IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Deepfake Detection Using LSTM and ResNext

[1]Trupti Kularkar, [2]Tanvi Jikar, [3]Vansh Rewaskar, [4]Krupali Dhawale, [5]Achamma Thomas, [6]Mangala Madankar
[1,2,3] Research Student, [4,5,6] Project Guide
[1,2,3,4,5,6]*Department of Artificial Intelligence*
[1,2,3,4,5,6]*G. H. Raisoni College of Engineering, Nagpur*

Abstract: Advanced by sophisticated deep learning algorithms, deepfake technology poses a substantial threat to contemporary society, facilitating the creation of remarkably convincing manipulated media. These manipulations often in-volve superimposing faces onto different bodies or generating fabricated speeches, thereby fueling concerns regarding misinformation, privacy breaches, and potential misuse. This study is dedicated to the creation and assessment of deep learning-based models, including CNN, RNN, ResNeXt, and LSTM, designed for the detection of deepfake videos. Additionally, we explore the potential challenges and future directions in deepfake detection research, underscoring the imperative need for ongoing advancements in this domain to stay ahead of evolving deepfake generation techniques. Ultimate-ly, this research contributes to the collective endeavor to counteract the mis-use of deepfake technology and safeguard the integrity of digital media and public discourse.

Keywords - Deepfake detection, Long-Short Term Memory (LSTM), Residual next convolution neural network.

INTRODUCTION

In the recent years, the deep learning (DL) computing has been become the top standard in the machine learning community. Deep learning progress has led to the development of software, like deepfakes, that poses threats to privacy, democracy, and national security. In a narrow definition, deepfake refers to manipulated digital media such as images of videos where the image or video of a person is replaced with another person's likeness [1]. Specifically Deep learning algorithms are employed to fabricate or alter video and audio content, making it seem as if an individual is saying or doing something they never actually did. Deepfake technology utilizes advanced AI algorithms to create manipulated videos or audio of a person, simulating their speech and actions. It's often used for humor, pornographic, or political purposes, presenting significant ethical concerns related to privacy, consent, and misinformation. Addressing these issues requires a comprehensive approach, including regulatory measures, public awareness, and responsible AI usage to mitigate potential harm [2]. Deepfakes specifically aim at social media platforms, taking advantage of the ease with which conspiracies, rumors, and misinformation can spread due to users' tendency to follow the crowd. The use of advanced deep neural networks and the abundance of data has made the manipulated images and videos nearly indistinguishable, fooling both humans and even advanced computer algorithms. This study promotes the research in the field of deep learning and deepfake detection. The main focus is on Long-Short Term Memory (LSTM), Convolution Neural Network (CNN), Recurrent Neural Network (RNN) algorithm which is specified to use in deepfake detection.

A. Convolution Neural Networks

Convolutional Neural Networks (CNNs), or convnets, are a specialized type of artificial neural network crafted for the purpose of processing and analyzing visual data. This specialization makes them a crucial tool in various applications of computer vision. They are a crucial part of the field of machine learning, particularly deep learning. CNNs excel in tasks such as image classification, object detection, image segmentation, image generation, and image enhancement [3].

A typical deep learning CNN architecture consists of three main types of layers: the convolutional layer, the pooling layer, and the fully connected (FC) layer.

1) **Convolutional Layer: -** The convolutional layer stands as a pivotal component within a CNN, playing a central role where the bulk of computations occur. It is the central building block that processes the input data. In this layer, convolutions are applied using small filter matrices (also called kernels or filters) on the input image to detect features like edges, textures, or shapes. The filter slides over the input image (or the output of the previous layer) to scan and detect patterns within the receptive field.

