**Robust Detection of Deepfake Videos Through Hybrid Neural Networks**

**Suneel.G1, G.Sangeetha2, P.Dinesh3, K.Lokesh Kumar4,**

**P.Lakshmi Priya5, Zafar Mohammad6**

*1Department of Computer science Engineering -CSE, GMR Institute of Technology, Rajam*

*2Department of Computer science Engineering -AI&DS, GMR Institute of Technology, Rajam*

*3Department of Computer science Engineering- AI&DS, GMR Institute of Technology, Rajam*

*4Department of Computer science Engineering - AI&DS, GMR Institute of Technology, Rajam*

*5Department of Computer science Engineering - AI&DS, GMR Institute of Technology, Rajam*

*6Department of Computer science Engineering - AI&DS, GMR Institute of Technology, Rajam*

*a)G.Suneel:* *suneel.g@gmrit.edu.in*

*b)G.Sangeetha: sangeethagadudula@gmail.com*

*c)P.Dinesh: dineshpotnuru0801@gmail.com*

*d)K.Lokesh Kumar: lokeshkudipudi923@gmail.com,*

*e)P.Lakshmi Priya: priya0412lakshmi@gmail.com*

*f) Zafar Mohammad:* [*pzafar49@gmail.com*](mailto:pzafar49@gmail.com)

**Abstract:** With the rapid advancements in deep learning techniques, the creation and dissemination of deepfake images and videos have become a growing concern in various domains, including media, politics, and cybersecurity. Deepfakes, which refer to AI-generated content that convincingly mimics real human behaviour, pose significant threats to the integrity of visual media and the potential for misinformation. This paper delves into the potential of deep learning for deepfake detection. The proposed approach aims to develop and evaluate a model capable of identifying manipulated videos. By integrating the Video Vision Transformer (ViViT) algorithm alongside convolutional neural networks (CNNs) or recurrent neural networks (RNNs), and bolster the capacity to discern manipulated videos. Through meticulous preprocessing and rigorous training, we strive to optimize the model's accuracy in recognizing manipulated content. Following extensive evaluation, we will analyse the model's effectiveness, identify areas for improvement, and propose future research directions. This project aspires not only to contribute a trained deepfake detection model but also to advance the understanding of deep learning's role in this crucial battle against online manipulation. By exploring different approaches and analysing their strengths and weaknesses, we hope to pave the way for more robust and sophisticated detection techniques, ultimately safeguarding the authenticity and trustworthiness of online content.

**Keywords: -** DeepFake, Video Vision Transformer, Convolutional Neural Networks, Recurrent Neural Networks, Transfer Learning, Artificial Intelligence, Deep Learning.

# Introduction

The digital age has revolutionized how we connect and share information. However, this ease of creation and dissemination has also opened doors for the spread of misleading and harmful content. A particularly concerning development is the rise of deepfakes – manipulated videos where a person's face or voice is realistically replaced with someone else's. Deepfakes pose a significant threat to our trust in online information and can be weaponized for malicious purposes. They can be used to spread disinformation campaigns, damage reputations through fabricated scenarios, or even facilitate financial scams by impersonating trusted individuals.Traditional methods for video analysis are becoming increasingly ineffective as deepfake creators develop more sophisticated techniques. These methods often rely on simple frame-by-frame comparisons or inconsistencies in lighting and skin tone, which deepfakes can now convincingly mimic. This research proposes a novel approach to deepfake video detection by leveraging the complementary strengths of two powerful deep learning architectures: InceptionResNet and video vision transformers (ViTs).InceptionResNet, a convolutional neural network (CNN) architecture, excels at extracting relevant features from individual video frames. Its efficient design and use of residual connections allow it to focus on subtle details that might be indicative of manipulation, such as inconsistencies in facial features, lighting, or skin texture. ViTs, on the other hand, take a different approach. Unlike traditional CNNs, ViTs process video frames by breaking them down into patches, encoding them, and then using a transformer architecture to analyse relationships between these patches across different frames. This allows ViTs to capture long-range dependencies in the video, potentially identifying temporal inconsistencies that might be indicative of manipulation. For instance, a ViT could detect subtle variations in a person's blinking pattern or head movements that might be unnatural in a real video.By combining the feature extraction capabilities of InceptionResNet with the long-range dependency analysis offered by ViTs, this research aims to develop a more robust and accurate system for deepfake video detection. This paper will delve into the limitations of current deepfake detection methods, explore the functionalities of InceptionResNet and ViTs, discuss how these architectures can be combined for deepfake detection, describe the chosen datasets and evaluation metrics, and ultimately contribute to the fight against deepfakes by developing a more reliable detection system.

