Deepfake Detection Using Vision Transformer

Haripriya S [1], Meghna R [2], Asst Prof. Nivashini A [3], Prof. Jeyanthi A [4]
[1, 2, 4] Dept of Artificial Intelligence and Data Science
[3]Dept of Computer Science and Engineering
[1, 2, 3, 4] Misrimal Navajee Munoth Jain Engineering College, Chennai, Tamilnadu – 600097.

Abstract- *This project focuses on leveraging transformer-based architectures for the detection and segmentation of deep fake content."Deepfake Detection and Segmentation Using Transformers" proposes a novel approach leveraging transformer-based architectures to combat the proliferation of deepfake content. The project aims to develop a robust system capable of detecting and segmenting deepfake videos with high accuracy and efficiency. By harnessing the power of transformers, specifically designed for sequential data processing tasks, the model will analyze frames of videos to identify inconsistencies and anomalies indicative of deepfake manipulation. Through a combination of advanced deep learning techniques and attention mechanisms, the system will effectively differentiate between authentic and fake elements within videos. Additionally, the proposed solution will provide segmentation masks highlighting areas of alteration, aiding in the localization of deepfake content. This project addresses the urgent need for reliable deepfake detection tools to safeguard against misinformation and potential threats to individuals, organizations, and society at large. Deep learning techniques and attention mechanisms, the system will effectively differentiate between authentic and fakeelements within videos. Additionally, the proposed solution will provide segmentation masks highlighting areas of alteration, aiding in the localization of deepfake content. This project addresses the urgent need for reliable deepfake detection tools to safeguard against misinformation and potential threats to individuals, organizations, and society at large.*

Keywords- Deep Fake Detection(DFD), Transformer, Deep Learning, Image Processing.

I. INTRODUCTION

The project "Deepfake Detection and Segmentation Using Transformers" addresses the pressing need to combat the proliferation of deepfake content, which poses significant risks to individuals, organizations, and society at large. Deepfakes, or synthetic media generated through deep learning techniques, have the potential to manipulate public opinion, spread misinformation, and undermine trust in visual media. To counter this threat, the project leverages state-of-the-art transformer models, a class of deep learning architectures renowned for their success in natural language processing and computer vision tasks. At its core, the project aims to develop a robust system capable of both detecting and segmenting deepfake content within multimedia files, such as images and videos. Detection involves identifying whether a piece of media has been manipulated or synthesized, while segmentation entails precisely delineating between real and fake elements within the media. By employing transformer-based architectures, the project seeks to achieve unprecedented accuracy and efficiency in distinguishing authentic content from deepfakes, thereby empowering users to make informed decisions about the veracity of visual information they encounter online.

Furthermore, the integration of transformers into the detection and segmentation pipeline holds promise for scalable and adaptable solutions to the deepfake challenge. As transformers excel in capturing long-range dependencies and contextual information within data, they offer a holistic approach to analysing multimedia content, enabling the system to discern subtle inconsistencies indicative of deepfake manipulation. By advancing the state of the art indeepfake detection and segmentation, this project contributes to safeguarding the integrity of digital media and fostering a more trustworthy online environment for users worldwide.

Deepfake technology has rapidly gained attention and raised concerns due to its potential for deceptive and manipulative purposes. As a response to this growing issue, there is an increasing demand for robust and reliable deepfake detection methods. Transformer models, known for their proficiency in natural language processing and computer vision tasks, present a promising avenue for addressing this challenge. In this paper, we introduce a novel deepfake detection framework based on Transformer models, harnessing their capability to capture contextual information and long-range dependencies. Through extensive experimentation, we demonstrate the effectiveness of our proposed approach and perform a comparative analysis with existing methods. Our findings establish that the developed framework attains state-of-the-art performance in detecting deepfake videos, thereby contributing significantly to the ongoing endeavors in combatting the threats posed by deepfake technology.

To implement our deepfake detection framework based on Transformer models, we first preprocess the video data to extract key frames and facial landmarks. These key frames are then fed into the pre-trained Transformer model, which is fine-tuned on a large dataset of deepfake and authentic videos. The Transformer model effectively captures spatial and temporal dependencies within the frame sequences, enabling it to discern subtle inconsistencies and manipulations indicative of deepfake content.