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

2) **Pooling Layer:-** Following convolutional layers in a CNN, pooling layers play a crucial role in diminishing the spatial dimensions of the input while preserving essential features.. Unlike convolutional layers that use filters, pooling layers employ kernels to aggregate and summarize information from small regions of the input. Typical pooling operations, such as max-pooling and average-pooling, are employed to downsample the input. This process reduces complexity and enhances the performance of CNNs. [4].

$$W_{out} = \frac{W - F}{S} + 1$$

3) **Fully Connected Layer:-** The fully connected layer is the last part of the CNN and is responsible for classification based on the features extracted in the previous layers. Fully connected implies that every node or neuron in this layer is connected to every activation unit or neuron from the preceding layer. The connections between these nodes involve weighted connections, and the activation units apply non-linear activation functions to produce the final classification output.

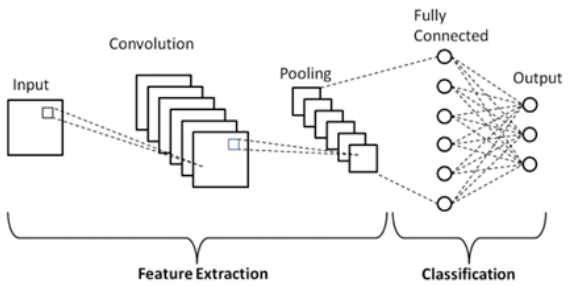

Fig 1: CNN Architecture

B. Residual Next Convolutional Neural Network

The Residual Next Convolutional Neural Network, abbreviated as ResNext, stands as a significant breakthrough in the landscape of deep learning architectures. Rooted in the fundamental concepts of the Residual Network (ResNet), ResNext introduces a novel notion known as "cardinality," significantly augmenting the model's representational prowess and computational efficiency. This innovative leap is attributed to a keen appreciation of the critical role that multi-scale information aggregation and collaborative learning play within convolutional layers. By harnessing this collaborative potential through the cardinality concept, ResNext attains exceptional accuracy in image classification tasks, all while upholding a streamlined and computationally efficient structural design.

The architecture of ResNext builds upon the Residual Network (ResNet) foundation by introducing a cardinality parameter, which controls the number of parallel paths within each building block. Each path is a small cardinal group of convolutional layers, and these paths work in parallel to enrich feature representations. The cardinality concept enhances the model's expressive capacity by promoting diversified feature learning,

enabling efficient information aggregation across multiple scales. This design allows ResNeXt to achieve remarkable accuracy and efficiency in various computer vision tasks, with the flexibility to scale up or down based on computational resources and specific application requirements.

$$F(x) := H(x) - x \; which \; gives \; H(x) := F(x) + x$$

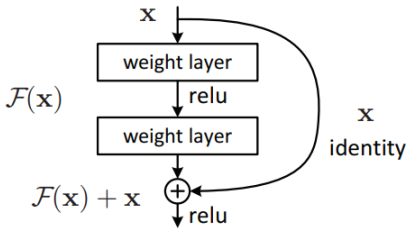

Fig. 2: ResNext Architecture

C. Recurrent Neural Network

A recurrent neural network (RNN) is a specific type of artificial neural network tailored for processing sequential or time series data. These sophisticated models excel at solving problems related to order or time, such as language translation, natural language processing (NLP), speech recognition, and image captioning. While sharing similarities with feedforward and convolutional neural networks (CNNs) in relying on training data to improve, RNNs distinguish themselves through their capacity to maintain "memory." They achieve this by incorporating information from past inputs, influencing current inputs, and generating corresponding outputs.

Standard RNNs encounter difficulties in maintaining long-term dependencies because of issues like the vanishing or exploding gradient problem during training. This can impede their ability to effectively learn and utilize information from distant past inputs. To mitigate this challenge, specialized variants such Long-Short Term Memory (LSTM) networks which is particular type of Recurrent Neural Network (RNN) are developed [6]. LSTMs incorporate distinct gating mechanisms that significantly enhance their capability to capture and retain long-term dependencies in the data. These mechanisms enable LSTMs to selectively remember or forget information, allowing for more effective learning and utilization of context from both recent and remote inputs.