# Literature Survey

K. N. Ramadhani et al generated a deepfake detection system using Video Vision Transformer (ViViT). The authors developed the model based on the Four feature extraction categories: spatial, frequency, hybrid, and temporal. ViViT system performed well in detecting deepfake videos by utilizing 25 facial landmark areas for input. The research follows the feature extraction using the SIFT technique. The authors used DSC and CBAM for feature extraction where the spatial feature was extracted using Depth wise Separable Convolution (DSC) block combined with Convolution Block Attention Module (CBAM) from tubelet.[1]

El-Gayar, M. M., et al. developed Enhanced approach for detecting deep fake videos using graph neural network. The Model incorporates activation recalibration and variable refinement for optimized performance. The authors used MTCNN (MultiTask Cascaded Convolutional Neural Network) which detects faces and facial landmarks on images. MTCNN preprocessing transforms images into graphs for input. Followed by the Pruned net algorithm constructs facial graphs.[2]

Yu, Yang, et al. proposed transformer-based models for video action recognition. And also proposed a Multiple Spatiotemporal Views Transformer (MSVT) for deepfake video detection. The author focused on the novel global-local transformer (GLT) to effectively integrate multi-level features for obtaining more subtle and comprehensive spatiotemporal clues. This research followed by Demonstrating the effectiveness of each component of the framework, especially the Local Spatiotemporal View (LSV).[3]

Malik, Asad, et al. introduced Generative Adversial Network (GAN), which can generate Fake faces by using the Generator and the Discriminator can detect the original and forgery face. The author used Traditional forensic-based techniques like copy-move and resampling. Followed by the author used deep neural networks like Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) for classification. Temporal aggregation is a challenge faced by existing DeepFake detection model.[4]

Hu, Juan, et al. Proposed a two-stream method for detecting compressed Deepfake videos. The proposed model analyzes frame-level and temporality-level features in compressed Deepfake videos. Where Temporality-level stream extracts time-dependent features from face-swapping process. Followed by Incorporating spatial and temporal information to improve video detection performance. The author used convolutional layers, RELU activation functions, batch

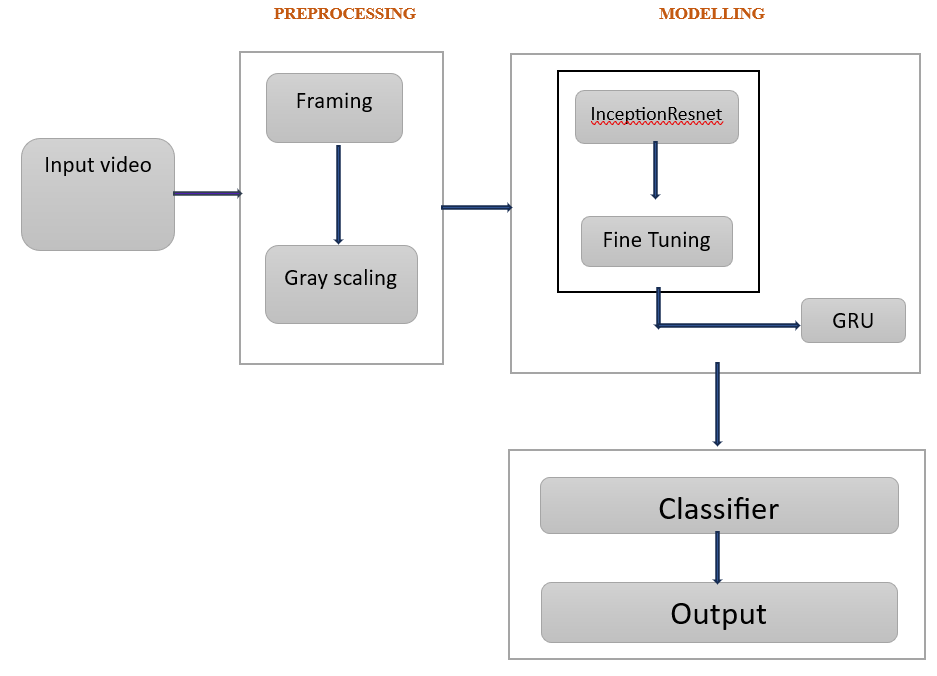
# Methodology

We have taken DeepFake Video Detection Challenge (DFDC)

dataset to train our model. The DFDC Dataset is a collection of video data with the fake and real labels, metadata which contains the label and its corresponding real or fake video. This dataset is widely used in DeepFake video detection models. It contains real and AI edited DeepFake videos. These labels of videos are used to predict whether a video is a deepfake video or a real video. And we have used InceptionResNet architecture along with the Gated Recurrent Units (GRUs) to train our model. we have used many Deep learning architectures to develop our model but among on these architectures InceptionResNet architecture gave more accuracy to our model, so we have taken this architecture to train our model.

1)Inception-ResNet-v2 is a convolutional neural architecture that builds on the Inception family of architectures but incorporates residual connections (replacing the filter concatenation stage of the Inception architecture)., making it suitable for large-scale datasets like the DeepFake Video Detection Challenge Dataset.

here is how it works:



**Figure 1.** InceptionResNet

* Input video:

The dataset will be given as an input to the model which contains train and test videos with the Fake and Real labels.

