DSViT: An Enhanced Transformer Model for Deepfake Detection

DOI: https://doi.org/10.54654/isj.v2i22.1055

Pham Minh Thuan, Bui Thu Lam, Pham Duy Trung*

***Abstract*— The rapid development of artificial intelligence and deep learning models has enabled the creation of highly realistic fake images and videos, posing significant threats to information security and safety. Accurate detection of these forged contents is crucial to prevent the spread of misinformation and to protect the integrity of digital media. Although several advanced studies in this field, such as Vision Transformer (ViT) and Convolutional Vision Transformer (CViT), have been conducted, there remain limitations that need to be addressed. In this paper, we introduce a novel model, improved from CViT, designed to optimize the process of deepfake detection, named DSViT (Deepfake Detection with SC-based Convolutional Vision Transformer). This model judiciously integrates Convolutions and a SCConvolution block with the ViT architecture. We conducted experiments on the Deepfake Detection Challenge (DFDC) dataset and compared the results with the CViT model to demonstrate the effectiveness of the proposed model.**

***Tóm tắt*— Sự phát triển nhanh chóng của trí tuệ nhân tạo và các mô hình học sâu đã cho phép tạo ra các hình ảnh, video giả mạo siêu thực, đe dọa nghiêm trọng đến an toàn, an ninh thông tin. Việc phát hiện chính xác các hình ảnh, video giả mạo này là rất quan trọng để ngăn chặn lan truyền thông tin sai lệch và đảm bảo tính toàn vẹn của phương tiện kỹ thuật số. Hiện đã có nhiều các công trình nghiên cứu tiên tiến về phát hiện deepfake như ViT, CViT, song vẫn còn có những hạn chế nhất định, dẫn đến cần tiếp các công trình nghiên cứu để cải tiến thêm. Bài báo này đề xuất một mô hình dựa trên cải tiến từ mô hình CViT để tối ưu hóa cho việc phát hiện deepfake, gọi là DSViT (phát hiện deepfake với SC-based Convolutional Vision Transformer). Mô hình đề xuất sử dụng các khối Convolution và SCConvolution được sắp xếp một cách hợp lý kết hợp với kiến trúc ViT. Chúng tôi đã thử nghiệm trên bộ dữ liệu DFDC và so sánh kết quả với mô hình CViT để chứng minh hiệu quả của mô hình.**

Keywords*— *deepfake detection; DSViT; deepfake; spatial deepfake.

Từ khóa*— *phát hiện deepfake; DSViT; deepfake; deepfake dựa trên không gian.

I. INTRODUCTION

In recent years, deepfake has emerged as a significant challenge for the computer science and cybersecurity communities, garnering substantial attention from both researchers and policymakers. Deepfake technology leverages deep learning models to create highly realistic and convincing fake videos, where a person's face or voice is altered or replaced to generate entirely new content. The capabilities of this technology have surpassed previous limits, making it exceedingly difficult to distinguish between real and fake, even for experienced professionals [1].

The danger of deepfake lies not only in its ability to deceive but also in its potential to cause serious harm across various domains, including politics, media, and personal life. Deepfake videos have been used to disseminate misinformation, perpetrate fraud, and extort individuals [2]. This situation underscores the urgent need for developing effective tools and methods for deepfake detection, particularly as this technology continues to evolve and become more sophisticated.

To counter the increasingly complex deepfake techniques, numerous studies have

The paper was reviewed, accepted and introduced by the XXVII National Conference "Some Selected Issues on Information and Communication Technology" to be published in the Journal on September 13, 2024, revised on September 26, 2024 and accepted on September 29, 2024.

* Corresponding author

focused on employing deep learning models to enhance detection accuracy and efficiency. One prominent approach involves the combination of Convolutional Neural Networks (CNNs) and Vision Transformer, aiming to leverage the strengths and mitigate the weaknesses of both architectures. Among these hybrid models, the CViT has gained recognition as a powerful tool for deepfake detection [3]. Specifically, CViT is designed to extract local features using CNNs while leveraging the global attention mechanisms of ViT to learn spatial relationships across image frames. However, despite its promising performance, CViT faces several critical limitations.

First, the CViT architecture, though effective at combining CNNs and ViT, suffers from high computational complexity due to its deep and intricate structure. This results in a high demand for memory and processing power, which can limit its applicability in real-time detection scenarios or environments with limited resources. Second, CViT struggles with handling the multidimensional nature of video data, particularly in scenarios where variations in lighting conditions, camera angles, and rapid motion are present. These factors lead to inconsistencies in feature extraction and, as a result, reduce the model's robustness in detecting deepfakes under diverse real-world conditions. Additionally, CViT's reliance on fixed convolutional operations makes it less adaptable to the dynamic nature of deepfake videos, where both subtle and large variations in facial expressions and movements are common.

