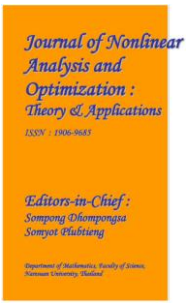
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1: 2024
ISSN : **1906-9685**



Deep Learning Fusion for Advanced Deepfake Detection with ResNet50 and LSTM

[1]**MD. Rijvana, P.V. Jahnavi, P. Harshitha,N. Mahesh Babu**, **K. Abishai**UG Students, Department of Computer Science and Engineering, Vignan's Institute ofInformation Technology(A), Visakhapatnam, Andhra Pradesh, India.
[2]**Dr. Mohan Mahanty**, Associate Professor, Department of Computer Science and Engineering, Vignan's Institute of Information Technology(A), Visakhapatnam, Andhra Pradesh, India.

ABSTRACT: In today's digital world, where what we see isn't always what it seems, the rise of deepfake technology presents a big challenge to trust in videos. These manipulated videos, created using advanced AI, it help in distinguishing between real and fake and makes it easier to spread false information. This study addresses the urgent need for reliable deepfake detection methods using Deep Learning (DL)techniques within the DFDC (DeepFake Detection Challenge) dataset. Leveraging the vast and diverse DFDC dataset, this research develops robust detection algorithms capable of discerning between authentic and manipulated media content. The proposed model aims to mitigate risks associated with deepfakes, including misinformation, reputation damage, and security threats. By utilizing the DL algorithms, thestudy contributes to the development of effective countermeasures against the harmful effects of deepfakes, safeguarding trust, credibility, and integrity in digital media. The evaluation metrics like F1-score, accuracy, and Logloss take advantage to calculate the performance. The paper includes a comprehensive review of prior studies, discussions on DL, detailed analysis of experimental data, and insights into the proposed model's effectiveness compared to existing methods. This research not only offers immediate applications but also fosters broader discussions around digital ethics and media literacy.

Keywords: Deep Learning, ResNet50, LSTM, Deepfake, DFDC

---

1. INTRODUCTION

Detecting deepfakes is critical in today's digital landscape, where the lines between reality and fabrication blur. In this study, it delves into the realm of deepfake detection, where we harness the capabilities of DL algorithms to discern between authentic and manipulated media content. The approach revolves around leveraging the vast and diverse DFDC dataset, which serves as a rich repository of deepfake videos encompassing various subjects, scenarios, and degrees of manipulation. The growth of deepfake technology poses consequential risks to human being, institutions and the community as a whole, including the potential for misinformation, reputation damage, and even security threats. By developing robust detection algorithms one can mitigate these risks and help maintain the integrity of digital content. Additionally, as the deepfake technology continuously evolving and become more advanced, it becomes a pressing need for advanced detection mechanisms to keep pace. Beyond immediate applications, the research also fosters broader discussions around digital ethics and media literacy. As shown in the Fig 1. shows the deepfake detection process.

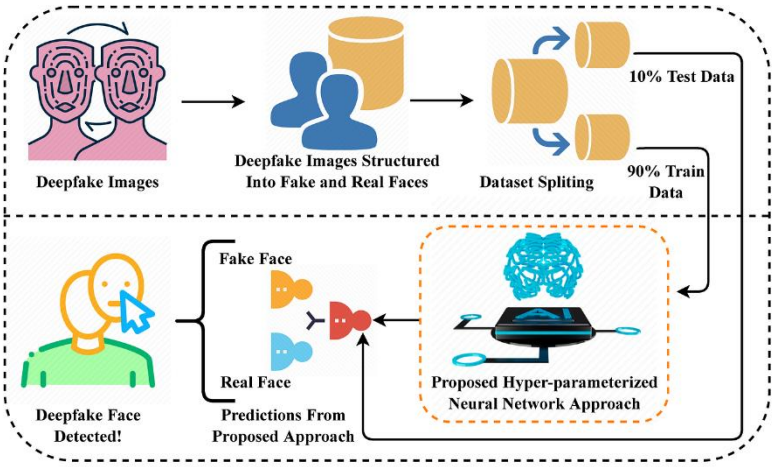


Fig 1. Deepfake Detection Process[1]

