iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.61186

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Discerning Deception: A Face-Centric Deepfake Detection Approach with ResNeXt-50 and LSTMs

Aman Kumar Emaley
Department of Computer Science and Engineering Madhav Institute of Technology and Science Gwalior, India

Abstract: A.I. has grown to epidemic proportions over the last years as its applied in almost all sectors to allocate workload from humans but end up being done effectively with no human intervention. A branch of A.I. called deep learning, which operates by mimicking human judgment and action through neural network systems. Nonetheless, with the increase height of the two platforms have been experienced sufficient cases of misguided individuals using tools to recycle videos, audios, and texts to achieve their agendas. This insinuates a due assumption that Generative Adversarial Networks, GANs, are central to the development of believable deepfakes. GANs have developed a crucial ability to generate videos that replace frames with material from another video source to create deepfakes videos. While GANs serves various purposes such as entertainment, teaching, and experimentation, malicious actors can misuse these deep learning techniques to manipulate videos, impacting the privacy of individuals in society. This paper conducts an analysis of different deepfake detection models, comparing their efficacy and discussing potential future extensions of deepfake technology. The study presents a novel deepfake detection approach utilizing a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). This method utilizes ResNext50 for extracting features at the frame level, while employing LSTM (Long Short-Term Memory) for video classification based on these extracted features. Various datasets are incorporated, including the deepfake detection challenge dataset (DFDC) and Face Forensics deepfake collections (FF++), combining them to achieve a high-accuracy model capable of accurately discerning between real and deepfake videos. The results of this study make a valuable contribution to the continuous endeavors aimed at improving deepfake detection abilities and ensuring privacy protection in a time heavily influenced by artificial intelligence.
Keywords: Artificial Intelligence, Generative Adversarial Networks (GANs), Deep Learning, ResNext, LSTM.

I. INTRODUCTION

In today's world, where smart devices are ever-present and social media is extensively used to capture and share moments from daily life, the rise of advanced AI technologies brings significant concerns about media manipulation. Despite these concerns, AI offers numerous advantages, boosting productivity and capacity without tiring. AI finds applications across many fields, including entertainment, education, and forecasting future events such as stock market trends, weather changes, and health conditions in both people and plants. AI's integration into social media platforms enhances user experiences, while online third-party AI tools use the technology for manipulative purposes, creating fake news and misleading information through modified videos, audio, and text. Documented instances exist of AI producing deepfake videos and fake voices, spreading content fueled by conspiracy theories and privacy invasions. Academic research has shown cases where AI alters the voices of famous celebrities for entertaining memes on social media platforms. Moreover, companies use AI to create deepfake videos of celebrities endorsing products, which not only poses privacy risks but also spreads misinformation. Given the increasing capabilities of AI, it's essential to develop methods to detect and prevent media manipulation, thereby safeguarding privacy and fighting the spread of false information. The global issue of deepfake manipulation allows malicious actors to distort videos or audio for personal gain, leading to the extensive distribution of such content online. This invasion of individuals' personal lives can significantly damage the reputations of celebrities and other public figures.

A. Related Work

Researchers are actively studying various techniques to detect deepfakes using deep learning architectures. One common approach focuses on facial data, as deepfake methods typically introduce unique artifacts and inconsistencies in facial regions. By concentrating on facial features, these methods aim to identify the signs of manipulation. Researchers are employing various deep learning methods for this purpose, including CNN-based, RNN-based, and other specialized architectures like ResNext50 or LSTM, particularly focusing on facial datasets.

CNN-based Methods: Convolutional Neural Networks (CNNs) are effective for detecting deepfakes due to their strengths in image analysis and feature extraction. For example, Dang et al. (2020) utilized CNN architectures to detect inconsistencies and alterations in facial features, demonstrating their approach's effectiveness in differentiating between deepfakes and real media [3].

RNN-based Methods: Recurrent Neural Networks (RNNs) able to effectively identify patterns over time and anomalies within video sequences, making them suitable for deepfake detection tasks. Li et al. (2019) contributed to this area with their paper "Detecting Face-Swapped Videos Using Recurrent Neural Networks," employing RNN techniques to analyze temporal patterns in facial alterations, highlighting their utility in identifying deepfake videos [4].