$$h^{(t)} = f\left(h^{(t-1)}, x^{(t)}; 0\right)$$

D. Long-Short Term Memory

Long Short-Term Memory Networks (LSTMs) are a specialized type of recurrent neural network (RNN) used in deep learning for handling sequential data. LSTM (Long Short-Term Memory) networks tackle the common issue of the vanishing gradient in traditional RNNs, enabling effective modeling of long-term dependencies. Critical components of an LSTM include the forget gate, input gate, and output gate. The forget gate decides whether information from the prior time step should be kept or discarded. The input gate learns new information from the current input, and the output gate facilitates passing updated information to the next time step. This architecture overcomes challenges associated with capturing and retaining important information over extended sequences. Each cycle of these operations within an LSTM constitutes a single time step.

In the context of deepfake detection in videos, LSTMs are employed to process sequences of frames extracted from the video [5]. These frames undergo preprocessing to extract relevant features. The LSTM model is then trained on this preprocessed data to capture temporal patterns and dependencies present across the frames. The training dataset typically comprises both authentic and deepfake videos, enabling the LSTM to learn distinctive patterns associated with real and manipulated content. Once the LSTM is trained, it can classify unseen videos, aiding in the detection of potential deepfake content based on identified patterns and deviations from the learned normal behavior of authentic videos.

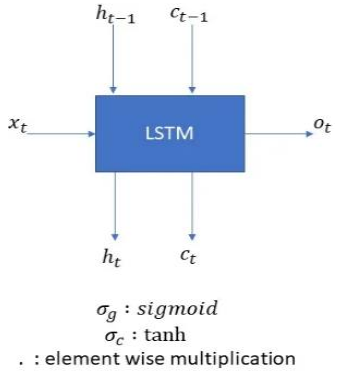

Fig. 3: LSTM Architecture

$$f_t(Forget\ gate) = \sigma_g\,(W_f \times x_t + U_f \times h_{t-1} + b_f)$$
$$i_t(Input\ gate) = \sigma_g\,(W_i \times x_t + U_i \times h_{t-1} + b_i)$$
$$o_t(Output\ gate) = \sigma_g\,(W_o \times x_t + U_o \times h_{t-1} + b_o)$$
$$c'_t = \sigma_g\,(W_c \times x_t + U_c \times h_{t-1} + b_c)$$
$$c_t = f_t . c_{t-1} + i_t . c'_t$$
$$h_t = o_t . \sigma_c(c_t)$$

LITERATURE REVIEW

In recent times, a plethora of emerging challenges are surfacing with the rapid advancement of AI technology. Particularly concerning is the potential for the misuse of deep learning algorithms to manipulate media and videos, leading to the creation of deceptive versions that pose significant security risks on various social media platforms. These manipulated media can be employed across multiple domains, including journalism, entertainment, and politics, exacerbating the potential harm they may cause. This growing trend raises important questions about the ethical use and regulation of AI technology in the context of media manipulation. Extensive research efforts have been committed to the identification of deepfakes, but the quest for achieving real-time detection remains a formidable challenge.

The study's author Jacob Mallet, Rushit Dave, Naeem Seliya and Mounika Vanamala [7] suggested that a diverse array of deep learning models employed for the detection of deepfakes were introduced and meticulously assessed. These models underwent testing using well-established datasets, recognized for their robustness in benchmarking and enabling comparative analysis. It's important to note that the outcomes of these models are significantly influenced by the specific dataset used, underlining the dataset's critical role in evaluation. Taking a broad perspective on the results, models leveraging temporal features showcased the highest levels of accuracy across both datasets. In contrast, only a limited number of models predominantly utilizing biological features were represented in this comprehensive survey. These findings showcase the efficacy of CNNs and various deep learning algorithms. In summary, the models demonstrated strong performance and highlighted the potential of current tools to mitigate the impact of deepfakes on the internet.