The problem statement for the project revolves around the need to develop reliable deepfake detection methods using DL techniques within the DFDC dataset. Exploiting the power of DL within the DFDC dataset, offers an opportunity to battle this expanding problem. This dataset not only provides a wealth of labeled examples but also poses significant challenges, including varying quality, resolution, and manipulation techniques. The increasing growth of deepfake technology is a growing distress about its possible misuse for spreading the wrong information, influencing public opinions and deceiving individuals and this poses challenges to different regions like journalism, politics, entertainment, and security. The objective is to address this problem by designing and implementing DL models, which are capable of identifying the differences between real and fake media with high accuracy. By doing so, the contribution to the development of effective countermeasures against the harmful effects of deepfakes, safeguarding trust, credibility, and integrity in digital media.

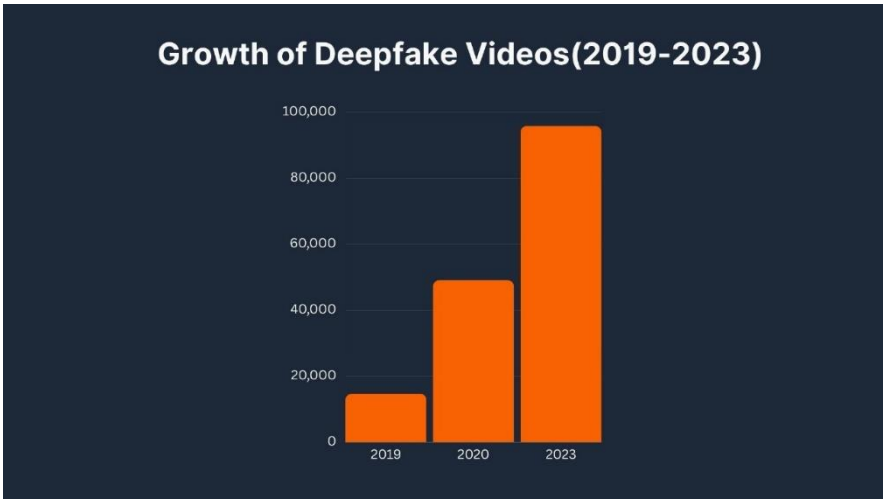


Fig 2. Number of cases recorded from 2019, 2020 and 2023[2]

The existing system for deepfake detection typically depends on a combination of traditional image and video processing techniques, manual inspection, and forensic analysis. These methods often involve identifying inconsistencies such as unnatural facial movements, mismatches in lip-syncing, and artifacts introduced during the manipulation process. Additionally, some systems utilize metadata analysis and reverse engineering of the editing software to detect signs of tampering but all these approaches have several limitations. They can be time-consuming, manual labor, and may not be scalable to handle the volume and complexity of deepfake content generated daily. Moreover, as deepfake technology advances, traditional

detection methods may become less effective at accurately identifying manipulated media. There are a lack of standardized datasets and benchmarks makes it challenging for comparing the performance of different detection mechanisms objectively. In the evaluation of the effectiveness of deepfake detection systems different kind of metrics play an important role in determiningthe performance. These metrics serve as benchmarks for assessing the system's accuracy, robustness, and reliability.

The metrics serve as essential tools for evaluating and comparing different deepfake detection systems, guiding researchers and practitioners in refining and improving the productiveness of their algorithms.

The remaining paper is arranged in the following order:

Part 2 elaborates about the literature review, in which we discussed the findings of prior studies and the model they preferred to conduct this job and their creativity in the work.
Part 3 offers a quick report on DL.
Part 4 Discussion about the proposed model and why we have chosen it over current ones.
Part 5 Has a detailed view of the experimental data analysis using our methodologies.
Part 6 discusses the problem description, conclusion, outcomes, and future scope.

2. REVIEW OF LITERATURE

Russ Howes et.al proposed the DFDC dataset[3] comprises 5K videos with two facial modification algorithms, accompanied by specific evaluation metrics and baseline performance from tested detection models. Wayne Wu et.al worked on the Deeper Forensics-1.0[4], it has a very large dataset for Authenticity Verification in Real Face Imagery. This helps in establishing benchmarks and initial insights into the challenges of detecting deepfakes, paving the way for further improvements in this field to improve the performance.