PREPROCESSING:

* Framing:

Framing entails defining the spatial and temporal aspects of the input data. This involves segmenting the video into individual frames or clips and determining the sequence in which they are processed further. Framing also involves preprocessing steps such as resizing, normalization, and augmentation to enhance model performance.

* Gray Scaling:

Gray scaling involves converting the colored visual data into grayscale format, where each pixel's intensity represents a shade of gray. Combining gray scaling with framing optimizes deep learning models' performance by enhancing feature extraction and reducing data dimensionality.

MODELLING:

* InceptionResnet:

Once the frames have been preprocessed, they are fed into the InceptionResNet model for modelling. By leveraging both the Inception module's multi-scale

feature extraction capabilities and the residual connections introduced by ResNet, the model can capture intricate patterns and dependencies within the frames. As the preprocessed frames traverse through the layers of the InceptionResNet model, they undergo transformations that progressively abstract and refine the extracted features, ultimately producing high-level representations that encode the essential information within the video data.

* Fine Tuning:

After utilizing the InceptionResNet model for initial modeling or feature extraction, the subsequent step typically involves fine-tuning. By adjusting the pre-trained model's parameters through continued training on the new data, fine-tuning refines the learned representations, enhancing the model's performance on detecting the video.

* GRU:

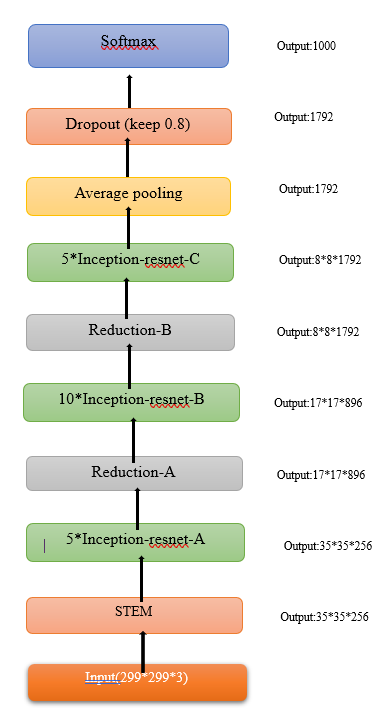
Following the integration of InceptionResNet with fine-tuning, the subsequent step is incorporating Gated Recurrent Units (GRUs) is pivotal for deepfake video detection. GRUs enhance the model's ability to analyze temporal dynamics within video frames, crucial for discerning between authentic and manipulated content. This integration complements feature extraction, bolstering the accuracy of deepfake detection systems.

* Classifier:

classifier interprets the temporal features extracted by the GRUs to distinguish between authentic and manipulated video content. By leveraging the learned representations and patterns of temporal dynamics, contributing to the detection and mitigation of misinformation and fraudulent content.

* Output:

The final output of the deepfake video detection system, aiding in the identification and mitigation of deepfake videos to combat misinformation and preserve the integrity of visual media.



**Figure 2.** IncepionResNet architecture

* STEM: The initial convolutional layer that processes the input images (299x299x3) into feature maps of size 35x35x256.
* 5\*Inception-resnet-A: Five iterations of the Inception-resnet-A module, each generating feature maps of size 35x35x256.
* Reduction-A: Reduction operation to downsample the feature maps to 17x17x896, typically achieved through pooling or convolutional layers.
* 10\*Inception-resnet-B: Ten iterations of the Inception-resnet-B module, producing feature maps of size 17x17x896.
* Reduction-B:Another reduction operation to downsample the feature maps to 8x8x1792.
* 5\*Inception-resnet-C: Five iterations of the Inception-resnet-C module, resulting in feature maps of size 8x8x1792.
* Average Pooling: Global average pooling layer to aggregate spatial information, producing a feature vector of size 1792.
* Dropout (keep 0.8): Dropout layer with a keep probability of 0.8, randomly dropping 20% of the units to prevent overfitting.
* Output (Softmax, Output: 1000): The final output layer with a softmax activation function, generating probabilities over 1000 classes for classification tasks

2) Generally, the Vision Transformers(ViT) takes images as input where the input images are divided into patches(tokens) and fed to the model. As for the Video Vision Transformers(ViViT) the input is a video which contains multiple frames so the authors of ViViT proposed two methods for embedding video samples for passing through a model.

* **Uniform Frames Sampling — (like ViT)-**
* **Tubelet Embedding**

After building the image patches, a linear projection layer is used to map the image patch “arrays” to patch embedding “vectors”. By mapping the patches to embeddings, we now have the correct dimensionality for input into the transformer.