To address these limitations, we propose an enhanced model, DSViT, which improves upon CViT by introducing several key modifications. First, we optimize the ConvBlock by incorporating a 3x3 convolution, along with Batch Normalization, the Mish activation function, and Max Pooling. Second, we add a ScconvBlock for spartial reconstruction and channel reconstrucstion. These techniques enable the model to dynamically adjust the importance of features and concentrate on the most informative regions of the video frame, thereby improving its ability to detect subtle deepfake artifacts. Additionally, we restructure the feature learning process to enhance both efficiency and relevance, which in turn reduces overall computational complexity while maintaining high accuracy. Our contributions include:

- Proposing a novel model, called DSViT, that is built on the CViT architecture by incorporating advanced feature recalibration namely reducing both spatial and channel redundancy, improving both accuracy and efficiency in detecting deepfake videos.

- Providing a comprehensive performance comparison between DSViT and the original CViT model, demonstrating that DSViT not only increases detection accuracy but also reduces processing time, thereby making it more suitable for real-world deepfake detection scenarios.

II. Related Works

1. The CViT Model

The Convolutional Vision Transformer represents a hybrid model that integrates CNN with ViT for the task of deepfake detection [4]. By combining these two architectures, CViT leverages the strengths of CNN in extracting localized, spatial features and the global attention capabilities of ViT to model dependencies across multiple frames in a video. The convolutional layers in CViT serve as the backbone for feature extraction, capturing fine-grained details from input video frames.

However, despite its promising performance, the convolutional blocks within CViT present several limitations. One significant drawback lies in the fixed nature of the convolutional operations, which makes the model less flexible when dealing with the dynamic nature of deepfake videos. Variations in facial expressions, movements, or lighting conditions often challenge the convolutional layers, as they are not inherently adaptable to such changes. Furthermore, these convolutional layers are parameter-heavy, with some layers requiring millions of parameters to extract features. For instance, a single 512-channel convolutional layer contains over 2.35 million parameters. This high parameter count, while beneficial for learning intricate features, increases the computational burden, making the model less

feasible for real-time deepfake detection or deployment in resource-constrained environments.

Additionally, CViT's convolutional layers face challenges in handling the multidimensional nature of video data. Variations in environmental factors, such as lighting and camera angles, can cause inconsistencies in feature extraction, reducing the model's robustness in diverse, real-world conditions. As a result, while CViT demonstrates notable accuracy in controlled settings, its computational complexity and inflexibility in adapting to dynamic video content limit its scalability. These shortcomings underscore the need for further optimization, particularly in the convolutional components, to enhance both the efficiency and generalization capabilities of the model.

2. Other Models for Deepfake Detection

The rise of deepfake technology has spurred extensive research into detection methodologies, ranging from traditional handcrafted feature approaches to advanced deep learning models. Initial efforts in deepfake detection focused on identifying visual inconsistencies, such as irregularities in lighting, unnatural facial movements, or texture discrepancies. However, as deepfake generation techniques evolved, these early methods struggled to generalize across different types of deepfakes, particularly when faced with more sophisticated manipulations.

With the advent of deep learning, CNN-based models have become the predominant approach for deepfake detection. One of the most widely used models is XceptionNet, originally designed for image classification tasks [5]. XceptionNet employs depthwise separable convolutions, which reduce computational cost while maintaining the model's ability to capture manipulation artifacts. However, like many CNN-based models, XceptionNet has limitations in handling temporal dependencies across video frames, which is critical for detecting subtle inconsistencies in dynamic video sequences [6].

Another prominent CNN-based approach is MesoNet, which was specifically designed for real-time deepfake detection [7]. MesoNet employs a lightweight architecture to prioritize speed, making it suitable for environments with limited computational resources. However, the trade-off for this efficiency is a reduction in accuracy, particularly when detecting more subtle or high-resolution deepfake manipulations. This balance between speed and accuracy is a common limitation among CNN-based models when compared to more sophisticated architectures like CViT.

EfficientNet, with its compound scaling strategy, introduced significant improvements in computational efficiency while maintaining accuracy [8]. However, despite these enhancements, EfficientNet shares a limitation with other CNN-based models in its inability to capture global relationships between frames, which is crucial for detecting temporal inconsistencies in deepfake videos. This issue becomes particularly pronounced in complex scenes where subtle changes in motion or facial expressions must be tracked over time.

To address the limitations of CNNs in capturing long-range dependencies and temporal inconsistencies, researchers have turned to ViT [9]. ViT, with their self-attention mechanisms, have demonstrated superior performance in capturing both local and global features from video data. These models excel at learning long-range interactions between frames, which is crucial for detecting deepfakes. However, ViT typically require substantial amounts of training data and computational resources, limiting their applicability in real-time scenarios or resource-constrained environments [10]. Additionally, ViT may struggle with overfitting on smaller datasets, particularly when there is a lack of diverse training examples.

Hybrid models, such as the Swin Transformer [11] and Pyramid Vision Transformer (PVT) [12], have been proposed to improve the balance between local and global feature learning. The Swin Transformer introduces a shifted window mechanism to handle high-resolution images more efficiently, while PVT adopts a hierarchical structure to facilitate global relationship learning. Despite these advancements, both models face challenges in real-time performance and computational efficiency, particularly in large-

scale video datasets, similar to the issues observed with CViT.