Lingzhi Li et al. developed Enhanced Face Forgery Detection using Face X-ray Technology[5], leveraging grayscale facial X-ray images to reveal if an input face image is a fusion of two distinct sources, achieving an impressive accuracy of 97.73% in identifying forgeries from prevalent face manipulation algorithms. Barsha Lamichhane et al. utilized both complete and sample datasets from the Deep Fake Detection Challenge (DFDC)[6] to evaluate their model against pretrained models like VGG-19, Xception, and Inception-ResNet-v2. Their research explores diverse constraints including different resolutions while maintaining aspect ratios of 1:1 and 9:16, achieving an accuracy of 0.3756.

Sohail Ahmed Khan et al. proposed and developed a fused Convolutional Neural Network (CNN) approach utilizing VGG16, InceptionV3, and XceptionNet architectures and Resilience Through Fused CNN Predictions[7], to enhance deepfake detection resilience, achieving a remarkable 96.50% accuracy on the dataset, surpassing other systems.

Nicol'o Bonettini et.al worked on Detecting Video Face Manipulations Using a CNN Ensemble Approach[8]. The researchers are working on detecting when faces in videos have been manipulated or altered using modern techniques. They're using a method that involves combining different types of trained CNN models. These models are based on a main network called EfficientNetB4, and they incorporate two different approaches: attention layers and Siamese training. Their solution involves using these various networks together, and they've found that this combination produces good results in detecting manipulated faces. They've tested their method on two datasets that contain over 119,000 videos available to the public, and the results look promising. The accuracy of Detecting Video Face Manipulations Using a CNN is 0.944.

Daniel Mas Montserrat et al. devised a method employing CNN Ensemble Approach[9] and RNNs to analyze facial features and temporal patterns in videos, effectively detecting manipulations with a

competitive accuracy of 92.61% on the DFDC dataset, showcasing its efficacy in identifying deepfake videos. They've tested their method on two datasets that contain over 119,000 videos available to the public, and the results look promising. The accuracy of Detecting Video Face Manipulations Using a CNN is 0.944.

Young-Jin Heo et al. proposed a novel approach using a Vision Transformer model with distillation techniques[10] to detect deepfake videos more effectively, overcoming issues like overfitting and false negatives. By employing patch embedding as input and CNN feature extraction, their method outperforms modern techniques on the dataset, and has an AUC of 0.978 and an F1 score of 91.9 without ensemble techniques.

Xiaodan Li et al. introduced the challenge of partial face attacks in DeepFake videos[11] and proposed a Sharp Multiple Instance Learning (S-MIL) approach to address it, achieving superior performance on standard datasets. Their method directly maps instance embeddings to video-level predictions, showing promise in safeguarding facial information and tackling deepfake detection.

3. DEEP LEARNING

DL is a subset of Machine Learning (ML). It computes of artificial neural networks with multiple layers. These are capable of understanding different complex patterns and representations directly from data. It has gained immense popularity and achieved remarkable success and improvement in various fields such as computer vision, natural language processing(NLP), speech recognition, and reinforcement learning. The major advantage of DL is its ability to automatically discover and learn features. This helps in making it particularly effective in handling large and complex datasets. It takes an inspiration from the structure and function of the human brain and it helps in enabling the machines to understand and learn from vastly different kinds of data. It helps in performing complex tasks with astounding performance. DL models have achieved highly developed performance in numerous amount of tasks including image classification, game playing, object detection and language translation.