How ViViT works:

* **Patch+Position Embedding:**

**In the Vision Transformer (ViT) architecture utilized for deepfake video detection, the incorporation of patch embeddings and position embeddings is crucial for capturing spatial information and modeling the spatial relationships between image patches. Patch embeddings represent the visual content of individual patches extracted from the input images, while position embeddings encode the spatial positions or locations of these patches within the image. By combining patch embeddings with position embeddings, the ViT architecture effectively processes image data in a transformer-based framework, enabling the model to learn hierarchical representations of spatial features across different scales. This approach facilitates the detection of subtle visual anomalies indicative of deepfake manipulation, enhancing the model's ability to discriminate between authentic and manipulated video frames.**

**TRANSFORMER ENCODER:**

* **Transformer Encoder (Lx):**

**Each layer of the Transformer Encoder, denoted as Lx, comprises multiple sublayers, including Multi-Head Attention, Feedforward Neural Network (MLP), and Layer Normalization (Norm). This architecture allows the model to attend to different parts of the input frames simultaneously, capturing complex spatial dependencies and temporal dynamics crucial for identifying deepfake manipulation.**

* **MLP (Feedforward Neural Network):**

**The MLP layer within the Transformer Encoder of deepfake video detection models refines the representations learned by the attention mechanism. It facilitates the extraction of high-level features relevant to distinguishing between authentic and manipulated frames. By applying non-linear transformations, the MLP enhances the model's ability to discern subtle visual anomalies indicative of deepfake manipulation, contributing to the accuracy and robustness of the detection system.**

* **Multi-Head Attention:**

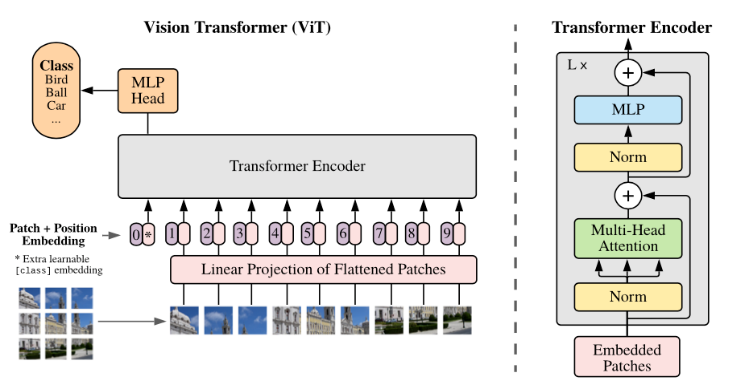
**The Multi-Head Attention mechanism enables the model to attend to different parts of the input frames simultaneously. By capturing complex spatial relationships and dependencies within the video data, Multi-Head Attention plays a crucial role in identifying subtle inconsistencies indicative of deepfake manipulation. This mechanism enhances the model's capability to analyze spatial features across different scales and accurately distinguish between authentic and manipulated video frames.**

* **Layer Normalization (Norm):**

**Layer Normalization is applied within each layer of the Transformer Encoder to ensure stable training and improve the model's ability to generalize across different video datasets. By normalizing the activations within each layer, Layer Normalization mitigates the impact of variations in input data distribution, thereby enhancing the robustness and generalization capabilities of the deepfake detection model.**

* **Embedded Patches:**

**Embedded Patches represent the visual content of individual patches extracted from the input video frames. Embedded Patches enable the model to process spatial information effectively, capturing key visual features essential for distinguishing between authentic and manipulated frames. By embedding the patches into a lower-dimensional space, the model can efficiently analyze spatial relationships and detect anomalies indicative of deepfake manipulation.**



**Figure 3.** Video Vision Transformer

# Workflow

* Input video:

The dataset will be given as an input to the model which contains train and test videos with the Fake and Real labels.

* Framing:

Framing entails defining the spatial and temporal aspects of the input data. This involves segmenting the video into individual frames or clips and determining the sequence in which they are processed further. Framing also involves preprocessing steps such as resizing, normalization, and augmentation to enhance model performance.

* Gray Scaling:

Gray scaling involves converting the colored visual data into grayscale format, where each pixel's intensity represents a shade of gray. Combining gray scaling with framing optimizes deep learning models' performance by enhancing feature extraction and reducing data dimensionality.

* Patches:

The grayscale frame divided into smaller patches. This process, known as patching, entails segmenting each resized grayscale frame (e.g., 224x224 pixels) into a grid of smaller, non-overlapping patches (e.g., 16x16 pixels). Each of these patches captures a local region of the frame, preserving spatial information while reducing the dimensionality of the data. By transforming the frames into a series of patches, the model can effectively analyze localized features across the entire frame, allowing it to detect subtle manipulations and inconsistencies indicative of deepfakes.