Authors Wahidul Hasan Abir, Faria Rahman Khanam, Kazi Nabiul Alam [8] has done a study which aims to conduct an extensive examination of deepfake detection utilizing Deep Learning (DL) methods and thoroughly analyze the outcomes of the most efficient algorithm using Local Interpretable Model-Agnostic Explanations (LIME) to ensure its trustworthiness and accuracy. The research involves the identification of authentic and deepfake images employing various Convolutional Neural Network (CNN) models to achieve optimal accuracy. Additionally, it elucidates the specific regions within the image that influenced the model's classification through the utilization of the LIME algorithm. The proposed system provides 99.87% accuracy in detecting deepfake images from real images. The study assures that, the potential market availability of such a system could alleviate the challenges posed by fake content and misinformation for numerous individuals. Ongoing advanced research holds great promise in enhancing the overall efficacy of this system, as well as in refining its explanatory capabilities. Moreover, further progress in the project can be achieved through a thorough examination of the model, incorporating updated transfer learning methods from its latest versions.

Dafeng GongIn, Yogan Jaya Kumar, Ong Sing Goh Zi Ye, Wanle Chi [9] are authors. In their research paper, a novel deepfake detection model named DeepfakeNet is introduced. The model comprises 20 network layers and draws inspiration from the concept of stacking in ResNet and the split-transform-merge approach in Inception to formulate the network block structure. This structure closely resembles the block structure found in ResNeXt. In this paper, fake face video detection is approached as a distinct problem of detecting tampering in image mosaics. The methodology involves leveraging image segmentation and deep residual networks to predict areas that have been tampered with, ultimately mitigating the influence of disparate training datasets and enhancing the detection algorithm's overall generalization performance. Through extensive experimentation on various widely-used face-swapping video datasets, the results demonstrate a substantial reduction in the average error rate of cross-dataset detection compared to similar algorithms, all while maintaining high accuracy in dataset-specific detection.

METHODOLOGY

A. Dataset

We are getting a dataset from kaggle from that comprising three different dataset: FaceForensic++ [10], Celeb-DF [11], and Deepfake Detection Challenge datasets (DFDC) [12], which has been curated to augment the video data quantity. The collective dataset encompasses approximately 6,000 videos, evenly distributed into two categories: genuine (real) and manipulated (fake) videos. For the purpose of model development, 70% of this dataset has been allocated for training, while the remaining 30% is earmarked for testing and evaluation.

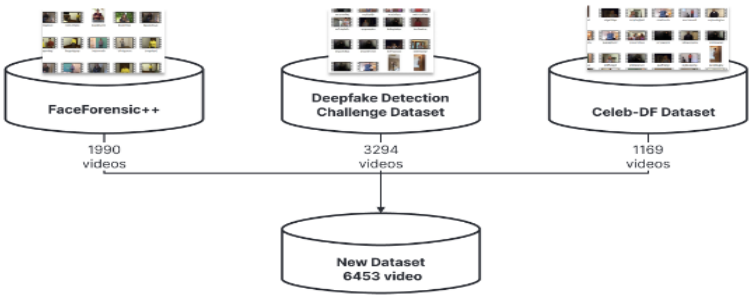

Fig. 4: Dataset

B. Preprocessing

Prior to model training, the videos undergo preprocessing. This involves face detection followed by precise face cropping to focus on facial features. This preprocessing step ensures that the model works with relevant data.

1) **Frame Extraction:**
   Initially, the video data is subjected to frame extraction, where each video is broken down into its constituent frames. This step is essential for further analysis.

2) **Face detection and cropping:**
   After frame extraction, a face detection algorithm is applied to identify and locate faces within each frame. Subsequently, the frames are cropped to isolate the detected faces. This is crucial for focusing the analysis on facial features.

3) **Uniform frame quantity:**
   To maintain consistency in the number of frames across the dataset, the mean number of frames across all videos is computed. Subsequently, a new processed dataset is generated by retaining only the frames equal to this computed mean. This ensures uniformity in the dataset for training and evaluation purposes.

4) **Frame selection:**
   Frames that do not contain detectable faces are excluded during the preprocessing stage, ensuring that only frames with relevant facial content are considered.