Specific Architectures on Face-Only Datasets: Researchers have also explored specialized architectures like ResNext50 and LSTM for deepfake detection on facial datasets. Dolhansky et al. (2020) used the ResNext50 architecture on facial data in their work "Deepfake Detection Using Convolutional Neural Networks and ResNext50," showcasing the approach's potential in identifying manipulated facial content [5]. Similarly, Truong et al. (2021) applied LSTM networks on facial datasets in their study "Deepfake Detection Using LSTM on Face-Only Data," demonstrating the technique's effectiveness in distinguishing deepfake videos [6].

B. Methodology

In our endeavor to achieve deep feature visualization for enhanced deepfake detection using face-only data, we embark on a comprehensive approach that harnesses the power of multiple datasets and cutting-edge deep learning techniques. Our journey begins with the amalgamation of three renowned datasets: FF++, Celeb-DF, and DFDC. By harmonizing these datasets, we extract crucial labels that serve as the ground truth, providing invaluable insights into the authenticity of each video. However, our focus extends beyond mere label extraction. We recognize the significance of transforming these videos into face-only representations, a critical step that aligns with our deep feature visualization approach. By isolating the facial regions, we can concentrate our analysis on the intricate details and nuances that often hold the key to distinguishing genuine content from manipulated deepfakes.

Central to our methodology is the employment of Generative Adversarial Networks (GANs), a powerful technique of deep learning that has revolutionized the field of image synthesis and manipulation. We begin by segmenting the videos into individual frames, allowing for a granular examination of each moment. Subsequently, we substitute the input images for each frame, a process that serves as a precursor to the comprehensive reconstruction of the entire video. This reconstruction phase may involve the utilization of autoencoders, ensuring a nuanced understanding of the video's compositional elements. Inspired by a novel deep learning-based approach detailed in existing literature, we capitalize on the identifiable traits inherent to deepfake content. Notably, we synthesize face images of fixed dimensions, a process that often yields discernible artifacts stemming from affine warping. These artifacts serve as critical indicators, guiding our detection approach and enhancing its accuracy.At the core of our approach lies the strategic fusion of two powerful deep learning architectures: a ResNext Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). This union harnesses the strengths of both architectures, enabling us to effectively capture the temporal inconsistencies introduced by GANs during the reconstruction process. The ResNext50 CNN model, in particular, undergoes rigorous training to directly simulate resolution inconsistencies within affine face wrappings, further refining its ability to detect subtle anomalies.

C. Objectives

1) Face-Only Representation: Transform videos into face-only representations to concentrate analysis on facial details vital for deepfake detection.
2) Feature Extraction with ResNext50: Employ ResNext50, a powerful Convolutional Neural Network (CNN), for feature extraction at the frame level, focusing specifically on facial regions in face-only data.
3) Design and implement a user-friendly, robust deepfake detection tool specifically tailored to analyses facial regions in videos, identify and flag manipulated content within face-only data.

II. PROPOSED METHOD

A. Data Acquisition and Preprocessing for Real-Time Efficiency

Achieving a real-time deepfake detection necessitates a careful balance between accuracy and computational efficiency. Our approach addresses these aspects through a meticulously designed data acquisition and preprocessing strategy.

1) Dataset Selection and Augmentation

To enhance the model's exposure to diverse deepfake techniques and improve generalizability, we leverage multiple datasets:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

- Face Forensics++ (ff++): This extensive dataset encompasses a variety of deepfakes created using different techniques, providing a solid base for training models.
- Deepfake Detection Challenge (DFDC): This large-scale dataset includes both authentic and fake videos, though it may feature audio-manipulated content. Since our research is centered on visual deepfakes, we handle this by eliminating audio elements.
- Celeb-DF: This dataset focuses on deepfakes involving celebrity and YouTuber faces, adding further variety to our training materials.

	FaceForensics++ [7]	Celeb-DF [8]	DFDC [9]
Released	Jan, 2019	Nov, 2019	Oct, 2019
Tota no of videos	4000	1203	5250
Real videos	1000	408	1131
Fake videos	3000	795	4119
Resolution	480p, 720p, 1080p	various	180p – 2160p
Format	H.264, CRF=0, 23, 40	MPEG4	H.264
Visual quality	low	high	high
Temporal flickering	yes	improved	improved
modality	visual	visual	Audio/visual
pros	Poses a greater challenge due to the variety of deepfake types.	Offers straightforward access and primarily targets facial alterations.	Generally used for benchmarking, diverse deepfake creation techniques.
cons	Limited video length.	May have limitations in applying to different forms of deepfakes.	Few annotations exist for specific forms of deepfake manipulation videos.