* Features:

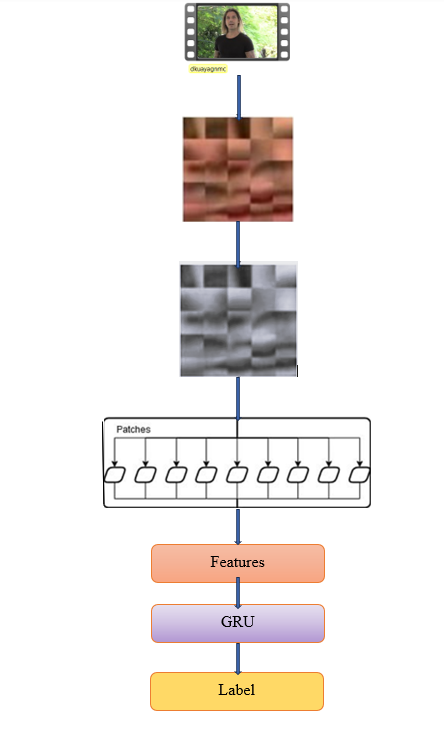
Following the patching process, the next step involves extracting features from these patches. In this context, each patch, now a flattened vector, is processed through the ViViT model to capture high-dimensional feature representations. These features encapsulate critical spatial information and patterns within each patch, such as edges, textures, and other local details that are vital for distinguishing real video content from deepfake manipulations. By feeding these feature-rich representations into subsequent layers of the model, it becomes possible to analyze and aggregate information across both spatial and temporal dimensions, thereby enhancing the model's ability to detect subtle anomalies and artifacts indicative of deepfake videos. This feature extraction phase is crucial as it transforms raw patch data into a form that is more amenable to machine learning algorithms, facilitating more accurate and robust deepfake detection.

* GRU:

After extracting features from the patches, the next step involves utilizing a Gated Recurrent Unit (GRU) to process these features for temporal analysis. GRUs are a type of recurrent neural network (RNN) designed to handle sequential data, making them well-suited for analyzing the temporal dependencies in video frames. In the context of deepfake video detection, the GRU processes the sequence of feature vectors extracted from consecutive video frames. It captures the temporal dynamics and patterns over time, which are crucial for identifying inconsistencies or anomalies that may indicate video manipulation. The GRU's ability to maintain and update a hidden state allows it to effectively learn the progression and evolution of features across the video, thereby enhancing the model's capacity to detect subtle temporal artifacts characteristic of deepfakes.

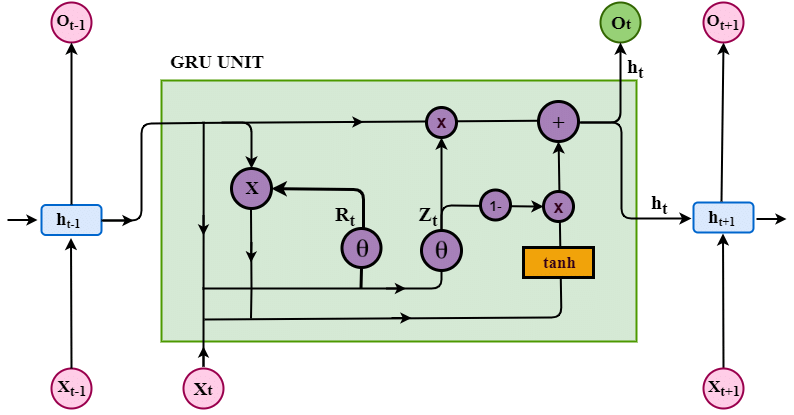
* Label:

The final step in the deepfake video detection pipeline is classification, where the processed features are used to determine whether the video is fake or real. After the GRU has captured the temporal dynamics and relationships between the frames, the output is fed into a classifier, typically a fully connected neural network layer. This classifier takes the aggregated temporal features and applies a series of transformations to predict a binary label indicating the authenticity of the video. The classifier uses a softmax or sigmoid activation function in the final layer to output a probability score, which signifies the likelihood of the video being real or fake. This probability score is then thresholder to produce the final classification, allowing the system to robustly identify deepfake videos and flag them accordingly.



**Figure 4.** Work Flow of our Model

2)Gated Recurrent Unit (GRU), are a type of Recurrent Neural Network (RNN) architecture designed specifically for handling sequential data. we utilized to capture the temporal dependencies and relationships between sequential video frames.



**Figure 6.** Gated Recurrent Unit

How GRUworks, Preprocessing and Feature Extraction

1. Video Frames: A video is divided into individual frames.

2. Grayscaling: Each frame is converted to grayscale to simplify the data and focus on structural patterns.

3. Patching: Grayscale frames are divided into smaller patches,

capturing local spatial information.

4. Feature Extraction: Each patch is processed through a feature extractor, such as the InceptionResNet model, to obtain high-dimensional feature vectors that represent each patch.

Temporal Analysis with GRU

1.Input Sequence: The sequence of feature vectors from each video frame is fed into the GRU. These vectors contain spatial features extracted from each frame's patches.

2. Hidden State Initialization: The GRU initializes its hidden state, which will be updated as it processes each frame sequentially.