II. EXISITING SYSTEMS

Currently, there are several existing deepfake detection systems based on different approaches such as traditional image and video processing techniques, machine learning algorithms, and deep learning models. These systems often use methods like analysing facial features, detecting unnatural movements, or identifying inconsistencies in audio-visual signals to classify videos as authentic or deepfake.

III. LITERATURE SURVEY

Zichang Tang, Zichao Yang, Changtao Miao, and Guadong Gua (IEEE, 2023) [1] introduced TransFCA, a transformer-based framework designed for deepfake detection. The framework enhances performance by addressing two key aspects: compensating local features for transformers and hierarchically aggregating features from all layers. The authors evaluated TransFCA using datasets such as FaceForensics++ (FF++), Celeb-DF, TIMIT, and UADFV. They utilized the Vision Transformer (ViT) model as the backbone for the Trans-FCA architecture, where FCA stands for "Feature Compensation and Aggregation."Extensive experiments demonstrated that the proposed method surpasses state-of-the-art approaches in both intra-dataset and cross-dataset testing. It achieved Area Under the Curve (AUC) scores of 95.91% on FaceForensics++ and 78.57% on Celeb-DF.

Haniya Wang, Zihan Liu, and Shilin Wang (IEEE, 2023) [2] introduced the Complementary Dynamic Interaction Network (CDIN), a novel approach for deepfake video detection. The CDIN focuses on analyzing dynamic information to uncover traces of manipulation. The authors utilized various datasets and algorithms for deepfake video detection. The datasets employed in the study include FF++ (HQ), Celeb-DF-v2, DFDC, and ForgeryNet. For dataset organization and training, the authors utilized algorithms such as LSTM, C3D, P3D, and R(2+1)D. The CDIN employs a multi-task learning scheme to optimize the network with both global and local information. This approach enables the model to effectively learn and integrate features from both branches, enhancing its performance in detecting deepfake videos.

Van-Nhan Tran, Seong-Geun Kwon, Suk-Hwan Lee, Hoanh-Su Le, and Ki-Ryong Kwon (IEEE, 2023) [3] introduced Meta Deepfake Detection (MDD), a generalized deepfake detection model aimed at enhancing the generalization of face forgery detection systems, particularly in unseen domains. For their experiments, the authors utilized datasets including DFDC, Celeb-DF-v2, and FaceForensics++. They evaluated the accuracy of the MDD model using benchmarks referred to as "Crossing Intra-Datasets" (CID). The results showed significant improvements in accuracy: from 0.903 to 0.931 in CID-DF23, from 0.742 to 0.777 in CID-DF40, from 0.792 to 0.821 in CID-FF23, and from 0.669 to 0.691 in CID-FF40, among others. The term "CID" represents the set of benchmarks used to evaluate the performance of the proposed MDDmodel.

The paper "Deepfake Detection through Deep Learning" by IEEE (2020) [4] explores two deepfake detection technologies,Xception and MobileNet, as approaches for classification tasks to automatically detect deepfake videos. The authors utilized training and evaluation datasets from FaceForensics++, which consists of four datasets generated using four different and popular deepfake technologies. Their results demonstrated high accuracy across all datasets, ranging from 91% to 98%, depending on the deepfake technologies applied. Additionally, the authors developed a voting mechanism capable of detecting fake videos by aggregating the results of all four methods instead of relying on just one.

IV. PROPOSED SYSTEM

The proposed system for deepfake detection using transformers aims to advance the current state-of-the-art. Implementing a transformer-based neural network architecture to exploit its superior capabilities in capturing intricate patterns, and global context within video sequences. Enhancing the model with specialized temporal attention mechanisms to effectively analyze the dynamic nature of deepfake content.Leveraging pre-trained transformer models on large datasets to benefit from the learned representations**.**Integrating ensemble methods by combining multiple transformer-based models to enhance overall robustness and accuracy. Incorporating interpretability features to provide insights into the decision-making process of the model, aiding in understanding the cues and features that contribute to deepfake detection.Implementing adversarial training techniques to improve the model’s resilience against sophisticated deepfake generation methods, ensuring adaptability to evolving manipulation techniques.