In conclusion, while deep learning-based models like XceptionNet, MesoNet, and ViT have made significant strides in deepfake detection, each comes with its own set of limitations. CNN-based models, while efficient in local feature extraction, often struggle with temporal dependencies. Transformer-based models excel in capturing global relationships but require significant computational resources. Hybrid models like CViT, which combine the strengths of both CNNs and Transformers, offer a promising direction but still face challenges related to computational efficiency and adaptability. Future research should focus on optimizing these hybrid architectures to enhance both performance and scalability, particularly for real-time, resource-constrained deepfake detection applications.

III. Proposed Deepfake Detection Model

In this section, we present our proposed model for video deepfake detection. Our model is developed based on the CViT framework [10], wherein we have restructured the convolutional blocks to integrate SCConv, a spatial and channel reconstruction convolution architecture proposed in the paper "SCConv: Spatial and Channel Reconstruction Convolution for Feature Redundancy" by J. Li et al. [13]. The architectural details of the model are illustrated in Figure 1.

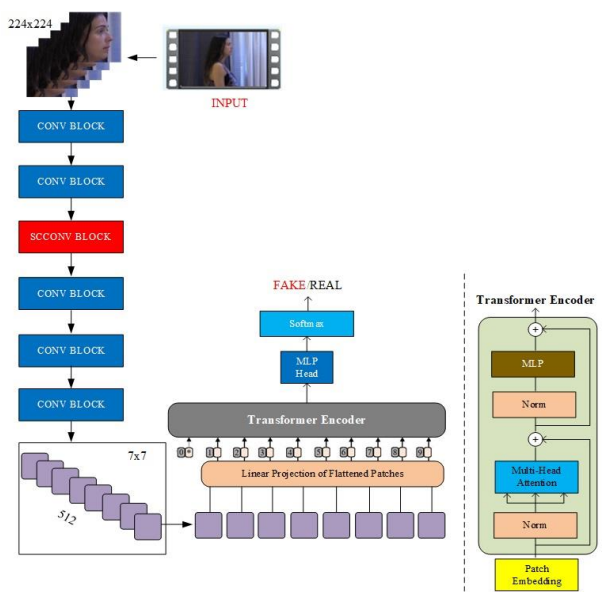

Figure 1. The architecture of DSViT model

The proposed architecture, DSViT, comprises key components: Feature Learning (including ConvBlocks and a SCConvBlock) and a Vision Transformer. Each component plays a crucial role in enhancing the overall performance of the model.

A. *Feature Learning*

Feature learning is a set of convolutional operations designed to extract important features from frames before feeding them into the Vision Transformer model. Unlike the CViT model, which uses a VGG architecture with 17 convolutional layers of size 3x3, our model only utilizes five 3x3 convolutional layers within ConvBlocks, combined with one SCConvBlock. This significantly reduces the number of parameters and computational complexity of the model while still achieving high efficiency. The detailed structure of feature learning blocks in Figure 2.

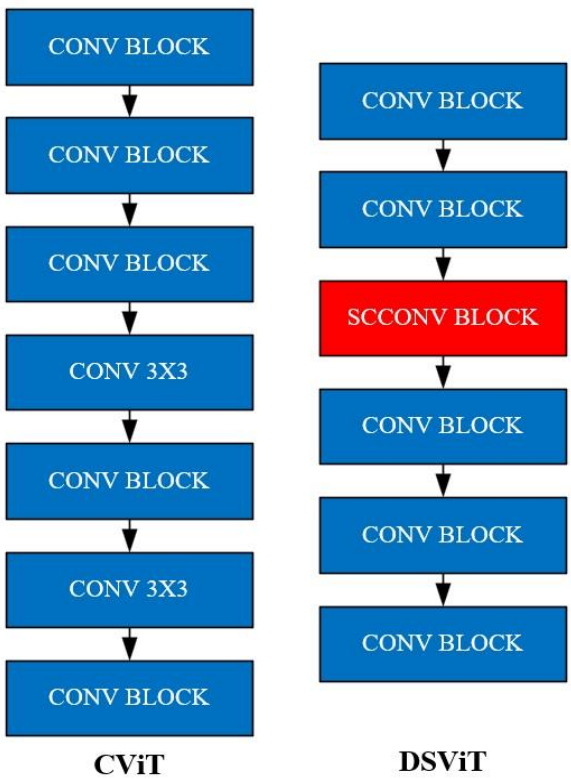

Figure 2. Structure of the Feature Learning Blocks

1. *ConvBlocks*

The ConvBlock in the DSViT model has been redesigned to optimize computational efficiency and enhance deep learning capabilities while reducing the risk of overfitting. In the

original CViT architecture, the ConvBlock consists of three repeated Conv 3x3 layers followed by Batch Normalization, ReLU activation, and MaxPooling layers. This design, while effective for feature extraction, significantly increases the model's complexity, resulting in a total of 10,813,248 parameters. Such a large number of parameters can lead to high computational costs and an elevated risk of overfitting, particularly when the dataset size is not sufficiently large or diverse. The complexity of this architecture may also necessitate more extensive computational resources and training time, further emphasizing the need for optimization in scenarios where data availability or computational power is limited.

Moreover, the use of the ReLU activation function can result in “dead neurons”, where some neurons do not reactivate during training, or cause the loss of information from negative values, potentially leading to the omission of critical information for feature learning.