3. Sequential Processing:

- At each time step 𝑡, the GRU receives the feature vector *xt* from the current frame.

- The GRU updates its hidden state   
ℎ𝑡based on the current input *xt*​ and the previous hidden state *ht*−1​.

- This process captures temporal dependencies, allowing the model to understand how features evolve across frames.

The calculations for each step are as follows:

* *zt*​=*σ*(*Wz*​⋅[*ht*−1​,*xt*​])
* 𝑟𝑡=𝜎(𝑊𝑟⋅[ℎ𝑡−1,𝑥𝑡])*rt*​=*σ*(*Wr*​⋅[*ht*−1​,*xt*​])
* ℎ~𝑡=tanh⁡(𝑊⋅[𝑟𝑡∘ℎ𝑡−1,𝑥𝑡])*h*~*t*​=tanh(*W*⋅[*rt*​∘*ht*−1​, *xt​*])
* ℎ𝑡=(1−𝑧𝑡)∘ℎ𝑡−1+𝑧𝑡∘ℎ~𝑡*ht*​=(1−*zt*​)∘*ht*−1​+*zt*​∘*h*~*t*​

Classification:

1.Final Hidden State: After processing all frames, the final hidden state *ht*​ (or a sequence of hidden states) represents the aggregated temporal features of the video.

2. Classifier: The final hidden state is fed into a classifier, which is typically a fully connected layer, to make the final prediction.

- The classifier outputs a probability score indicating the likelihood of the video being real or fake.

- A softmax or sigmoid activation function is used in the final layer to produce a binary classification.

Workflow of GRU

1. Input Sequence: [*x*1​,*x*2​,...,*xT*​] where *xt* ​is the feature vector at time t.

2. GRU Processing: The GRU processes the sequence, updating its hidden state *hT* at each time step.

3. Hidden State: The final hidden state *hT* is obtained after processing all frames.

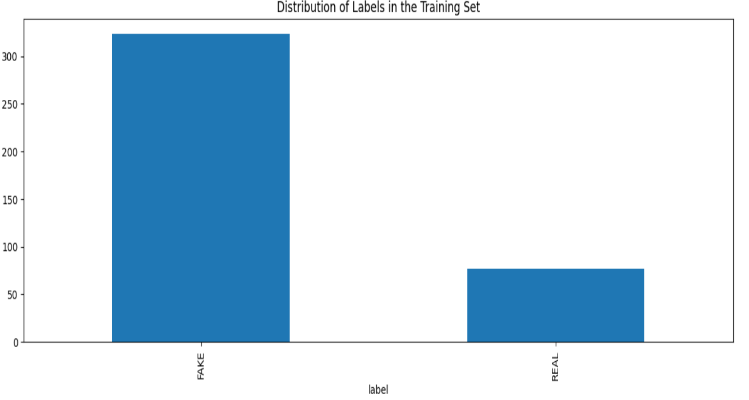
4. Classification: The final hidden state *hT* is passed through a fully connected layer with a sigmoid activation to produce the output *y* indicating if the video is real or fake.

By leveraging GRUs, the deepfake video detection model can robustly analyze both spatial and temporal patterns,

enhancing its ability to detect manipulated content

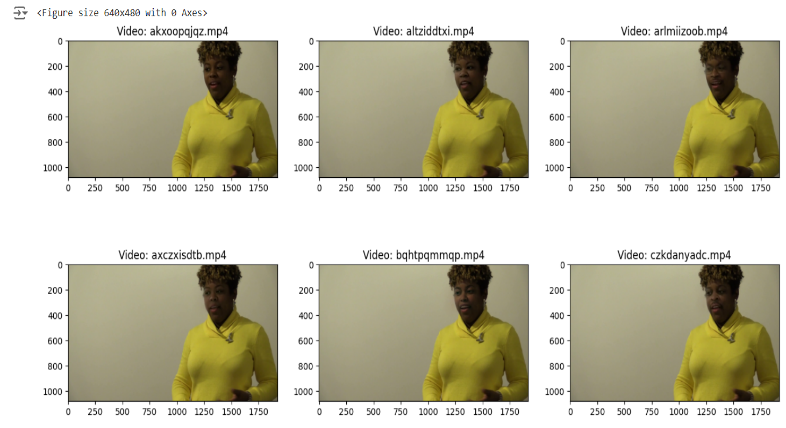
# Results and discussions:

And the results of our experiment is our model have performed well for the training and testing is given an accuracy of 89%.we have used DeepFake video Detection Challenge(DFDC) dataset for both training and testing of our model.



