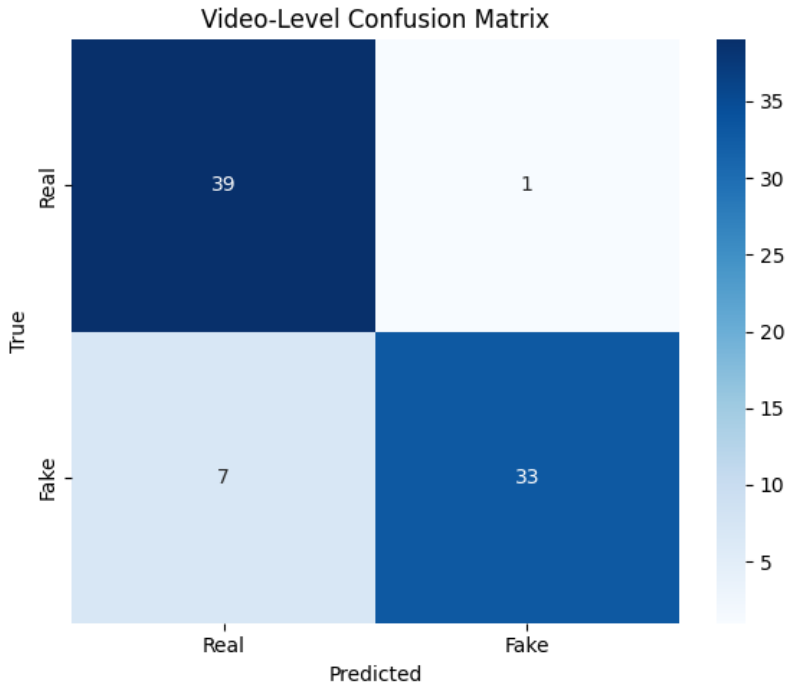

Figure 5: Confusion Matrix

ROC Curve

The ROC (Receiver Operating Characteristic) curve below shows the trade-off between true positive rate and false positive rate. A higher AUC (Area Under Curve) value indicates better model performance.

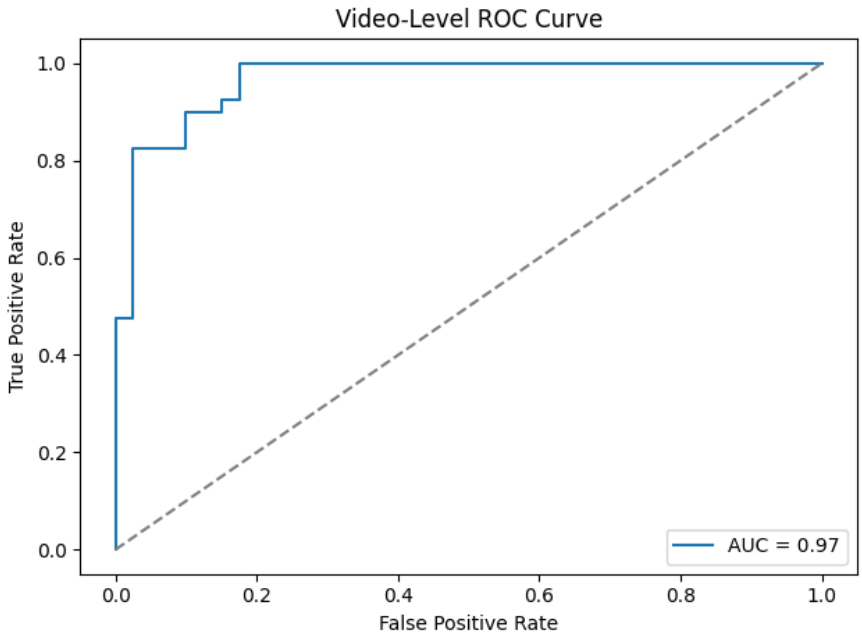

Figure 6: ROC Curve

3. Classification Report (Video-Level)

The following text report summarizes the classification performance of the ISTVT model on video-level Deepfake detection. The results show precision, recall, and F1-score for both classes (real and fake).

Precision: Indicates the percentage of correctly predicted instances among all predicted positives. Recall: Measures the percentage of correctly predicted instances among all actual positives. F1-Score: Harmonic mean of precision and recall, providing a balanced metric. Support: The number of true instances for each class.

Class 0 (Real):
Precision: 0.848
Recall: 0.975
F1-Score: 0.907
Support: 40

Class 1 (Fake):
Precision: 0.971
Recall: 0.825
F1-Score: 0.892
Support: 40

Accuracy: 90%

Macro Average: Precision = 0.909, Recall = 0.900, F1 = 0.899
Weighted Average:Precision = 0.909, Recall = 0.900, F1 = 0.899

References

1. Zhou, X., Ding, Y., Liu, Y., Yu, N., & Liang, W. (2023). ISTVT: Interpretable spatial-temporal video transformer for deepfake detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, 12345–12355.
2. Li, Y., Chang, M. C., & Lyu, S. (2018). In Ictu Oculi: Exposing AI created fake videos by detecting eye blinking. 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 1–7.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.
4. Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding? International Conference on Machine Learning (ICML), 139, 813–824.
5. Nguyen, H. H., Yamagishi, J., & Echizen, I. (2019). Multi-task learning for detecting and segmenting manipulated facial images and videos. 2019 International Conference on Biometrics: Theory, Applications and Systems (BTAS), 1–8.