To address these issues, we redesigned the ConvBlock to include four layers: Conv 3x3, BatchNorm, Mish activation, and MaxPooling. This new structure significantly reduces the number of parameters, bringing the total down to 1.578.176, which in turn decreases computational costs and training time, and mitigates the risk of overfitting. Specifically, the use of the Mish activation function retains negative values, improving the model’s convergence and reducing the vanishing gradient problem issues that are particularly critical in deep networks.

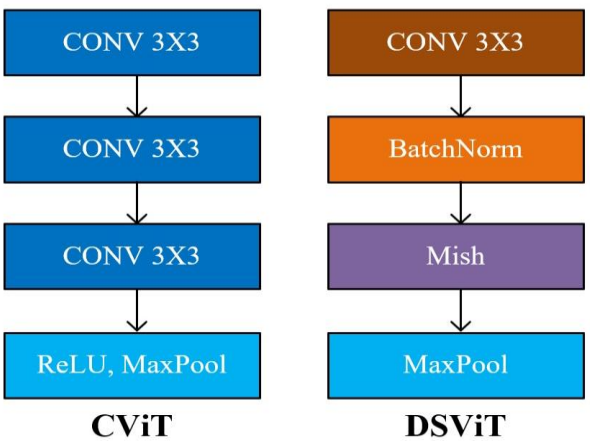

Figure 3. Components in the ConvBlock

a. Conv 3x3

The convolutional layer with a kernel size of 3×3 is designed to extract spatial features from the input image. The input to the ConvBlock consists of tensors with dimensions H, W, C, where H represents the height, W the width, and C the number of input channels. This convolution operation is performed with a 3×3 kernel, a stride of 1, and padding of 1, ensuring that the spatial dimensions of the input are preserved. This convolution helps to extract features such as edges, corners, and fine details from the image, which can be mathematically represented as follows:

$$Y[i, j] = \sum_{m=1}^{M} \sum_{n=1}^{N} X[i+m, j+n] \cdot W[m, n] + b \qquad (1)$$

where $Y[i, j]$ represents the output at position i, j, $X[i+m, j+n]$ denotes the input at the corresponding position, $W[m, n]$ is the filter weight, and b is the bias term.

b. Batch Normalization

After feature extraction, the output from the 3×3 convolutional layer is passed through the Batch Normalization (BatchNorm) layer to normalize these features. BatchNorm reduces the variance between the output scales of the previous layers, stabilizing the training process and preventing the gradient from either vanishing or exploding during backpropagation. This process is mathematically represented by the following equation:

$$BN(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \qquad (2)$$

where μ and σ are the mean and standard deviation of the input, respectively, ϵ is a small constant to avoid division by zero, and γ and β are learnable parameters.

c. Mish Activation Function

An important enhancement in the DSViT model is the use of the Mish activation function instead of ReLU. Mish is a smooth, non-truncated activation function that preserves

negative values throughout the training process and enhances gradient stability, particularly in deep networks. The activation function is defined as follows:

$$Mish(x) = x \cdot \tanh(\ln(1+e^{x})) \quad (3)$$

The Mish activation function, defined above, allows the model to retain more information compared to ReLU, while also stabilizing the gradient flow during training.

d. MaxPooling

Finally, the output from the Mish activation function is passed through a MaxPooling layer to reduce the spatial dimensions of the output tensor. The MaxPooling layer reduces the model's complexity by retaining only the maximum value in each 2×2 region, focusing on the most salient features:

$$Y[i,j] = \max_{m,n\in\{1,2\}} X[i+m, j+n] \quad (4)$$

The MaxPooling operation helps the model focus on the strongest features while reducing the computational complexity in subsequent layers.

The first two ConvBlocks are used to extract basic features from the input image with a size of 224×224. During this process, the number of channels increases from 3 to 32 and 64 through convolution layers, allowing the model to capture diverse features, thereby establishing a solid feature foundation for subsequent processing stages. The output size is halved after each ConvBlock due to MaxPooling, while the number of channels increases to extract more features.

2. SCConvBlock

The SCConvBlock, positioned at the center of the DSViT architecture, plays a crucial role in refining and optimizing the features extracted by the previous ConvBlocks. The SCConvBlock consists of two primary units: SRU (Spatial Reconstruction Unit) and CRU (Channel Reconstruction Unit), arranged sequentially.

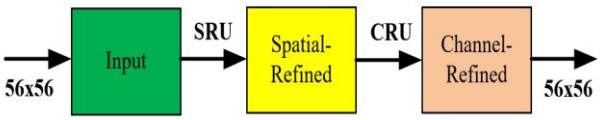

Figure 4. The architecture of SCConv integrated with Spatial Reconstruction Unit (SRU) and Channel Reconstruction Unit (CRU)

a. Spatial Reconstruction Unit

The SRU is designed to exploit spatial redundancy in features. This process involves two key steps: Separation and Reconstruction.