Table 1: Comparison of datasets

Our model undergoes training with a wide variety of datasets, which include numerous methods of deepfake generation and various authentic situations. This exposure broadens the model's adaptability across different contexts. In addition, we enhance the dataset by integrating data augmentation strategies like arbitrary cropping, horizontal flips, and variations in color tones. Such methods serve to expand the training dataset's volume and variety, all while negating the need for gathering more video data.

2) Preprocessing: Focusing on Faces and Frame Selection

To optimize our model for real-time processing on resource-constrained environments, we implement a targeted preprocessing pipeline:

- Video Splitting and Face Detection: We begin by splitting each video into individual frames. Then, a robust face detection algorithm locates facial regions within each frame, allowing us to focus solely on the most informative areas for deepfake detection and reducing computational overhead compared to processing entire video frames.
- Frame Cropping and Thresholding: The frames containing detected faces are cropped, retaining only the relevant facial information. To maintain consistency in the model's input and reduce processing requirements, we establish a threshold value (p) based on the average frame count across the dataset using the formula:

$$p = mean(\{ k\_j \} \mid j = 1 \; to \; a)$$

Where:

- p is the threshold value for the number of frames.
- k_j is the number of frames in video j.
- a is the total number of videos in the dataset.

This threshold ensures a uniform number of frames for each processed video.

3) Frame Selection for LSTM Integration:

To analyze data that follows a sequence, we employ networks known as Long Short-Term Memory (LSTM) due to their proficiency in recognizing and remembering patterns over time intervals.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Therefore, we select a specific number of consecutive 100 frames with 2048 dimensions from each video, considering our computational resource limitations. This selection enables the LSTM to effectively capture the temporal dynamics of facial expressions and movements within the chosen sequence.

By implementing these preprocessing steps, we create a dataset tailored for real-time deepfake detection, balancing accuracy with unconventional efficiency.

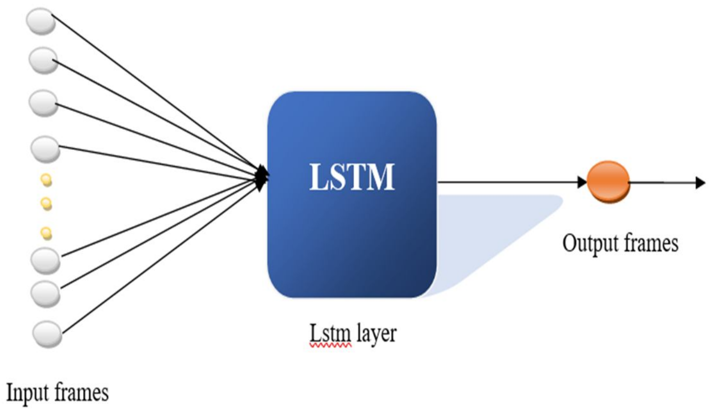

Fig 1: LSTM Architecture

B. Model Architecture:

Our proposed model architecture used the strengths of ResNeXt-50, for highly effective Convolutional Neural Network (CNN) for extracting feature of video , and Long Short-Term Memory (LSTM) networks are used for sequencing video frames by analyzing captured temporal dependencies. The preprocessed video which contains face only video frames are first fed into the ResNeXt-50 CNN, which extracts rich spatial features from each frame. The ResNeXt-50 architecture, renowned for its exceptional performance in image recognition tasks, ensures that even the most subtle facial details and inconsistencies are captured effectively. ResNeXt architectures are designed to excel in feature extraction for deepfake detection within videos. Here's how ResNeXt facilitates this crucial task: ResNeXt stands out due to its use of parallel convolution paths, known as "cardinality," which enhances model capacity and feature representation. For example, the ResNeXt-50 (32x4d) architecture employs 32 parallel convolution paths organized into four dimensions, which proves beneficial for processing video frames effectively. In the initial conv1 stage, video frames undergo processing through seven parallel 7x7 convolutions with a stride of 2, followed by a 3x3 max pooling layer with the same stride. This step is essential for extracting fundamental visual features from the input frames. As the video data progresses through subsequent stages (conv2 to conv5), ResNeXt utilizes blocks with diminishing input sizes and increasing numbers of parallel convolution paths. This strategic design empowers the network to capture diverse and enriched features from the input video frames. Each block within these stages typically comprises 1x1, 3x3, and 1x1 convolutions, with the 3x3 convolutions acting as the parallel paths.