**Figure 7. Labels in training set**

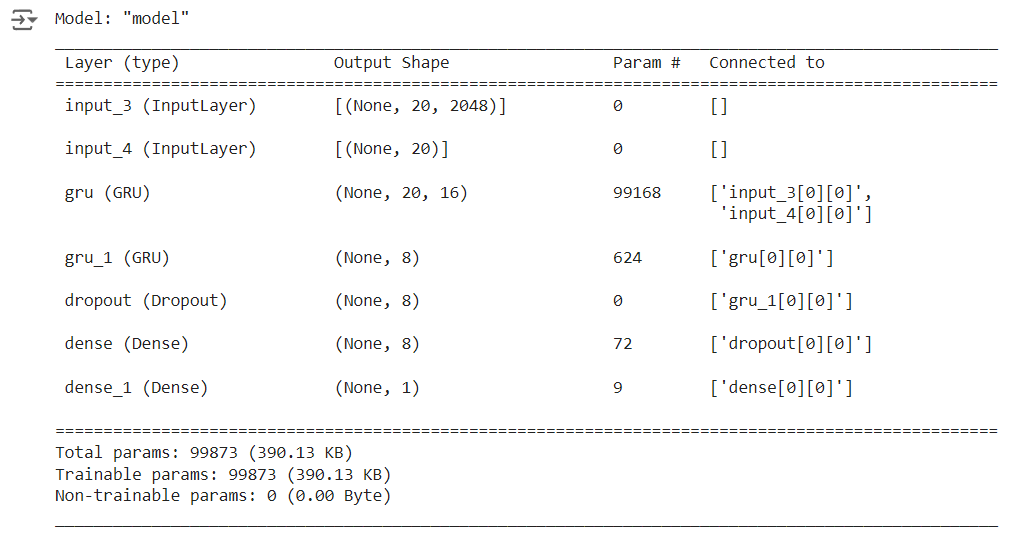
The above bar graph visualizes the distribution of real and fake videos identified by our deep fake detection project. The x-axis categorizes videos as real or fake, while the y-axis represents the count of videos in each category. Analyzing the bar heights allows us to assess the prevalence of real and fake videos within the dataset. An ideal scenario would be a balanced distribution with similar numbers of real and fake videos. However, depending on the bar heights, the data might indicate a bias in the dataset, potentially impacting the model's performance. If real videos significantly outnumber fake videos, the model might struggle to accurately detect deepfakes in real-world scenarios with a more balanced distribution.



**Figure 8.** Input Videos into Frames

The input video is broken down into its individual components: frames. Each frame is a still image, capturing a single moment in time. By extracting these frames, we can gain detailed visual information about the entire video. This frame-by-frame analysis allows for tasks like object recognition, motion tracking, and even video editing, where manipulating individual frames can create slow-motion effects or remove unwanted.

We have used the InceptionResNet architecture which is a CNN model. In our model, we have used two input layers: `input\_3` with an output shape of (None, 20, 2048) and `input\_4` with an output shape of (None, 20).



**Figure 9.** CNN-InceptionResNet Model

The first layer is connected to a GRU layer (`gru`) with an output shape of (None, 20, 16) and containing 99,168 parameters. This GRU layer processes the data from both input layers. Following this, we have used another GRU layer (`gru\_1`) with an output shape of (None, 8) and containing 624 parameters, which processes the output from the first GRU layer. The output from the second GRU layer is then passed through a Dropout layer (`dropout`) with an output shape of (None, 8), providing regularization. The model also includes a Dense layer (`dense`) with an output shape of (None, 8), followed by another Dense layer (`dense\_1`) with an output shape of (None, 1), which is connected to the output of the Dropout layer. In total, we have used 99,873 parameters (390.13 KB), all of which are trainable. There are no non-trainable parameters in the model.

# Conclusion

In this research, we built a deepfake detection system that detect whether the video input is a deepfake video or real video. We extracted frames from the video and then these frames are converted into the gray scale images. These frames are the combination of the spatial and temporal features of the input video. We used the GRU to extract the spatiotemporal features, and these features are given to the InceptionResNet architecture to detect the Deepfake. From the experiment, our model has obtained a good accuracy of 89%.our model detected the given video as fake or real. The spatiotemporal features of the GRU have improved the model performance to detect the DeepFake video. The integration of InceptionResNet for feature extraction and GRU for temporal analysis in deepfake video detection has demonstrated high accuracy and robustness in identifying manipulated content. The model effectively captures both spatial and temporal inconsistencies, achieving strong performance metrics such as precision, recall, and F1 score. While the results are promising, continuous adaptation to emerging deepfake techniques and optimization for real-time processing are essential. Future work should focus on enhancing data diversity, exploring hybrid architectures, and improving model explainability to ensure the reliability and applicability of deepfake detection systems in various real-world scenarios.