- ***Separation:*** The process begins by separating informative features from less informative ones based on spatial content. To assess the informativeness of different features, SRU utilizes expansion parameters in the Group Normalization layer. The intermediate feature X is normalized, and the γ parameters are used to measure the variation in spatial points.

$$GN(X) = \gamma \frac{X-\mu}{\sqrt{\sigma^{2}+\epsilon}} + \beta \quad (5)$$

where μ and σ are the mean and standard deviation, ϵ is a small constant to ensure stability during division, and γ and β are learnable parameters. The normalized correlation weights W_{γ} are computed as follows:

$$W_{\gamma} = \left\{ w_{i,j} = \frac{\gamma_i}{\sum_{j=1}^{C}\gamma_j} \right\}, \qquad i,j = 1,2,....,C \quad (6)$$

These weights are then normalized using the sigmoid function and thresholded to separate informative features W_1 from less informative features W_2:

$$W = \text{Gate}(\text{Sigmoid}(W_{\gamma} \cdot GN(X))) \quad (7)$$

Finally, the input features X are multiplied by W_1 and W_2, creating two weighted features: informative feature X_1^w and less informative feature X_2^w.

Reconstruction: After separation, SRU reconstructs the features by combining the

informative features with less informative ones to enhance information flow and reduce spatial redundancy.

$$X^w = X_{11}^w \oplus X_{22}^w \cup X_{21}^w \oplus X_{12}^w \quad (8)$$

Here, X_{11}^w and X_{22}^w represent the processed spatial information, while X_{21}^w and X_{12}^w are the reconstructed features that strengthen the relationship between informative and less informative features.

b. Channel Reconstruction Unit

The CRU is designed to exploit channel redundancy in features through a process of Separation, Transformation and Fusion.

- Separation: The spatially refined features Y are divided into two parts: the upper part Y_{up} and the lower part Y_{low}, with a split ratio α. After splitting, these features are compressed using 1x1 convolutions to increase computational efficiency.

- Transformation: Y_{up} is fed into the upper transformation stage, acting as a rich feature extractor. The GWC (Grouped Weight Convolution) and PWC (Point-wise Convolution) operations are applied to extract high-level representative information and reduce computational costs.

$$Y_1 = \text{GWC}(Y_{\text{up}}) + \text{PWC1}(Y_{\text{up}}) \quad (9)$$

Y_{low} is fed into the lower transformation stage, where PWC operations are applied to create feature maps with shallow hidden details. The result of this stage is:

$$Y_2 = \text{PWC2}(Y_{\text{low}}) \oplus Y_{\text{low}} \quad (10)$$

- Fusion: First, global average pooling is applied to gather global spatial information with channel-wise statistics:

$$G = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} Y[i, j] \quad (11)$$

Next, the global channel features of the upper and lower parts are overlaid and a channel attention function is used to create an important feature vector Z:

$$Z = \text{Attention}(G_{\text{up}}, G_{\text{low}}) \quad (12)$$

Finally, the refined channel features Y are fused by channel-wise addition:

$$Y_{\text{final}} = Y_1 + Y_2 \quad (13)$$

The SCConvBlock not only reduces spatial and channel redundancy in the intermediate features but also enhances feature representation, making the DSViT model more robust in recognizing subtle deepfake characteristics in videos.

3. Three Final ConvBlocks

The next three ConvBlocks in the DSViT architecture are designed to further extract complex features after the data has been refined through the SCConvBlock. The third ConvBlock increases the number of channels from 64 to 128, continuing to extract and normalize intermediate features while reducing the spatial dimensions of the image. The fourth and fifth ConvBlocks extend the number of channels from 128 to 256 and 512, optimizing the model's representational power before transitioning to the Vision Transformer stage.

B. Vision Transformer

The Vision Transformer component in the DSViT model is designed to learn complex spatial relationships between different parts of the feature maps. This component comprises three parts: Patch Embedding, Transformer Encoder, and MLP Head.

1. Patch Embedding

To utilize spatial features within the Transformer architecture, the input image is divided into small patches and then converted into embedding vectors. This process involves main steps:

Dividing the image into patches: The input image is divided into patches of size 7×7, forming a sequence of patches.

Linear embedding: Each patch x_p is then transformed into a linear embedding vector y_p through a linear layer:

$$y_p = \text{Linear}(x_p) \tag{14}$$

Positional Encoding: To retain the positional information of the patches within the image, the embedding vectors y_p are added to positional encodings:

$$z_p = y_p + \text{PE}(p) \tag{15}$$

Here, $\text{PE}(p)$ is the positional encoding vector corresponding to patch p. This step allows the model to understand the relative position of each patch in the entire image, helping it learn spatial relationships between them.

2. Transformer Encoder

The Transformer Encoder is the core of the Vision Transformer, responsible for learning spatial and temporal relationships between patches. The Transformer Encoder consists of two main components: Multi-Head Self Attention and Feed-Forward Network.

Multi-Head Self Attention: Multi-Head Self Attention is a mechanism that enables the model to learn relationships between different patches in the image. The Self-Attention calculation is defined as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{16}$$

where $Q = W_Q \cdot X$, $K = W_K \cdot X$, and $V = W_V \cdot X$. W_Q, W_K, and W_V are trainable weight matrices.