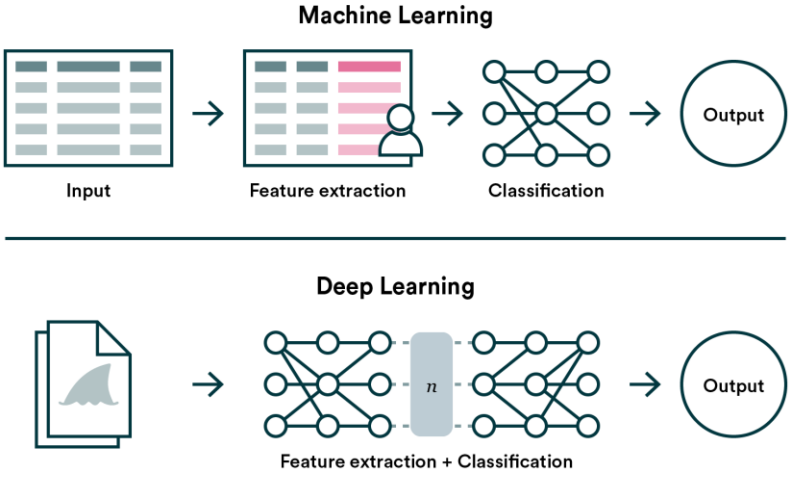


Fig 3. ML and DL process

As shown in the above Fig 3. It shows the difference between ML and DL processes. Popular architectures in DL include CNN for image-related tasks, RNN for sequential data such as text or time-series data, and transformer models for NLP tasks. From image recognition and NLP to autonomous driving and healthcare, DL has revolutionized numerous industries and continues to drive innovation.