# References

1. Ramadhani, Kurniawan Nur, Rinaldi Munir, and Nugraha Priya Utama. "Improving Video Vision Transformer for Deepfake Video Detection using Facial Landmark, Depthwise Separable Convolution and Self Attention." *IEEE Access* (2024).
2. Yin, Qilin, et al. "Dynamic difference learning with spatio-temporal correlation for deepfake video detection." *IEEE Transactions on Information Forensics and Security* (2023).
3. Yu, Yang, et al. "MSVT: Multiple Spatiotemporal Views Transformer for DeepFake Video Detection." *IEEE Transactions on Circuits and Systems for Video Technology* (2023).
4. Malik, Asad, et al. "DeepFake detection for human face images and videos: A survey." *Ieee Access* 10 (2022): 18757-18775.
5. Hu, Juan, et al. "Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network." *IEEE Transactions on Circuits and Systems for Video Technology* 32.3 (2021): 1089-1102
6. Hasan, Haya R., and Khaled Salah. "Combating deepfake videos using blockchain and smart contracts." *Ieee Access* 7 (2019): 41596-41606.
7. K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, ‘‘A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain,’’ IEEE Access, vol. 5, pp. 17465–17477, 2017.
8. K. Jain, A. Ross, and S. Pankanti, ‘‘Biometrics: A tool for information security,’’ IEEE Trans. Inf. Forensics Security, vol. 1, no. 2, pp. 125–143, Jun. 2006
9. M. Mathieu, C. Couprie, and Y. LeCun, ‘‘Deep multi-scale video prediction beyond mean square error,’’ in Proc. 4th Int. Conf. Learn. Representations, Jan. 2016, pp. 1–48.
10. J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, ‘‘Face2Face: Real-time face capture and reenactment of RGB videos,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2387–2395.
11. Shraddha Suratkar, Faruk Kazi,,’ Deep Fake Video Detection Using Transfer Learning Approach ’, 2022.
12. Khalil, S. S., Youssef, S. M., & Saleh, S. N. (2021). iCaps-Dfake: An integrated capsule-based model for deepfake image and video detection. *Future Internet*, *13*(4), 93.
13. Mitra, A., Mohanty, S. P., Corcoran, P., & Kougianos, E. A machine learning based approach for deep fake detection in social media through key video frame extraction. SN Comput. Sci. 2 (2), 98 (2021).
14. M. M. El‑Gayar , Mohamed Abouhawwash, S. S. Askar4 and Sara Sweidan,’ A Novel Approach For Detecting Deep Fake Videos Using Graph Neural Network’, 2024.
15. Aarti Karandikar, Vedita Deshpande, Sanjana Singh, Sayali Nagbhidkar, Saurabh Agrawal,’ Deepfake Video Detection Using Convolutional Neural Network’ **,** 2020.
16. Deng, Liwei, Hongfei Suo, and Dongjie Li. "Deepfake video detection based on EfficientNet-V2 network." *Computational Intelligence and Neuroscience* 2022 (2022).
17. Saravana Ram, R., et al. "Deep Fake Detection Using Computer Vision-Based Deep Neural Network with Pairwise Learning." *Intelligent Automation & Soft Computing* 35.2 (2023).
18. Amerini, Irene, et al. "Deepfake video detection through optical flow based cnn." *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 2019.
19. Rana, Md Shohel, et al. "Deepfake detection: A systematic literature review." *IEEE access* 10 (2022): 25494-25513.
20. Khalil, Samar Samir, Sherin M. Youssef, and Sherine Nagy Saleh. "iCaps-Dfake: An integrated capsule-based model for deepfake image and video detection." *Future Internet* 13.4 (2021): 93.
21. Ramadhani, Kurniawan Nur, and Rinaldi Munir. "A comparative study of deepfake video detection method." *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*. IEEE, 2020.
22. Mitra, Alakananda, et al. "A machine learning based approach for deepfake detection in social media through key video frame extraction." *SN Computer Science* 2.2 (2021): 98.
23. Tyagi, Shobhit, and Divakar Yadav. "A detailed analysis of image and video forgery detection techniques." *The Visual Computer* 39.3 (2023): 813-833.
24. Hasan, Haya R., and Khaled Salah. "Combating deepfake videos using blockchain and smart contracts." *Ieee Access* 7 (2019): 41596-41606.
25. Malik, Asad, et al. "DeepFake detection for human face images and videos: A survey." *Ieee Access* 10 (2022): 18757-18775.
26. Hu, Juan, et al. "Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network." *IEEE Transactions on Circuits and Systems for Video Technology* 32.3 (2021): 1089-1102.
27. Kim, Eunji, and Sungzoon Cho. "Exposing fake faces through deep neural networks combining content and trace feature extractors." *IEEE Access* 9 (2021): 123493-123503.
28. Hu, Zhihang, Turki, and Jason TL Wang. "Generative adversarial networks for stochastic video prediction with action control." *IEEE Access* 8 (2020): 63336-63348.
29. Hoque, Mohammad A., et al. "Real, forged or deep fake? Enabling the ground truth on the internet." *IEEE Access* 9 (2021): 160471-160484.
30. Saravana Ram, R., et al. "Deep Fake Detection Using Computer Vision-Based Deep Neural Network with Pairwise Learning." *Intelligent Automation & Soft Computing* 35.2 (2023).