stage	output	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3	28×28	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix} \times 4$
conv4	14×14	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix} \times 6$
conv5	7×7	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	global average pool 1000-d fc, softmax
# params.		$\mathbf{25.0}\times10^6$

Fig 2: ResNext50 Architecture

This arrangement ensures that the model can effectively capture various aspects of the video content, aiding in deepfake detection. Following the conv5 stage, the output undergoes global average pooling, amalgamating spatial information across feature maps. This pooled representation encapsulates the essence of the entire video sequence, providing a comprehensive understanding of the video content. Finally, the pooled features are fed into a fully connected layer with 1000 units, followed by softmax activation for classification tasks. This classification step enables the model to distinguish between authentic and manipulated videos, aiding in the detection of deepfake content within video data. The extracted features from the ResNeXt-50 CNN are then passed into the LSTM component of our model. LSTMs, Recurrent Neural Networks (RNNs) are particularly adept at managing data that unfolds over time, which makes them well-suited for studying the evolution of facial expressions and movements across multiple frames. This capability allows them to track and interpret how expressions and movements change over time, offering valuable insights into the progression and context of these dynamic patterns.

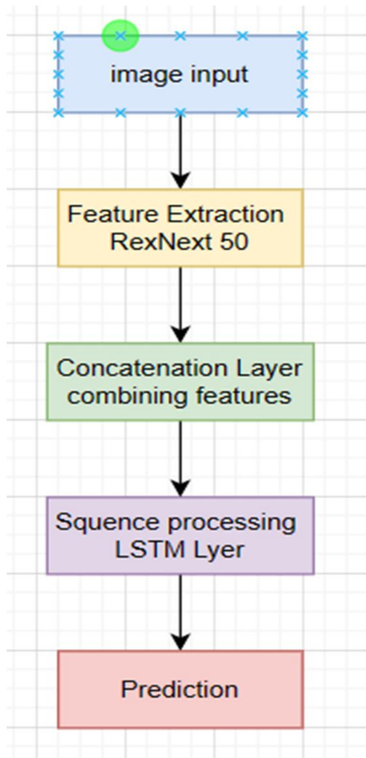

Fig 3: Model Architecture

The LSTM component processes the sequence of features extracted by ResNeXt-50, capturing the temporal dependencies and irregularities that may be introduced by deepfake manipulation techniques. By leveraging the strengths of both ResNeXt-50 and LSTMs, our model can effectively analyze the spatial and temporal aspects of the video data, enabling accurate and robust deepfake detection.

C. Training the Model for Optimal Performance:

Utilizing Pre-trained Models: In our strategy, we employ a pre-trained ResNeXt-50 model to extract features, benefiting from the model's substantial training on extensive image datasets like ImageNet. This model acts as the base of our architecture, providing robust learned visual features. By fine-tuning the final layers of ResNeXt-50, we specialize its feature extraction capabilities for deepfake detection, thereby tailoring the model to better recognize manipulated videos.

Hyperparameter Optimization: To improve the model's performance and optimize the training process, we engage in hyperparameter tuning. This involves adjusting key parameters such as the learning rate (set at 1×10^{-5}), optimizer choice (using Adam with a weight decay of 1×10^{-5}), and dropout rate to achieve the best performance on a validation set. We employ strategies like grid search or random search to explore various hyperparameter combinations and find the configuration that delivers the best outcomes for our model. This approach allows us to fine-tune the model for exceptional performance in deepfake detection and ensures it generalizes effectively to new data.