To learn more complex relationships, multiple self-attention heads are used in parallel, and their outputs are concatenated:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(h_1,h_2,\ldots,h_n)W^O \tag{17}$$

where h_i represents the output of each Self-Attention head, and W^O is the weight matrix of the output layer.

The output of the Multi-Head Attention is added to the original input via a Residual Connection to retain the original information and prevent information loss:

$$z' = \text{LayerNorm}(z + \text{MultiHead}(Q,K,V)) \tag{18}$$

This output is then normalized using Layer Normalization to maintain stability during training.

Feed-Forward Network: After the Self-Attention process, the features are processed through a Feed-Forward Network (FFN). The FFN in the Transformer Encoder comprises two linear layers with a non-linear activation function in between, allowing the model to learn more complex, non-linear features.

$$y = \text{GELU}(\text{Linear}_1(x)) \tag{19}$$

Here, GELU is the Gaussian Error Linear Unit, a non-linear activation function that enhances the model's non-linear capabilities.

The output is then passed through a second linear layer to restore the feature dimensions:

$$\text{FFN}(x) = \text{Linear}_2(y) \tag{20}$$

As in the Multi-Head Self Attention module, a Residual Connection is used to add the FFN output to the original input, followed by Layer Normalization for training stability:

$$z'' = \text{LayerNorm}(z' + \text{FFN}(z')) \tag{21}$$

3. MLP Head

After passing through the Transformer Encoder, the representative vector for the entire image is fed into the MLP Head for final classification. The MLP Head in the Vision Transformer functions similarly to a fully connected layer in traditional neural networks, with two linear layers and a ReLU activation function between them.

$$h = \text{ReLU}(\text{Linear}_3(z'')) \tag{22}$$

The second linear layer is used to classify the output into target classes, in this case, determining whether the image is real or fake:

$$\mathrm{MLP}(x) = \mathrm{Linear}_4(h) \quad (23)$$

Finally, the output is passed through a Softmax function to convert the values into probabilities, facilitating classification:

$$\mathrm{Softmax}(x) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (24)$$

The final output from the MLP Head indicates the probability of the frames being real or fake, allowing the DSViT model to make a decision about the frames's authenticity.

IV. Experiments and Evaluation

1. Data Preprocessing

In this study, we used the DFDC dataset to evaluate the data preprocessing method and performance of our model. DFDC is a dataset for deepface detection consisting of more than 100,000 videos [14]. Due to resource constraints, we focused on the first six segments out of a total of 50 segments within this dataset for our experiments. Our test dataset consists of 3.694 videos in mp4 format. We conducted a preprocessing step to extract 10 frames from each video, followed by utilizing BlazerFace to detect faces with a confidence threshold of 0.75. Each image is in JPEG format with a resolution of 224x224 pixels. Specifically, the datasets are organized as follows:

- Training set: Contains 25.550 frames (3.060 real images and 22.490 fake images).

- Validation set: Contains 5.651 frames 673 real images and 5.014 fake images).

- Testing set: Contains 5.662 frames (645 real images and 5.017 fake images).

The model was trained on the training and validation datasets and then tested on the test dataset to evaluate model's performance with new data. Determining whether a video is real or fake is based on the average of the results for each frame in the video.

2. Methodology

We conducted two experiments with the DSViT and CViT models on a computer equipped with an Intel Core i5-9600k CPU @3.7GHz, 16GB RAM, and an NVIDIA GeForce RTX 2060 GPU with 6GB memory. Both models were trained for 20 epochs using the Adam optimizer with a learning rate of 10^{-4}, weight decay of 10^{-7}, and a batch size of 32. Cross-entropy loss was employed to optimize classification performance. Data augmentation techniques such as random cropping, flipping, and normalization were applied to enhance the model's robustness and fault tolerance.

To determine the classification accuracy of the models, we used the LogLoss loss function. This function transforms network outputs into a probability distribution ranging from 0 to 1, where $0 < y < 0.5$ represents the real class, and $0.5 \leq y < 1$ represents the fake class. We selected the LogLoss metric to save the model, as it heavily penalizes random predictions and incorrect predictions with high confidence.

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \quad (25)$$

Additionally, we employed metrics such as Accuracy, ROC-AUC, Precision, Recall, F1 Score, FPR-FNR, and the Confusion Matrix to further evaluate the model. These metrics provide a comprehensive assessment of the model's performance across different aspects, from overall accuracy to the ability to accurately detect true positive and true negative samples.

3. Experimental Results

After evaluating the model, we obtained the following results (Figures 5, 6, 7 and Tables 1, 2):

Table 1. Experimental results

Model	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*LogLoss*
CViT	0.9262	0.8290	0.4434	0.5778	0.2168
DSViT	0.9415	0.8204	0.6233	0.7084	0.1511