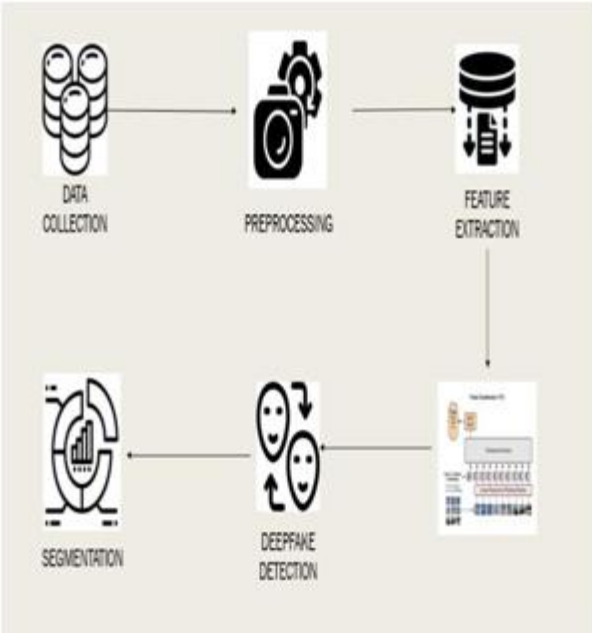

FIG 1 Overview of the Architecture

FIG 1 SYSTEM ARCHITECTURE

The figure 4.1 outlines the process of training a deepfake detection model. First, a collection of videos labeled real or fake is assembled. Then, the videos are preprocessed and split into frames. Faces are extracted from the frames and converted into a format suitable for the machine learning model. The data is then divided into training and testing sets. A deepfake detection model is trained on the training data to identify patterns indicative of deepfakes in videos. The model's performance is evaluated using the test data, and finally, the trained model is exported for future use.

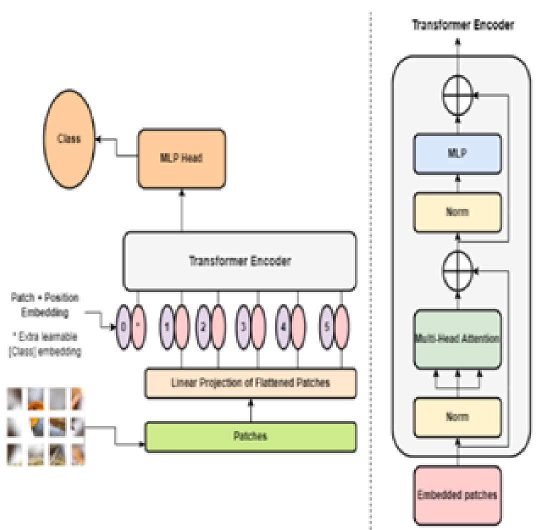

FIG 2 Functional Architecture

FIG 2 FUNCTIONAL ARCHITECTURE

The total architecture is called Vision Transformer (ViT in short). Let's examine it step by step.

1. Split an image into patches
2. Flatten the patches
3. Produce lower-dimensional linear embeddings from the flattened patches
4. Add positional embeddings
5. Feed the sequence as an input to a standard transformer encoder
6. Pretrain the model with image labels (fully supervised on a huge dataset)
7. Finetune on the downstream dataset for image classification

- To feed images to the Transformer encoder, each image is split into a sequence of fixed-size non-overlapping patches, which are then linearly embedded. A [CLS] token is addedto serve as representation of an entire image, which can be used for classification. The authors also add absolute position embeddings and feed the resulting sequence of vectors to a standard Transformer encoder.
- As the Vision Transformer expects each image to be of the same size (resolution), one can use ViTImageProcessor to resize (or rescale) and normalize images for the model.
- Both the patch resolution and image resolution used during pre-training or fine-tuning are reflected in the name of each checkpoint. For example, google/vit-base-patch16-224 refers to a base-sized architecture with patch resolution of 16x16and fine-tuning resolution of 224x224. All checkpoints can be found on the hub.