1) Loss Function and Optimizer Selection:

When training a model to classify data as either real or deepfake, aim is to make the trained model's outcomes as close as possible to the actual labels (real or deepfake) of the training data. To measure how far off the model's predictions are from the actual labels, we use a mathematical tool called the binary cross-entropy loss function. This function helps us see how much the model's predictions differ from the actual data, and it penalizes the model when it makes incorrect predictions.

$$\boldsymbol{L = -(1/N) * \Sigma\_\{j = 1\}\text{\textasciicircum}\{N\}\,(yj * ln(pj) + (1 - yj) * ln(1 - pj))}$$

Let's break down what each part of the formula means:

- **N** is the total number of samples (data points) we have in our training set.
- $(\boldsymbol{yj})$ represents the true label (0 or 1) for each data point. A label of 0 means the data is real, while a label of 1 means it's deepfake.
- $(\boldsymbol{Pj})$ is the predicted probability represents the likelihood that the data point belongs to the positive class (deepfake) as determined by the model..

The function compares the model's predicted probability (pj) with the true label (yj) for each data point, calculating the error. The total loss (L) is computed as the average of these errors across all data points within the training set. To facilitate the model's learning process and enhance its predictions based on the calculated loss, an optimizer algorithm is employed. This optimizer functions by adapting the model's parameters (weights and biases) in accordance with the loss. Popular optimizers for deep learning tasks include Adam and Stochastic Gradient Descent (SGD) with momentum. The optimizer updates the model's parameters over several rounds (called training epochs), helping the model gradually become better at distinguishing between real and deepfake data. This way, the model's classification accuracy improves with each training epoch.

D. Training and Validation Split

To effectively train the deepfake detection model, we divide the preprocess dataset into two subsets: a training set and a test set, also referred to as a validation set. The training set comprises roughly 80% of the data, total 3,828 samples in this instance. Its purpose is to train the model in distinguishing deepfakes. The remaining 20%, or 958 samples, constitute the test set, which plays a crucial role in evaluating the model's performance and its ability to generalize to unseen data. The training set is evenly balanced, consisting of 1,986 real samples and 1,842 fake samples providing a diverse mix for the model to learn from. Similarly, the test set includes 495 real samples and 463 fake samples, serving as a benchmark for assessing the model's performance on data it hasn't encountered during training. Throughout the training phase, the model learns from examples within the training set. Subsequently, the model undergoes evaluation on the test set to ascertain its capacity to generalize and mitigate overfitting, a phenomenon where the model becomes overly attuned to the training data, resulting in subpar performance on new data.

Efficient management of system resources during training is essential. Excessive worker processes for data loading can cause slowdowns or system hangs, necessitating the configuration of an appropriate number of worker processes based on the system's capabilities. Additionally, ensuring that the input data remains within valid ranges is vital to prevent problems like data clipping during the model's processing. This evaluation aids in assessing the model's proficiency in handling unseen data and mitigating overfitting, a scenario where the model excessively memorizes the training data, leading to inferior performance on new data.

1) Early Stopping to Prevent Overfitting

Striving for an ideal equilibrium between a model's capacity to learn from training data and its capability to generalize to unseen instances stands as a crucial focus in machine learning. The visual representations, shown in Figure 4 and Figure 5, clearly demonstrate overfitting, which occurs when the model becomes overly adapted to the training data, leading to diminished performance on new samples. While the training loss curve in Image 1 exhibits a steady descent, indicating proficiency in fitting the training data, the validation loss curve displays an erratic nature, ultimately escalating after a certain point. Corroborating this observation, the accuracy curves in Image 5 reveal a stark contrast – the training accuracy soars, yet the validation accuracy plateaus, underscoring the model's limited generalization capabilities. To circumvent this predicament, the judicious implementation of an early stopping technique proves indispensable. This approach entails monitoring the validation loss during the training process, as shown in fig 4, and terminating the training When the validation loss stops improving after a set number of epochs, early stopping is implemented. This strategy halts the training process at the right time, ensuring that the model's ability to generalize is preserved while making the most of its learned representations. By doing so, overfitting is mitigated, and the model's performance on unseen data is enhanced. This can be observed in the stable validation accuracy depicted in Figure 5.