4. PROPOSED MODEL

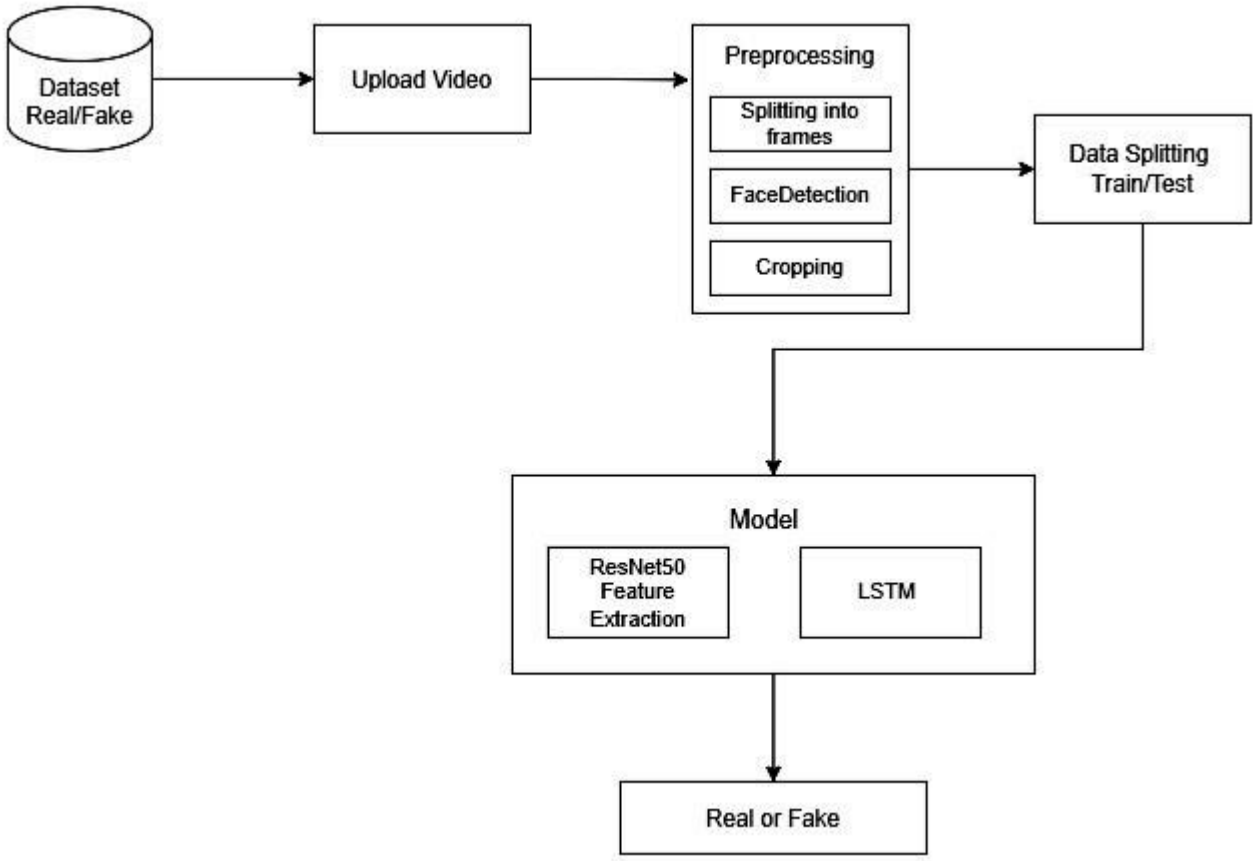


Fig 4. Deepfake Detection Model

The above Fig 4. Shows the blueprint of the deepfake detection. In the project, we need a large dataset of different kind of videos. The dataset which is being used is DFDC (Deep Fake Detection Challenge) from Kaggle and then the dataset is uploaded and sent for the pre-processing stage which includes dividing videos into frames, detecting faces, and cropping the images.

4.1 Pre-processing:

The preprocessing stage involves the extraction of frames that are uploaded from the dataset. The uploaded videos are divided into frames using the Opencv library function. We utilize the OpenCV library to perform frame extraction, this process involves iterating through each frame of the input video and saving it as an image file. After all the frames are extracted then the process of face detection occurs by capturing face details for every frame. Detecting faces for video frames is done to extract relevant features for deep fake detection. We employ face detection algorithms to locate and localize faces accurately. Once the detection of faces is completed, cropping is performed to focus on facial regions by eliminating irrelevant background information. Proper cropping ensures the model receives clean input data, enhancing its ability to distinguish between real and manipulated faces. At last, all the images are resized to the same size.

Fig 5. Augmented Images

As shown in figure 5, shows about flipping the image horizontally, like looking at it in a mirror. The flipped image becomes an augmented version of the original image. And the image obtained after applying a rotation transformation to the image, rotating it by an angle of 20 degrees. The next image obtained after applying a shearing transformation to the image along a specified axis (typically horizontal or vertical). Next is Zoom means applying a scaling transformation to the image, either zooming in (enlarging) or zooming out (shrinking) the image. And in the above image, it displays the zoomed effect of the original image. The image depicts a translation transformation to the image, shifting its pixels horizontally.

Our model contains the integration of both CNN and RNN architectures. The CNN architecture is ResNet50 and the RNN architecture is Long Short-Term Memory (LSTM) is used to improve the architecture. The feature extraction is performed by Resnet50 architecture. ResNet50 is a variant of ResNets and consists of 50 layers. Residual Networks (ResNets) introduced a breakthrough in image classification tasks by reducing the vanishing gradient problem through skip connections. All the extracted features are used by LSTM model. LSTM is a type of recurrent neural network (RNN), is specifically used for capturing dependencies over the long distribution of data. It is the fusion of ResNet50 and LSTM architectures involves integrating the ResNet50 layers with LSTM layers to enable end-to-end learning from sequential data. The ResNet50 component serves as a feature extractor, extracting hierarchical features from input data, while the LSTM component processes these features sequentially, capturing temporal dependencies and context information.

4.2 ResNet50

ResNet-50, is a CNN architecture that is part of the ResNet family. The name "ResNet50" originates from its structure as a residual network with 50 layers. ResNet50 belongs to the family of CNNs and has garnered widespread acclaim for its exceptional performance in image classification tasks. The fundamental building blocks of ResNet-50 are residual blocks, specifically the bottleneck residual block.