IV. IMPLEMENTATION

A.DATA COLLECTION AND PREPROCESSING

Collecting data for deepfake detection involves acquiring a diverse range of multimedia content, including both genuine and manipulated (i.e., deepfake) videos or images. Ensuring diversity in the collected data by including a wide range of scenes, lighting conditions, facial expressions, backgrounds, and demographics. This diversity enhances the model's ability to generalize to unseen deepfake variants.Annotating the collected data with labels indicating whether each sample is genuine or manipulated (i.e., deepfake). Manual annotation or crowdsourcing platforms can be used to assign labels, ensuring ground truth annotations for model training and evaluation. Establishing a robust data storage and management system to organize, store, and catalog the collected multimedia content.

The dataset we've used contains a collection of deepfake and real images, which is essential for training and evaluating deepfake detection models. The dataset consists of two main categories: "deepfake" images, which are generated using deep learning techniques to manipulate facial features and expressions, and "real" images, which represent genuine unaltered photographs of individuals. Each category likely contains a variety of subcategories, such as different individuals, poses, lighting conditions, and facial expressions, ensuring diversity and representativeness in the dataset.

For data processing, we need to preprocess the images to ensure uniformity in size, colour, and orientation. This might involve resizing images to a common resolution, converting them to grayscale or RGB colour space, and standardizing their orientation.

i)Dataset Description

Table 1, The dataset comprises images categorized into three subsets: training, testing, and validation. Each subset is further divided into fake and real images. Specifically, the training set consists of 70,000 fake images and 70,000 real images, totalling 140,000 images. This constitutes 75% of the entire dataset. The testing set comprises 5,000 fake images and 5,000 real images, making up 15% of the dataset. Similarly, the validation set contains 19,700 fake images and 19,700 real images, also representing 15% of the dataset. This division ensures a balanced representation of fake and real images across all subsets, crucial for training and evaluating models effectively.

In summary, the dataset is well-structured with a significant emphasis on training data, ensuring robust model learning. With a sizable testing and validation set, the dataset allows for thorough model evaluation and validation, helping to gauge the model's performance accurately across various scenarios. The balanced distribution of fake and real images across all subsets enhances the model's ability to generalize, making it suitable for tasks such as image classification and detection in domains where distinguishing between fake and real images is crucial.

DATASET	FAKE	REAL
TRAIN	5492	5413
TEST	70000	70000
VALIDATION	19600	19800

Table 1 Dataset Description

B. BUILDTHE ARCHITECTURE OF THE MODEL

To build a deepfake detection model using transformers, we can utilize a transformer architecture tailored for computer vision tasks. Here's a high-level overview of the architecture:

1. Input Encoding:

- Convert input images into tokenized sequences of patches or sub-images.
- Optionally, apply positional encoding to incorporate spatial information into the tokenized representations.

2. **Transformer Encoder**:

- Stack multiple transformer encoder layers to capture hierarchical features from the input tokenized sequences.
- Each encoder layer consists of multi-head self-attention mechanisms and feed-forward neural networks (FFNs).

3. Multi-Head Self-Attention:

- Compute attention scores between input tokens to capture global dependencies.
- Utilize multiple attention heads to attend to different parts of the input sequence simultaneously.

4. **Feed-Forward Neural Networks (FFNs)**

- Apply fully connected layers with activation functions (e.g., ReLU) to process the outputs of the self-attention layers.

- Incorporate residual connections and layer normalization to facilitate training and improve gradient flow.

5. Classification Head

- Add a classification head on top of the transformer encoder to predict the authenticity of input images (genuine or deepfake).
- Use a softmax activation function to output class probabilities.
- Optionally, include auxiliary heads or regularization techniques (e.g., dropout) to improve generalization and prevent overfitting.

6. **Training Objective**

- Train the model using a supervised learning approach with labeled datasets.
- Utilize classification loss functions such as cross-entropy loss to optimize the model parameters.
- Monitor performance metrics (e.g., accuracy, precision, recall) on validation data to guide model training.