5) **Experimental frame limitations:**
   Given the computational intensity of processing a 10-second video at a rate of 30 frames per second, resulting in a total of 300 frames, it is proposed to limit the frame selection to the initial 100 frames for experimental purposes. This pragmatic choice is made to manage computational resources while still retaining a substantial amount of data for model training. This selection of the first 140 frames is deemed adequate for preliminary experiments and model development.

C. Model

The model architecture employed for this study comprises a ResNeXt50_32x4d as the initial backbone, followed by a single Long Short-Term Memory (LSTM) layer. The Data Loader component of the system is responsible for loading the preprocessed face-cropped videos and performing a split to create separate training and testing datasets. Subsequently, the frames extracted from the preprocessed videos are fed into the model for both training and testing purposes, utilizing mini-batch processing to efficiently handle the data. This approach allows for the training and evaluation of the model on the provided video dataset.

D. Feature Extraction

Rather than creating a new classifier from scratch, we propose leveraging the ResNeXt Convolutional Neural Network (CNN) classifier to extract relevant features and achieve accurate frame-level feature detection. Subsequently, we intend to fine-tune this network by introducing additional necessary layers and optimizing the learning rate to ensure proper convergence of the gradient descent during model training. The 2049-dimensional feature vectors obtained after the final pooling layers of the ResNext network are then utilized as the input for the sequential Long Short-Term Memory (LSTM) layer, thereby preserving the critical information for further analysis and classification.

E. Processing

In our scenario, we are working with a sequence of ResNext CNN feature vectors, representing input frames, and the task at hand is binary classification, specifically discerning whether the sequence corresponds to a deepfake video or an unaltered one. A primary challenge we face is designing a model that can effectively process sequences in a coherent and meaningful manner. To overcome this challenge, we suggest employing a Long Short-Term Memory (LSTM) unit equipped with 2049 memory cells and a dropout rate of 0.4. This specific LSTM configuration proves effective in achieving our intended goal.

LSTM is chosen for its capacity to process frames sequentially, enabling temporal analysis of the video. This analysis involves comparing the frame at time 't' seconds with frames from 't-n' seconds, where 'n' can vary, encompassing any number of frames preceding the 't' timestamp. The LSTM's sequential processing capability allows us to capture temporal dependencies and patterns, which are crucial for discerning between deepfake and genuine video sequences effectively.

F. Model Architecture

Our model architecture, represented in a coherent diagram, outlines the workflow for deepfake detection. The process begins with the 'Upload Video' phase, introducing the video into the system. This is followed by 'Video Processing,' which involves breaking down the video into individual frames for further analysis. The frames are then subjected to in-depth 'Processing,' where relevant features and data are extracted. The model, central to our system, takes on the crucial role of analysing these processed frames. Finally, based on the model's intricate analysis, the video is categorized as 'Real' or 'Fake.' This streamlined model architecture ensures not only efficiency but also robust defence against the proliferation of deceptive deepfake content.

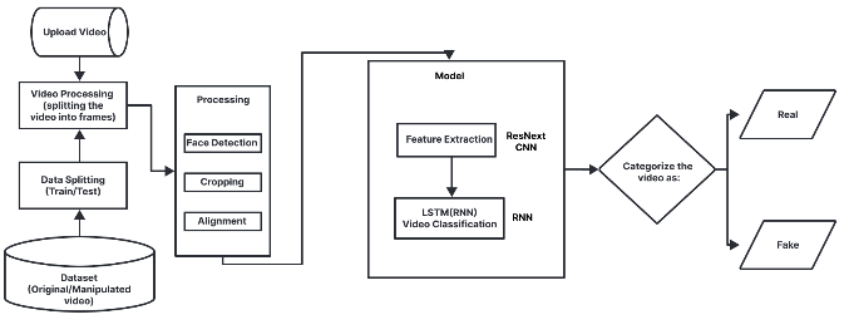

Fig. 5: Model Architecture

G. Predict

In the prediction phase, when a novel video is presented to the pre-trained model for classification, a series of preprocessing steps is applied to the incoming video to align it with the format compatible with the trained model. This preprocessing workflow involves the segmentation of the video into individual frames, subsequent extraction of facial regions through cropping, and notably, the avoidance of local storage for the video content. Instead, the cropped frames are directly fed into the trained model for the purpose of deepfake detection. This direct pipeline from frame extraction to model input streamlines the inference process and enhances computational efficiency during real-time prediction.