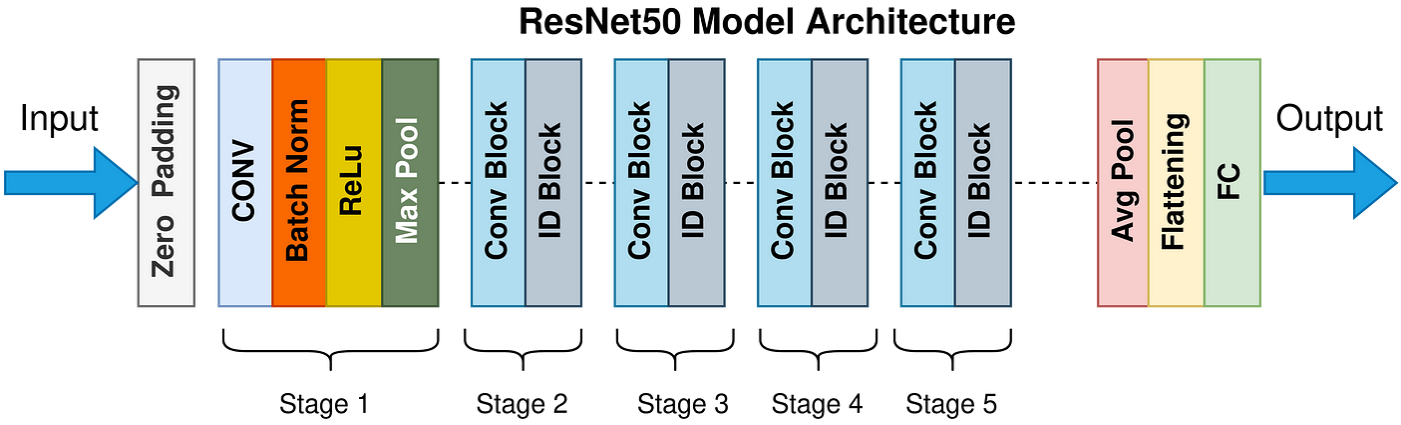


Fig 6. ResNet50 Model[12]

ResNet50 introduces skip connections to tackle the degradation problem. The above Fig 6. Shows the ResNet50 model architectureBy utilizing the connections, the network can effectively learn the residual functions by referring to the inputs of each layer, instead of trying and learning the desired underlying mapping. The architecture of ResNet50 is structured around a series of residual blocks, each containing layers featuring batch normalization, multiple convolutional rectified linear unit (ReLU) activations, and subsequently, shortcut connections. These bottleneck blocks help reduce computational complexity while maintaining representational power.

4.3 LSTM (Long Short-Term Memory)

LSTM networks are a class of a RNNs. This is designed to address the limitations of traditional RNNs in capturing long-term dependencies in sequential data. It is designed in such a way to overcome the problems and limitations of traditional RNNs in capturing and retaining long-term dependencies in sequential data. Designed as such, ResNet50 effectively mitigates the vanishing gradient problem, ensuring that the training model remains unchanged throughout the process. LSTM offer a diverse array of parameters which include learning rates, as well as input and output biases, providing flexibility and having control over the model's behavior and training dynamics.

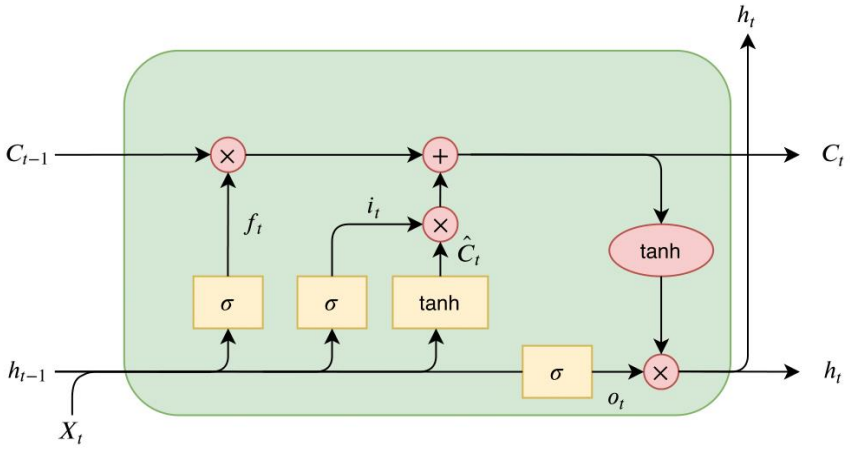


Fig 7. LSTM Model[13]

The image in Fig 7. represents the LSTM model architecture. In LSTM networks there are three types of gates which are namely forget gate, input gate, and output gate which helps in integrating to regulate the flow of data, facilitating effective memory management and learning. Various gates within LSTM networks regulate the flow of information to control the state of the cell. The forget gate plays an important role in determining which data from the previous cell state should be retained. The input gate is responsible for updating the cell state. The output gate helps in regulating the flow of information from the current cell state to the output.