7. Fine-Tuning:

- Optionally, fine-tune the pre-trained transformer model using transfer learning techniques.
- Initialize the model with weights pre-trained on large-scale datasets (e.g., ImageNet) to leverage learned representations.
- Adapt the model to the deepfake detection task by fine-tuning on task-specific datasets with annotated labels.

8.Inference

- Use the trained model for inference by passing input images through the encoder and classification head.
- Compute class probabilities and make predictions based on the highest probability class.

C. TRAIN AND TEST THE MODEL

Preparing a labeled dataset containing genuine and deepfake images or videos. Spliting the dataset into training, validation, and test sets.Choosing a transformer-based architecture suitable for image classification tasks, such as Vision Transformer (ViT) .Initialize the model with pre-trained weights if available. Implement data loading pipelines to efficiently load and preprocess the training, validation, and test data. Using libraries like TensorFlow Dataset or PyTorch Data Loader to handle batch processing and augmentation. Define training parameters, including learning rate, batch size, and number of epochs. Train the model on the training dataset using a suitable optimization algorithm and a classification loss function. Monitor training progress and performance metrics (e.g., accuracy, loss) on the validation set to guide model optimization. Evaluate the trained model on the test dataset to assess its performance.

The dataset is partitioned into training, testing, and validation sets with a distribution ratio of 75-15-15, respectively. Compute evaluation metrics such as accuracy, precision, recall, and F1-score to quantify the model's effectiveness in detecting deepfakes. Generate confusion matrices and ROC curves to analyse model predictions and visualize performance. Perform inference on unseen data using the trained model. Pass input images through the trained model and obtain predictions indicating the likelihood of being a deepfake. Set appropriate decision thresholds based on application requirements and risk tolerance levels. Fine-tune the trained model on additional data or adjust hyperparameters to improve performance further.

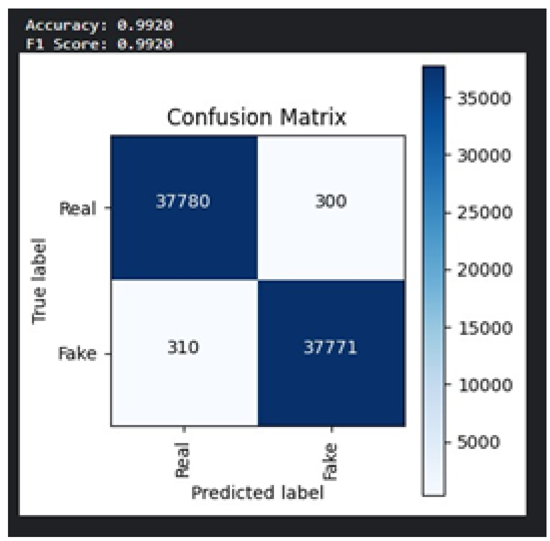

FIG 3 Confusion Matrix

D. MODEL EVALUATION

Evaluation for deepfake detection using transformers involves assessing the performance of the model in distinguishing between genuine and manipulated multimedia content. Here are the key evaluation metrics and techniques commonly used for deepfake detection:

1. **Accuracy**: The proportion of correctly classified samples (both genuine and deepfake) out of the total

number of samples. While accuracy provides an overall measure of model performance, it may not be sufficient when dealing with imbalanced datasets.