Components	Value
Dataset	6453
Image Dimensions	224x224 pixels
Average Frame	148 frames per video
Learning Rate	0.001 or 0.0001
Epochs	50 epochs
Calculated Accuracy	94.0980392156862%

Table 1

RESULT

The model's output will comprise a binary classification indicating whether the input video is a fake or a genuine (real) video, accompanied by a confidence score reflecting the model's level of certainty regarding the classification. A demonstrative instance of this output is visually represented in Figure for clarity and reference.

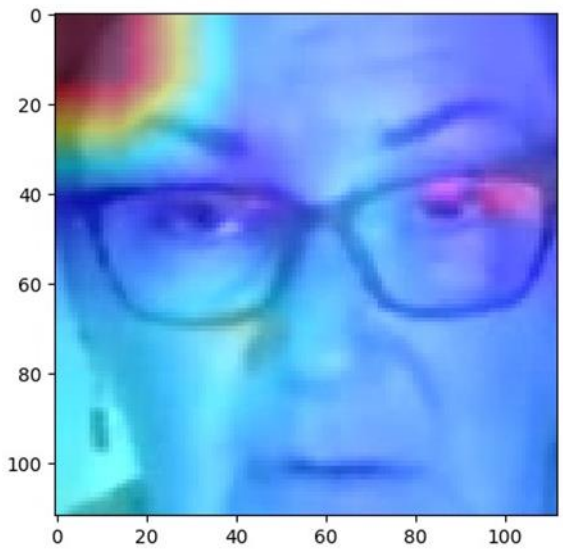

Fig 6(a): Output: Real

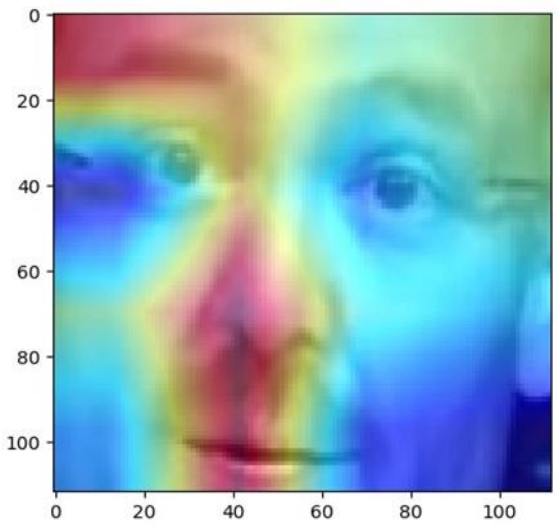

Fig. 6(b): Output: Fake

In Fig. 6(a) and Fig. 6(b), heatmap graphs are presented. These graphs depict the percentage of manipulated area in the video, ranging from Violet (0%, indicating no manipulation) to Red (100%, indicating high manipulation probability). In Fig. 6(a), the blue-highlighted portion signifies the genuine aspect of the image, with an accuracy of 80.9099, while Fig. 6(b) shows a heatmap with a red-highlighted area indicating 41.2341% manipulation probability. The model demonstrates a robust deepfake detection accuracy of 94.0980.

CONCLUSION

In conclusion, our research has successfully yielded a robust deepfake detection model, built upon the foundation of ResNeXt and LSTM networks. This model showcases promising performance in distinguishing between genuine and manipulated content, fulfilling our primary objective of enhancing deepfake detection capabilities. This achievement underscores the importance of continued research and innovation in the field of deepfake detection to bolster the security and authenticity of digital media.As deepfake technology becomes more advanced, our model is stepping up to protect us from fake videos. We're constantly working on improving it to better spot these tricky fakes and safeguard the truth in our digital world

FUTURE SCOPE

In this work we make a model which will detect deepfakes using deep learning. The future aim of this research will be Investigate the integration of multiple modalities (e.g., audio, video, metadata) for deepfake detection.