5. EXPERIMENTAL RESULTS

5.1 Hardware Environment:

In our hardware environment, we operate on an Intel Core i7-12700 CPU running at 2.10GHz, providing substantial processing power for our computational tasks. Supported by 32.0 GB of RAM, our system boasts ample memory capacity, facilitating efficient data processing and manipulation. Additionally, with 2.47 GB of storage space, we ensure sufficient storage capacity to accommodate datasets, models, and other essential files necessary for our computing tasks. This hardware configuration enables us to execute computationally intensive operations, such as training deep learning models and processing large-scale datasets, with speed and reliability, enhancing the overall performance and productivity of our computational workflows.

5.2 Software Environment:

In our project, we leverage a suite of powerful Python libraries and frameworks to tackle computer vision tasks and implement DL models. OpenCV (cv2) serves as a cornerstone for image processing and manipulation, facilitating tasks such as image reading and preprocessing. NumPy complements this by providing essential support for array manipulation and mathematical operations, crucial for handling large, multi-dimensional data arrays efficiently. For ML tasks, we depend on Scikit-learn, leveraging its straightforward yet highly efficient tools for data preprocessing and splitting into training and testing sets. TensorFlow, developed by Google, this is a open-source DL framework widely utilized for various ML and

artificial intelligence tasks. Keras reduces the creation and training of neural networks with its user-friendly interface, seamlessly integrating with TensorFlow for smooth implementation.

Furthermore, we integrate pre-trained models like ResNet50V2, harnessing its image classification capabilities as a feature extractor in our TrustNet model. To handle sequential data, particularly in the context of sequence modeling tasks, we incorporate LSTM, a type of RNN architecture renowned for its effectiveness in capturing temporal dependencies and patterns. Together, these tools and technologies form a robust foundation for our computer vision and DL endeavors, empowering us to address complex tasks with efficiency and effectiveness.

Epochs denote the number of times the entire dataset is forwarded and backward through the neural network during the training phase. Each epoch encompasses one forward pass and one backward pass. In our trained model we specify the number of epochs is 5 means that the training process will iterate over the training dataset 5 times. Increasing the number of epochs can lead to improve the model's performance. Learning algorithms typically require hundreds or thousands of epochs to minimize the error in the model. The number of epochs can range from as low as ten to as high as 1000 or even more, depending on the complexity of the task and the convergence criteria.

5.3 Metrics:

In the ResNet50 architecture employed in our model, residual blocks constitute the backbone, featuring convolutional layers alongside shortcut connections. Expanding upon the foundational ResNet design, ResNet50 incorporates refinements like pre-activation residual units. These units integrate batch normalization and ReLU activation prior to each convolutional operation, thereby enhancing the efficiency of training. The mathematical representation of a single residual block in ResNet can be represented as:

$$H(X) = F(X) + X \qquad \text{Eq (1)}$$

As shown in equation (1) F(X) represents residual mapping, X represent input of the block, H(X) represent output of the block.

While the equations provided describe the mathematical operations of a generic LSTM unit, the TensorFlow library handles the implementation of these operations which internally performs the mathematical operations necessary for the LSTM to function.

Input Gate: $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$ Eq (2)

Forget Gate: $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$ Eq (3)

Output Gate: $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$ Eq (4)

Candidate Cell State: $\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$ Eq (5)

New Cell State: $c_t = f_t \,.\, c_{t-1} + i_t \,.\, \tilde{c}_t$ Eq (6)

Hidden State Output: $h_t = o_t \,.\, \tanh(c_t)$ Eq (7)

In the equation (2-7) described above, σ represents the sigmoid activation function, tanh denotes the hyperbolic tangent activation function.x_t represents the input at time step t, $h_t$is the hidden state at time step t, c_t is the cell state at time step t,i_t, f_t,o_t,$\tilde{c}_t$and respectively represent the input gate, forget gate, output gate, and candidate cell state at time step t. The W matrices and b vectors are the weight matrices and bias vectors, respectively, which are the learnable parameters of the LSTM layer.