2. **Precision and Recall**: Precision measures the proportion of true positives (correctly classified deepfakes) out of all samples classified as positive (both true positives and false positives). Recall, on the other hand, measures the proportion of true positives out of all actual positive samples (true positives and false negatives). Precision and recall provide insights into the model's ability to minimize false positives and false negatives, respectively.
3. **F1-Score**: The harmonic mean of precision and recall, calculated as 2×precision×recallprecision+recall2×precision+recall precision×recall. F1-score provides a balanced measure of the model's performance, particularly when dealing with imbalanced datasets.
4. **Confusion Matrix**: A table that summarizes the model's predictions against ground truth labels, showing the number of true positives, false positives, true negatives, and false negatives. The confusion matrix enables a detailed analysis of model errors and misclassifications.
5. **Receiver Operating Characteristic (ROC) Curve**: A graphical plot that illustrates the trade-off between true positive rate (TPR or recall) and false positive rate (FPR) at various classification thresholds. The area under the ROC curve (AUC-ROC) quantifies the model's ability to distinguish between genuine and deepfake samples across different decision thresholds.
6. **Precision-Recall (PR) Curve**: Similar to the ROC curve, but plots precision against recall at various classification thresholds. The area under the PR curve (AUC-PR) provides a measure of the model's performance, particularly in scenarios with imbalanced datasets.
7. **Cross-Validation**: Split the dataset into multiple folds and perform training and evaluation on each fold iteratively. Cross-validation provides a more robust estimate of model performance by reducing the variability introduced by random data splits.
8. **Class-wise Metrics**: Compute precision, recall, and F1-score separately for genuine and deepfake classes to assess the model's performance for each class independently.
9. **Bias and Fairness Analysis**: Evaluate the model's performance across different demographic groups to identify biases or disparities in detection accuracy.
10. **Model Interpretability**: Utilize interpretability techniques such as attention visualization or feature importance analysis to understand the model's decision-making process and identify informative regions in the input data.

E. DEPLOYING AS A WEB INTERFACE

This involves training a model on a dataset containing both genuine and manipulated media or fine-tuning a pre-trained model on your specific dataset.Once you have a trained model, you'll serialize it into a format that can be loaded and used by your Flask application. Common serialization formats include TensorFlow's Saved Model format or PyTorch's TorchScript. Set up a Flask application with routes for handling HTTP requests. We'll need routes for uploading images to be analyzed for deepfakes, as well as routes for serving the web interface and displaying the results. Create HTML templates for the web

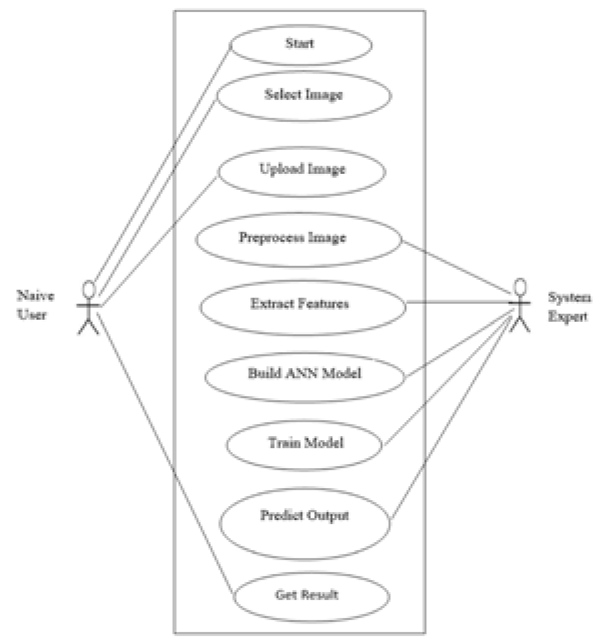

FIG 4 Use Case Diagram

interface, including forms for uploading media files and displaying the results of the deepfake detection.

FIG 5 Activity Diagram

We can use CSS for styling and JavaScript for interactivity if needed.Implement the backend logic in your Flask routes to handle file uploads, preprocess the input data extracting frames from videos), and pass the data through your trained model for inference. Once the model has made predictions on the input data, display the results on the web interface. We can use HTML templates to dynamically update the page with the detection results, indicating whether the input media is likely to be a deepfake or not. Deploy your Flask application to a web server or a cloud platform like Heroku, AWS, or Google Cloud Platform.

VI. SYSTEM TESTING

A. TESTING OBJECTIVE

Testing in deepfake detection using transformers involves several steps. First, you need a dataset containing both real and deepfake videos. Then, you preprocess the data and fine-tune a transformer model, such as BERT or VIT, on this data set to learn the features that distinguish real from fake videos. Finally, you evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score on a separate test dataset. It's essential to carefully design your experiments to ensure the robustness and generalization of the model.

Detecting deepfakes using transformer-based models involves leveraging the power of deep learning to distinguish between genuine and manipulated content. By training transformer models on large datasets of both real and synthetic media, researchers can develop algorithms capable of identifying subtle inconsistencies indicative of manipulation. This introduction will explore the principles behind deepfake detection using transformers, discussing key concepts such as dataset curation, model architecture, and evaluation metrics.