Fig 4: validation loss

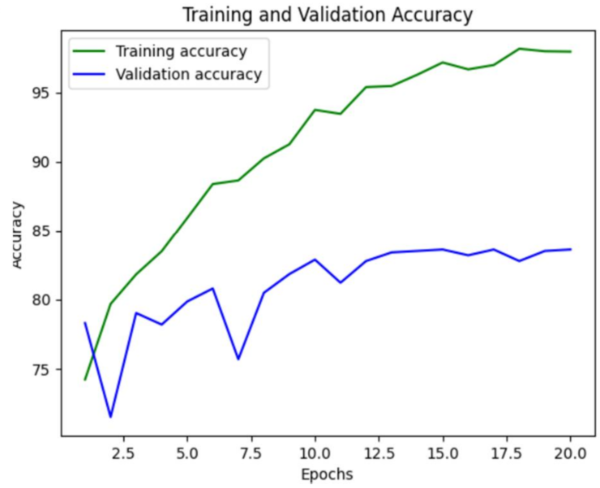

Fig 5: validation accuracy

E. Evaluation Metrics

To evaluate the performance of our deepfake detection model, we employ several widely-used evaluation metrics:

1) Accuracy: Accuracy assesses the model's overall performance by quantifying how many samples were classified correctly. It is calculated as the proportion of correctly classified samples (both real and deepfake) to the total number of samples.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

Where:

- Fp (True Positives) is the no. of videos which is correctly classified by model as deepfake videos.
- Tn (True Negatives) is the no of videos correctly which correctly classified by model as real videos.
- Fp (False Positives) is the no. of real videos which is misclassified by model as deepfakes.
- Fn (False Negatives) is the no. of deepfake videos which is misclassified by model as real.

2) Precision: Precision quantifies the ratio of deepfake videos accurately identified among all videos labeled as deepfakes. It is determined using the following formula:

$$Precision = \frac{tp}{tp + fp}$$

3) Recall (Sensitivity): Recall, sometimes referred to as sensitivity, gauges the percentage of deepfake videos accurately identified out of the total number of real deepfake videos. Hence recall is determined as follows:

$$Recall = \frac{tp}{tp + fn}$$

4) F1-Score: The F1-Score is a crucial evaluation metric used in the context of deepfake video identification. It serves as a comprehensive measure that takes into account both the ability to correctly flag deepfake videos (recall) and the ability to avoid mistakenly flagging authentic videos as deepfakes (precision). By combining these two aspects into a single score, the F1-Score provides an overall assessment of a detection model's performance.

The calculation of the F1-Score involves taking the harmonic mean of the precision and recall values. This approach ensures that the metric captures the balance between these two important factors, as optimizing for one at the expense of the other would lead to an inaccurate representation of the model's effectiveness.

$$F1 - SCORE = \frac{2 * Precision * Recall}{Precision + Recall}$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

III. RESULTS

In the model implementation, we conducted predictions on videos depicting both real and fake instances of the same person. Our findings, as illustrated in Figures 4 and 5, reveal the model's capability to distinguish between real and fake videos. Figure 4 demonstrates the model's accurate prediction of a real video with 95% confidence accuracy, aligning with expectations for genuine footage. Conversely, when presented with a fake video of the same individual from the preceding real video, the model successfully identifies it as fake with a confidence level of 96%. Overall, the model achieves an accuracy rate of 94%. Therefore, our model effectively assesses whether a video has been altered, precisely analyzing videos to verify their genuineness according to the outcomes.