5.4 Results and Discussions

S.No	Model	Accuracy	F1 Score	Logloss
1.	EfficientNet[8]	87%	-	0.4658
2.	VSS16 + InceptionV3 + XceptionNet[7]	96%	-	0.11140
3.	EfficientNet[10]	97.8%	91.9%	-
4.	VGG-19[6]	37%	26.8%	-
5.	Face X-Ray[5]	80.92%	-	-
6.	Our Proposed Model	93.4%	90.6%	-

Table 1. Accuracy, F1 score and Logloss values of different models

As shown in Table 1, the table gives a comparative study of acurracy, Fl score and Logloss values of different existing models and the proposed model.

6. CONCLUSION

Despite the progress made in deepfake detection, we have created a model architecture to find whether a video is real or fake. We have used ResNet50 and LSTM models to improve the efficiency of the project. By combining these models, we have achieved notable success in differentiating between genuine and manipulated videos. Moving forward, there are several avenues for further research and improvement in deepfake detection using ResNet50 and LSTM. Firstly, exploring ensemble approaches that combine multiple detection models could enhance robustness and generalization. Secondly, investigating the integration of other modalities, such as audio and text, could provide complementary information for more accurate detection. Additionally, developing techniques to mitigate the impact of adversarial attacks on detection models is crucial. Furthermore, addressing ethical considerations and privacy concerns associated with deepfake detection and deployment is essential for the responsible use of such technology.

REFERENCES:

[1] A. Raza, K. Munir, and M. Almutairi, "A Novel Deep Learning Approach for Deepfake Image Detection," *Appl. Sci. 2022, Vol. 12, Page 9820*, vol. 12, no. 19, p. 9820, Sep. 2022, doi: 10.3390/APP12199820.

[2] A. Lewis, P. Vu, R. M. Duch, and A. Chowdhury, "Deepfake detection with and without content warnings," *R. Soc. Open Sci.*, vol. 10, no. 11, Nov. 2023, doi: 10.1098/RSOS.231214.

[3] B. Dolhansky *et al.*, "The DeepFake Detection Challenge (DFDC) Dataset," Jun. 2020, Accessed: Apr. 04, 2024. [Online]. Available: https://arxiv.org/abs/2006.07397v4

[4] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2886–2895, Jan. 2020, doi: 10.1109/CVPR42600.2020.00296.

[5] L. Li *et al.*, "Face X-ray for More General Face Forgery Detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 5000–5009, Dec. 2019, doi: 10.1109/CVPR42600.2020.00505.

[6] B. Lamichhane, K. Thapa, and S. H. Yang, "Detection of Image Level Forgery with Various Constraints Using DFDC Full and Sample Datasets," *Sensors 2022, Vol. 22, Page 9121*, vol. 22, no. 23, p. 9121, Nov. 2022, doi: 10.3390/S22239121.

[7] S. A. Khan, A. Artusi, and H. Dai, "Adversarially robust deepfake media detection using fused convolutional neural network predictions," Feb. 2021, Accessed: Apr. 04, 2024. [Online]. Available:

https://arxiv.org/abs/2102.05950v1

[8] N. Bonettini, L. Bondi, E. D. Cannas, P. Bestagini, S. Mandelli, and S. Tubaro, "Video Face Manipulation Detection Through Ensemble of CNNs," *Proc. - Int. Conf. Pattern Recognit.*, pp. 5012–5019, Apr. 2020, doi: 10.1109/ICPR48806.2021.9412711.

[9] D. M. Montserrat *et al.*, "Deepfakes Detection with Automatic Face Weighting," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 2851–2859, Apr. 2020, doi: 10.1109/CVPRW50498.2020.00342.

[10] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Deepfake Detection Scheme Based on Vision Transformer and Distillation," Apr. 2021, Accessed: Apr. 04, 2024. [Online]. Available: https://arxiv.org/abs/2104.01353v1

[11] X. Li *et al.*, "Sharp Multiple Instance Learning for DeepFake Video Detection," *MM 2020 - Proc. 28th ACM Int. Conf. Multimed.*, pp. 1864–1872, Aug. 2020, doi: 10.1145/3394171.3414034.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, Dec. 2016, doi: 10.1109/CVPR.2016.90.

[13] "Long Short-Term Memory (LSTM). Long Short-Term Memory (LSTM) is a type… | by Saba Hesaraki | Medium." Accessed: Apr. 04, 2024. [Online]. Available: https://medium.com/@saba99/long-short-term-memory-lstm-fffc5eaebfdc