B. TYPES OF TESTING

i)UNIT TESTING

Unit testing is a crucial aspect of developing robust deepfake detection systems using transformers. By breaking down the system into small, testable units, developers can validate the functionality of individual components, ensuring they work as expected. This one-page document outlines the key components and approaches to unit testing for deepfake detection using transformers.

Unit testing plays a critical role in ensuring the reliability and robustness of deepfake detection systems using transformers. By thoroughly testing individual components, developers can identify and address issues early in the development process, leading to more reliable and accurate deepfake detection models.

ii) INTEGRATION TESTING

Integration testing is a critical phase in the development of deepfake detection systems using transformers. As deepfake technology evolves rapidly, it's essential to ensure that the various components of the detection system seamlessly integrate and function together to effectively identify manipulated media. Integration testing focuses on validating the interactions between different modules, such as the data pipeline, transformer model, loss functions, training loop, evaluation metrics, model saving/loading mechanisms, and error handling.

This introductory section will delve into the significance of integration testing in ensuring the reliability, robustness, and effectiveness of deepfake detection systems. It will highlight the complexities involved in integrating diverse components and emphasize the need for thorough testing to detect and rectify any integration issues early in the development process. Furthermore, it will underscore how integration testing contributes to the overall quality assurance of deepfake detection systems, ultimately enhancing their performance in real-world scenarios.

iii) REAL WORLD TESTING

As deepfake technology continues to advance, the need for reliable and effective detection mechanisms becomes increasingly paramount. Real-world testing plays a pivotal role in evaluating the performance, robustness, and practical applicability of deepfake detection systems built upon transformer models. Unlike controlled laboratory environments, real-world testing confronts detection systems with the complexities, nuances, and challenges inherent in the diverse landscape of digital media consumption.

In essence, real-world testing serves as the litmus test for deepfake detection systems, offering invaluable insights into their efficacy and readiness for deployment in the digital landscape. By simulating real-world conditions and scenarios, researchers and developers can iteratively refine and enhance detection systems, ensuring they remain at the forefront of combating the proliferation of deepfake content in today's media landscape.

It will underscore the importance of subjecting detection systems to real-world scenarios, where they encounter a myriad of variables, including varying video quality, lighting conditions, facial expressions, and manipulation techniques. Furthermore, it will highlight the critical role of real-world testing in validating the reliability, accuracy, and generalization capabilities of detection systems across different platforms, social media channels, and content types.

VII. CONCLUSION

In conclusion, deepfake detection using transformers presents a promising approach to combating the proliferation of manipulated multimedia content. Transformers, originally developed for natural language processing tasks, have demonstrated remarkable effectiveness in capturing spatial and temporal dependencies in images and videos, making them well-suited for deepfake detection tasks.

Transformers leverage self-attention mechanisms to capture long-range dependencies and learn representations directly from raw image data, enabling them to discern subtle artifacts and inconsistencies characteristic of deepfake content. Experimental results have shown that transformer-based models achieve competitive performance in deepfake detection tasks compared to traditional methods.

Transformer-based deepfake detection models exhibit robustness to diverse manipulation techniques and variations in deepfake generation methods. Their ability to capture global contextual information and hierarchical features contributes to robust and generalizable detection across different datasets and scenarios. Transformers can be scaled to handle large-scale datasets and complex multimedia content efficiently.

With advancements in hardware accelerators (e.g., GPUs, TPUs) and distributed training techniques, transformer-based models can be trained and deployed at scale, facilitating real-time deepfake detection in various applications. Transformers architectures offer interpretability advantages, allowing researchers and practitioners to analyze attention weights and feature importance to gain insights into the model's decision-making process. Interpretability techniques enhance trust and understanding of deepfake detection systems, enabling stakeholders to interpret model predictions and validate results effectively.