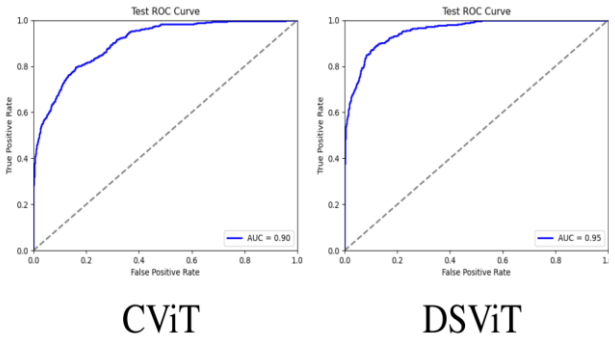

Figure 5. Comparison of ROC-AUC Results

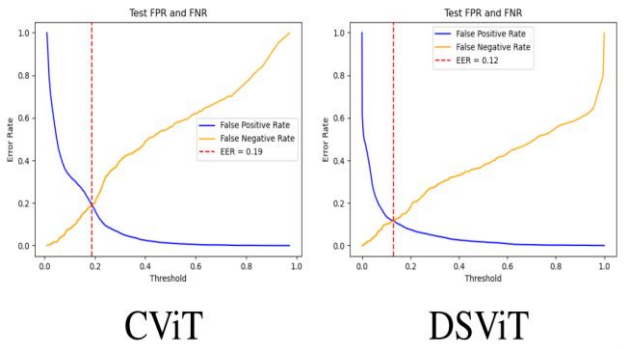

Figure 6. Comparison of FPR-FNR Results

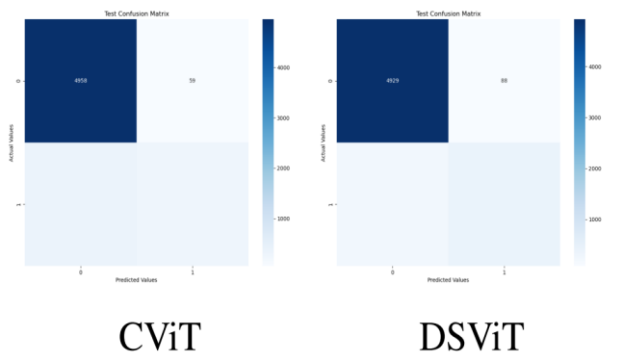

Figure 7. Comparison of Confusion Matrix Results

From the results, it can be observed that the DSViT model outperforms the CViT model on most evaluation metrics. Specifically, the DSViT model has only nearly 79 million parameters, nearly 9 million fewer than the CViT model, but achieves Log Loss of 0.1511, Accuracy of 0.9415, Precision of 0.8204, Recall of 0.6233, and F1 Score of 0.7084, which are superior to the corresponding metrics of the CViT model. The results from the ROC, FPR-FNR, and Confusion Matrix charts also show that the DSViT model is better at distinguishing between real and fake videos, while minimizing misclassification errors.

TABLE 2. SUMMARY OF DSVIT FEATURES LEARNING

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 224, 224]	896
BatchNorm2d-2	[-1, 32, 224, 224]	64
Mish-3	[-1, 32, 224, 224]	0
MaxPool2d-4	[-1, 32, 224, 224]	0
ConvBlock-5	[-1, 32, 224, 224]	0
Conv2d-6	[-1, 64, 112, 112]	18,496
BatchNorm2d-7	[-1, 64, 112, 112]	128
Mish-8	[-1, 64, 112, 112]	0
MaxPool2d-9	[-1, 64, 56, 56]	0
ConvBlock-10	[-1, 64, 56, 56]	0
GroupNorm-11	[-1, 64, 56, 56]	128
Sigmoid-12	[-1, 64, 56, 56]	0
SRU-13	[-1, 64, 56, 56]	0
Conv2d-14	[-1, 16, 56, 56]	512
Conv2d-15	[-1, 16, 56, 56]	512
Conv2d-16	[-1, 64, 56, 56]	4,672
Conv2d-17	[-1, 64, 56, 56]	1,024
Conv2d-18	[-1, 48, 56, 56]	768
AdaptiveAvgPool2d-19	[-1, 128, 1, 1]	0
CRU-20	[-1, 64, 56, 56]	0
SCConv-21	[-1, 64, 56, 56]	0
Conv2d-22	[-1, 128, 56, 56]	73,856
BatchNorm2d-23	[-1, 128, 56, 56]	256
Mish-24	[-1, 128, 56, 56]	0
MaxPool2d-25	[-1, 128, 28, 28]	0
ConvBlock-26	[-1, 128, 28, 28]	0
Conv2d-27	[-1, 256, 28, 28]	295,168
BatchNorm2d-28	[-1, 256, 28, 28]	512
Mish-29	[-1, 256, 28, 28]	0
MaxPool2d-30	[-1, 256, 14, 14]	0
ConvBlock-31	[-1, 256, 14, 14]	0
Conv2d-32	[-1, 512, 14, 14]	1,180,160
BatchNorm2d-33	[-1, 512, 14, 14]	1,024
Mish-34	[-1, 512, 14, 14]	0
MaxPool2d-35	[-1, 512, 7, 7]	0

IV. CONCLUSION

In this study, we proposed evaluated a new model, DSViT, specifically designed for detecting deepfakes in videos. The DSViT model builds upon the foundational CViT model, incorporating several enhancements aimed at reducing feature redundancy and strengthening the model's ability to represent diverse data effectively.

The experimental outcomes indicate that the DSViT model has achieved notable advancements compared to the CViT model. These results underscore the value of integrating components such as SCConv and refining the Convolutional blocks, which together allow the model to more accurately identify local features and capture intricate spatial relationships among features.