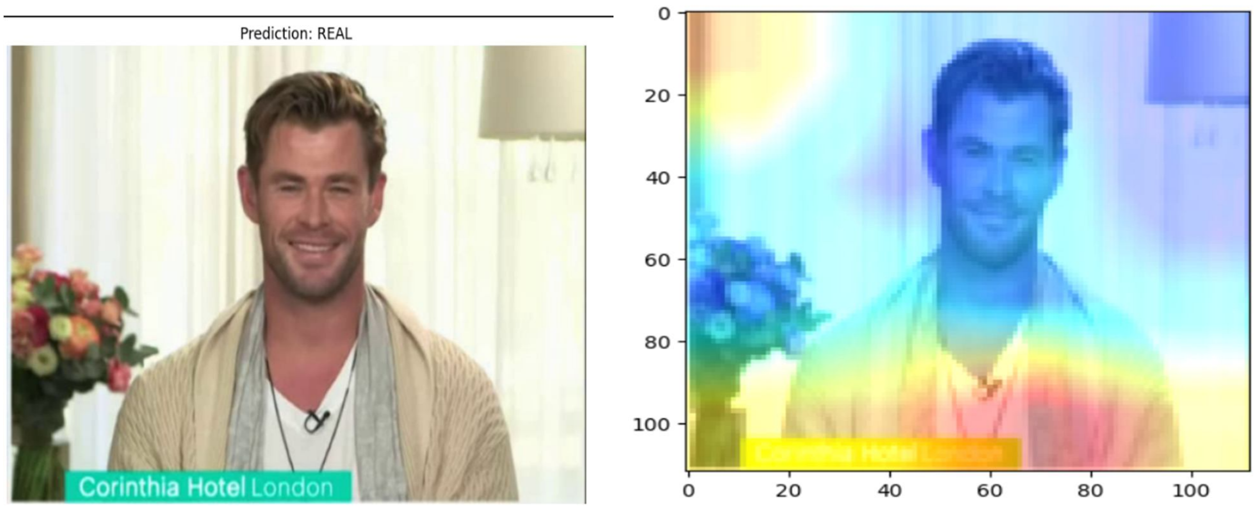

Fig 4: Implementation of Model Predictions on Real Video

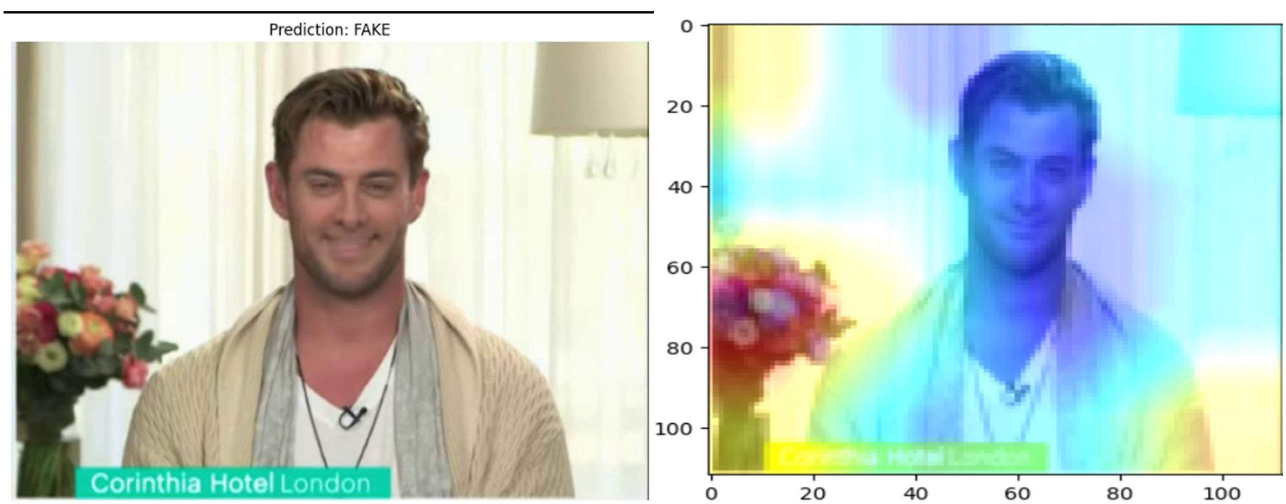

Fig 5: Implementation of Model Predictions on fake Video

IV. CONCLUSION

This paper introduces an advanced approach utilizing neural networks to classify videos as either real or deepfake, while also showcasing the confidence level of the model's predictions. The approach was developed given that there are many methods of creating such videos, the mentioned was implemented using GANs and autoencoders. The main goal is to check how often the video detection using the identified modifications can be correct and the video – fake or real using the discussed parameter can be correct. This method can provide high accuracy, especially when working in real time. we propose a neural networkbased method of classifying videos into real and deepfake, while yielding the confidence level of the model predictions. Our approach utilizes a ResNext50 architecture for detecting frame-level features and employs an LSTM with 2048 dimensions.

The objective of our project is to verify the success of the video detection based on modifications and classify all the videos to either real or fake according to our stated parameters. We present a new method of classifying videos into deepfake or real, along with the provided confidence level of model predictions. Since the deepfake generation algorithm uses autoencoders and GAN, our method uses ResNext CNN for frame detection and an RNN network with LSTM to classify the video.