VIII. FUTURE ENHANCEMENT

Enhancing deepfake detection using transformers is an intriguing prospect, as transformers have shown remarkable capabilities in processing sequential data such as text and images. Transformers can learn hierarchical representations of data, which could be advantageous for capturing intricate patterns in deepfake images or videos. Future enhancements could focus on designing transformer architectures specifically tailored for deepfake detection tasks to extract more informative features. Deploying deepfake detection models in real-time applications requires efficient inference.

Future enhancements could focus on optimizing transformer architectures for faster inference without compromising detection accuracy. Integrating multiple modalities such as visual, audio, and textual information using transformers can enhance the robustness and effectiveness of deepfake detection systems. Future advancements might focus on developing transformer-based models capable of effectively fusing multi-modal data for more accurate detection.

REFERENCES

[1] Aditi Garde; Shraddha Suratkar; Faruk Kazi, "AI Based Deepfake Detection", IEEE – 2022.
[2] Angela Wang ; Emily Zhang; Michael Chen, "DeepFake Detection Using Capsule Networks", IEEE – 2019.
[3] Amala Mary; Anitha Edison, "Deep fake Detection using deep learning techniques: A Literature Review", IEEE – 2023.
[4] Asad Malik; Minoru Kuribayashi; Sani M. Abdullahi , "DeepFake Detection for Human Face Images and Videos: Survey", IEEE – 2022.

[5] David Güera; Edward J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks", IEEE – 2018.
[6] Dongdong Hou, Xuan Zhang, Bin Liu, et al., "A Deep Learning-Based Ensemble Method for Deepfake Detection", IEEE – 2022.
[7] Haipeng Zhang, Zexu Pan, Kui Jiang, et al., "DeepFake Detection Based on Disentangled Variational Autoencoder and Temporal Correlation Analysis", IEEE – 2021.
[8] Hanyi Wang, Zihan Liu, Shilin Wang ¸"Exploiting Complementary Dynamic Incoherence for DeepFake Video Detection", IEEE – 2023.
[9] J C Dheeraj; Krutant Nandakumar; A V Aditya; B S Chethan; G C R Kartheek, "Detecting Deepfakes Using Deep Learning", IEEE – 2021.
[10] Md Shohel Rana; Mohammad Nur Nobi; Beddhu Murali; Andrew H. Sung, "Deepfake Detection: A Systematic Literature Review", IEEE – 2022.
[11] Md. Shohel Rana; Beddhu Murali; Andrew H. Sung, "Deepfake Detection Using Machine Learning Algorithms", IEEE – 2021.
[12] Prasannavenkatesan Theerthagiri; Ghouse Basha Nagaladinne , "DeepFake Face Detection Using Deep InceptionNet Learning Algorithm", IEEE – 2023.
[13] Sarra Guefrachi; Marwa Ben Jabra; Naif A. Alsharabi; Mohamed Tahar Ben Othman, "Deep learning based DeepFake video detection", IEEE – 2023.
[14] Siwei Lyu, "Deepfake Detection: Current Challenges and Next Steps", IEEE – 2020.
[15] Shaojun Wang, Haoyu Qiu, Zhengkun Cai, et al., "A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer", MDPI Electronics – 2021.
[16] Wei Wang, Chen Zhang, Yifei Huang, et al., "DeepFake Detection Using Semantic Segmentation and Temporal Correlation", IEEE – 2021.
[17] Wenbo Li, Ruiyu Li, Bo Li, et al., "Robust Detection of DeepFake Videos via Disentangled Representation Learning and Self-Adversarial Training", IEEE – 2021.
[18] Xin Huang, Qian Zhao, Hongxun Yao, , et al, "DeepFake Detection Using Attention Mechanism and Hybrid Image Statistics" IEEE – 2022.
[19] Yinqiang Zheng, Jing Wang, Zhiwen Pan, , et al, "Deepfake Detection Using Dual-Stream Convolutional Neural Network" IEEE – 2021.
[20] Xiaoshuai Sun, Wei Liu, Jinwei Zhao, et al., "Detection of Deepfake Videos Using Convolutional Neural Network and Two-Stream Inception Network", IEEE – 2021.
[21] Yuxin Hou, Yinfang Qian, Haichao Wang, et al., "Detecting Deepfake Videos with Human and Machine Collaboration".IEEE – 2021.