REFERENCES

[1] F. Abbas and A. Taeihagh, "Unmasking deepfakes: A systematic review of deepfake detection and generation techniques using artificial intelligence," Expert Systems With Applications, 2024: 124260.

[2] A. Naitali, M. Ridouani, F. Salahdine, M. Kaabouch, "Deepfake attacks: Generation, detection, datasets, challenges, and research directions," Computers, vol. 12, no. 10, pp. 216, Oct 2023.

[3] X. Li, H. Zhou, and M. Zhao, "Transformer-based cascade networks with spatial and channel reconstruction convolution for deepfake detection," Mathematical Biosciences and Engineering, vol. 21, no. 3, pp. 4142-4164, 2024.

[4] D. Wodajo & S. Atnafu, "Deepfake Video Detection Using Convolutional Vision Transformer", arXiv preprint arXiv:2102.11126, 2021.

[5] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul 2017.

[6] H. H. Nguyen, N. T. Tieu & I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos", ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2307-2311, May 2019.

[7] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network," arXiv:1809.00888, Sep 2018.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in International Conference on Machine Learning (ICML), Jun 2019.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in International Conference on Learning Representations (ICLR), May 2021.

[10] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić & C. Schmid, "ViViT: A Video Vision Transformer", Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6816-6826, 2021.

[11] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in International Conference on Computer Vision (ICCV), Oct 2021.

[12] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Feb 2022.

[13] J. Li, Y. Wen, and L. He, "SCConv: Spatial and channel reconstruction convolution for feature redundancy," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[14] Kagge (2020), Deepfake Detection Challenge. Accessed September 10, 2024, from: https://www.kaggle.com/c/deepfake-detection-challenge/data.

ABOUT THE AUTHORS

Pham Minh Thuan

Workplace: Center for Information Technology and Cyber Security Monitoring, Vietnam Government Information Security Commission.

Email: pmthuan@bcy.gov.vn

Education: Bachelor's degree in Information Security at Academy of Cryptography Techniques - Vietnam in 2009, Master's degree in Cryptographic Techniques at Academy of Cryptography Techniques, Vietnam in 2013.

Recent research interests: Information Security, Machine learning, Deepfake Detection.

Tên tác giả: **Phạm Minh Thuấn**

Đơn vị công tác: Trung tâm Công nghệ thông tin và Giám sát an ninh mạng, Ban Cơ yếu Chính phủ, Việt Nam

Email: pmthuan@bcy.gov.vn

Quá trình đào tạo: Kỹ sư An toàn thông tin tại Học viện Kỹ thuật mật mã - Việt Nam năm 2009, Thạc sỹ Kỹ thuật Mật mã tại Học viện Kỹ thuật mật mã - Việt Nam năm 2013.

Hướng nghiên cứu hiện nay: An toàn thông tin, học máy, phát hiện Deepfake.

Bui Thu Lam

Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission.

Email: lambt@actvn.edu.vn

Education: Bachelor's degree in Information Technology at Military Technology Academy - Vietnam in 1998, Master's degree in Information Technology at The University of New South Wales - Australia in 2002, Doctor of Philosophy in Computer Science at The University of New South Wales - Australia in 2007, Assoc.Prof in Information Technology in 2012.

Recent research interests: Artificial intelligence; Machine learning; Evolutionary computation.

Tên tác giả: **Bùi Thu Lâm**

Đơn vị công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: lambt@actvn.edu.vn

Quá trình đào tạo: Kỹ sư Công nghệ thông tin tại Học viện Kỹ thuật Quân sự - Việt Nam năm 1998, Thạc sỹ Công nghệ thông tin tại Đại học tổng hợp New South Wales - Úc năm 2002, Tiến sĩ Khoa học máy tính tại Đại học tổng hợp New South Wales - Úc năm 2007, Phó giáo sư Công nghệ thông tin năm 2012.

Hướng nghiên cứu hiện nay: Trí tuệ nhân tạo, học máy, tính toán tiến hóa.

Pham Duy Trung

Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission.

Email: trungpd@actvn.edu.vn

Education: Bachelor's degree in Information Technology at Hanoi University of Science and Technology - Vietnam in 2005, Master's degree in Information Technology and Systems at University of Canberra - Australia in 2012, Doctor of Philosophy in Information Sciences and Engineering at University of Canberra - Australia in 2018.

Recent research interests: Privacy, Machine learning, Safe Machine learning.

Tên tác giả: **Phạm Duy Trung**

Đơn vị công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: trungpd@actvn.edu.vn

Quá trình đào tạo: Kỹ sư Công nghệ thông tin tại Đại học Bách khoa Hà Nội - Việt Nam năm 2005, Thạc sĩ Công nghệ thông tin và Hệ thống thông tin tại Đại học tổng hợp Canberra - Úc năm 2012, Tiến sĩ Khoa học và Kỹ thuật thông tin tại Đại học tổng hợp Canberra - Úc năm 2018.

Hướng nghiên cứu hiện nay: Tính riêng tư, học máy, an toàn trong học máy.