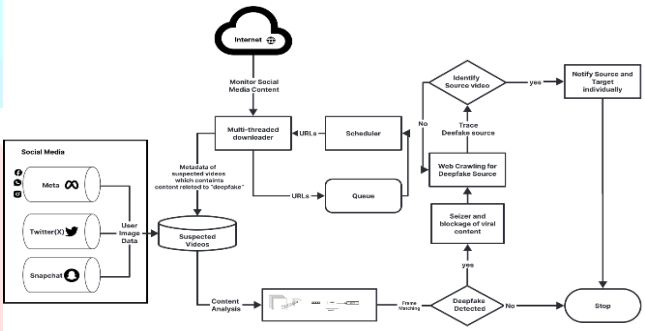

Fig. 7: Advanced Deepfake Detection Workflow

In our envisioned future developments, our advanced model, as depicted in Fig. 7, proactively surveils social media platforms for "Deepfake" content. Employing sophisticated analyses for pattern recognition and intent identification, subscribers can choose a protective service that assembles personalized datasets. Upon detecting potential deepfakes, the system promptly notifies involved individuals and restricts sharing options. Utilizing semantic analysis, the system identifies deepfake-related content, even without explicit keywords, and employs frame matching to validate the alignment of user identity with video content. Confirmed deepfakes trigger immediate content seizure, followed by source tracking using parallel computing to identify the originating video. Notifications reach both source and target individuals, ensuring a comprehensive and swift response to mitigate the impact of the deepfake. This future-focused approach integrates cutting-edge technologies and methodologies, enhancing the overall efficacy of deepfake detection and response mechanisms.

REFERENCES

[1] Nguyen, T.T., Nguyen, Q.V.H., Nguyen, D.T., Nguyen, D.T., Huynh-The, T., Nahavandi, S., Nguyen, T.T., Pham, Q.V. and Nguyen, C.M., 2022. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, *223*, p.103525.

[2] Westerlund, M., 2019. The emergence of deepfake technology: A review. *Technology innovation management review*, *9*(11).

[3] Thippanna, G., Priya, M.D. and Srinivas, T.A.S., An Effective Analysis of Image Processing with Deep Learning Algorithms. *International Journal of Computer Applications*, *975*, p.8887.

[4] Indolia, S., Goswami, A.K., Mishra, S.P. and Asopa, P., 2018. Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*, *132*, pp.679-688.

[5] Ralf C. Staudemeyer, "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks", arXiv:1909.09586v1 [cs.NE] 12 Sep 2019

[6] Güera, D. and Delp, E.J., 2018, November. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1-6). IEEE.

[7] Mallet, J., Dave, R., Seliya, N. and Vanamala, M., 2022, November. Using deep learning to detecting deepfakes. In *2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI)* (pp. 1-5). IEEE.

[8] Abir, W.H., Khanam, F.R., Alam, K.N., Hadjouni, M., Elmannai, H., Bourouis, S., Dey, R. and Khan, M.M., 2023. Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods. *Intelligent Automation & Soft Computing.*, pp.2151-2169.

[9] Gong, D., Kumar, Y.J., Goh, O.S., Ye, Z. and Chi, W., 2021. DeepfakeNet, an efficient deepfake detection method. *International Journal of Advanced Computer Science and Applications*, *12*(6).

[10] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Nießner, M., 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1-11).

[11] DFDC data from Kaggle:- https://www.kaggle.com/competitions/deepfake-detection-challenge (Accessed on 13/09/2023)

[12] Li, Y., Yang, X., Sun, P., Qi, H. and Lyu, S., 2020. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3207-3216).