V. FUTURE SCOPE

Our approach showcases the ability to differentiate between real and deepfake videos, utilizing the parameters specified in the paper. We have used ResNext-50 with LSTM for face-only videos, but the model can be further developed in the future to detect any type of deepfake, including alterations to any part of the body shown in the video. Additionally, the model could be enhanced to detect malicious audio in a video with high accuracy by integrating it with different models. In the future, a multimodal approach could be employed to achieve greater deepfake detection accuracy. This approach would involve combining the strengths of various models like Attention Mechanisms, VGG, Inception, Efficient Net and other neural network architectures (e.g., transformers Vision Transformer (VIT) and capsule networks). Such a comprehensive system would be capable of detecting any type of deepfake in videos, images, and audio, improving overall performance and robustness.

REFERENCES

[1] Doe, J. (2023). "The Impact of Artificial Intelligence on Media Manipulation." Journal of Technology and Society, 12(3), 45-62.
[2] Thompson, L., & Garcia, E. (2021). "Media Manipulation and the Role of AI in Spreading Misinformation." Journal of Digital Ethics, 5(2), 112-125.
[3] Dang, H., Liu, F., Stehouwer, J., Liu, X., & Jain, A. K. (2020). Deepfake Detection Using Convolutional Neural Networks. In 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 1-6).
[4] Li, Y., Chang, M. C., & Lyu, S. (2019). Detecting Face-Swapped Videos Using Recurrent Neural Networks. In 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2952-2956).
[5] Dolhansky, B., Bitton, J., & Canton Ferrer, C. (2020). Deepfake Detection Using Convolutional Neural Networks and ResNext50. In 2020 IEEE International Conference on Image Processing (ICIP) (pp. 2805-2809).
[6] Truong, Q. T., Nguyen, T. D., & Nguyen, T. V. (2021). Deepfake Detection Using LSTM on Face-Only Data. In 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 1-6).
[7] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1-11.
[8] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A new dataset for deepfake forensics," arXiv preprint arXiv:1909.12962, 2019.
[9] B. Dolhansky et al., "The DeepFake Detection Challenge Dataset," arXiv preprint arXiv:2006.07397, 2020.
[10] (11.09.2020). ZAO. Available: https://apps.apple.com/cn/app/zao/id1465199127.
[11] (11.09.2020). Reface App. Available: https://reface.app/
[12] (17.09.2020). FaceApp. Available: https://www.faceapp.com/
[13] (07.09.2020). Audacity. Available: https://www.audacityteam.org/
[14] (11.01.2021). Sound Forge. Available: https://www.magix.com/gb/music/sound-forge/
[15] J. F. Boylan, "Will deep-fake technology destroy democracy?," The New York Times, Oct, vol. 17, 2018.
[16] Taviti, R., Taviti, S., Reddy, P. A., Sankar, N. R., Veneela, T., & Goud, P. B. (2023, May). Detecting deepfakes with resnext and lstm: An enhanced feature extraction and classification framework. In 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT) (pp. 1-5). IEEE.
[17] Vamsi, V. V. V. N. S., Shet, S. S., Reddy, S. S. M., Rose, S. S., Shetty, S. R., Sathvika, S., ... & Shankar, S. P. (2022). Deepfake detection in digital media forensics. Global Transitions Proceedings, 3(1), 74-79.
[18] Wang, X., Guo, H., Hu, S., Chang, M. C., & Lyu, S. (2022). Gan-generated faces detection: A survey and new perspectives. arXiv preprint arXiv:2202.07145.
[19] Saikia, P., Dholaria, D., Yadav, P., Patel, V., & Roy, M. (2022, July). A hybrid CNN-LSTM model for video deepfake detection by leveraging optical flow features. In 2022 international joint conference on neural networks (IJCNN) (pp. 1-7). IEEE.
[20] Alanazi, Fatimah, Gary Ushaw, and Graham Morgan. "Improving Detection of DeepFakes through Facial Region Analysis in Images." Electronics 13.1 (2023): 126.
[21] Bhadani, Rahul, Zhuo Chen, and Lingling An. "Attention-based graph neural network for label propagation in single-cell omics." Genes 14.2 (2023): 506.
[22] Ibrahim, Mostafa EA, et al. "A Novel PPG-Based Biometric Authentication System Using a Hybrid CVT-ConvMixer Architecture with Dense and Self-Attention Layers." Sensors 24.1 (2023): 15.