IOT-INTEGRATED DETECTION AND CLASSIFICATION OF DEEPFAKE IMAGES AND VIDEOS USING CUSTOM DEEP LEARNING MODELS

K. Praveen Kumar
School of Computer Application
+Karpagam college of Engineering
Coimbatore,Tamilnadu
praveenkk418@gmail.com

R. Ramprashath
Assistant Professor
School of Computer Application
Karpagam College of Engineering
Coimbatore,Tamilnadu
ramprashath.r@kce.ac.in

Abstract— Deep fakes are becoming more common; they include editing previously published films and photos to produce content that appears authentic but is wholly fake. The development process has been considerably expedited by the widespread availability of deep learning techniques, such as autoencoders, **Generative Adversarial Networks (GANs)**, and user-friendly software. These sophisticated algorithms adeptly fuse and modify visual and audio elements, facilitating the production of content that closely mimics genuine footage, even for those without specialized knowledge. The malicious manipulation of images and videos poses significant security and societal concerns. With an emphasis on facial alteration, the goal of this research is to create a deep learning system integrated with IoT for the detection and classification of deepfake images and videos. The dataset used for the project is either **Face Forensics++, Celeb-DF, or the Deepfake Detection Challenge Dataset (DFDC)**, available on Kaggle, consisting of real and deepfake images and videos. By utilizing Recurrent and Convolutional Neural Networks, we have made developments in deepfake detection. Commencing with preprocessing the data, extracting frames from the videos, and separating the dataset into training and validation sets, IoT devices continuously capture and transmit data for real-time analysis. For the detection and classification of deepfake images and videos, **OpenCV and Face Recognition** are employed for facial detection, while Convolutional Neural Networks (CNNs) are used by the system to extract features at the frame level. A Recurrent Neural Network (RNN) is trained using these features. Various techniques such as data augmentation, learning rate scheduling, and early stopping enhance model performance. This comprehensive IoT-integrated approach ensures accurate discrimination between authentic and deepfake content, addressing concerns regarding the integrity of digital media.

Keywords: Deepfake video Detection, Image Forgery Detection, IoT Integration, Custom Deep Learning Models, IoT-based Deepfake Detection.

I. INTRODUCTION

The widespread use of smartphones with advanced cameras and easy access to high-speed internet has made it effortless to create and share digital videos on social media. Concurrently, the rapid advancements in computational capabilities have empowered deep learning techniques, notably Generative Adversarial Networks (GANs) and autoencoders, enabling the creation of deepfakes—artificially manipulated videos and audio clips. The proliferation of deepfakes across social media platforms has engendered significant concerns, precipitating the dissemination of misinformation and posing grave threats to societal integrity. [2]

Recognizing the imperative to counteract the proliferation of deepfakes, this project endeavors to develop a robust deep-learning model specialized in detecting and categorizing deepfake images and videos, with a particular emphasis on identifying facial manipulations. Leveraging datasets such as Face Forensics++, Celeb-DF, or the Deepfake Detection Challenge Dataset (DFDC), encompassing a diverse array of real and manipulated media, forms the cornerstone of this endeavor.

Our approach to detecting deepfakes is grounded in understanding the underlying mechanisms of Generative Adversarial Networks (GANs), which are pivotal in generating such content. GANs operate by substituting faces within videos through a process of frame-by-frame manipulation, typically employing autoencoders for reconstruction. By analyzing artifacts created during the face substitution process, our system makes use of these subtleties and exploits the resolution discrepancies between the modified facial areas and their surrounding context. The core of our detection methodology is based on the use of CNNs, LSTM cells to capture temporal inconsistencies between frames.

Through a thorough analysis and comparison of the generated facial regions with their surrounding environment, our method seeks to identify minute but noticeable differences suggestive of deepfake manipulation. Augmented with techniques such as data augmentation, learning rate scheduling, and early stopping, our comprehensive methodology endeavors to enhance model performance, ensuring accurate discrimination between authentic and deepfake content. Through these concerted efforts, we strive to mitigate the pervasive threat posed by deepfake media, safeguarding the integrity of digital discourse and societal trust. [4]

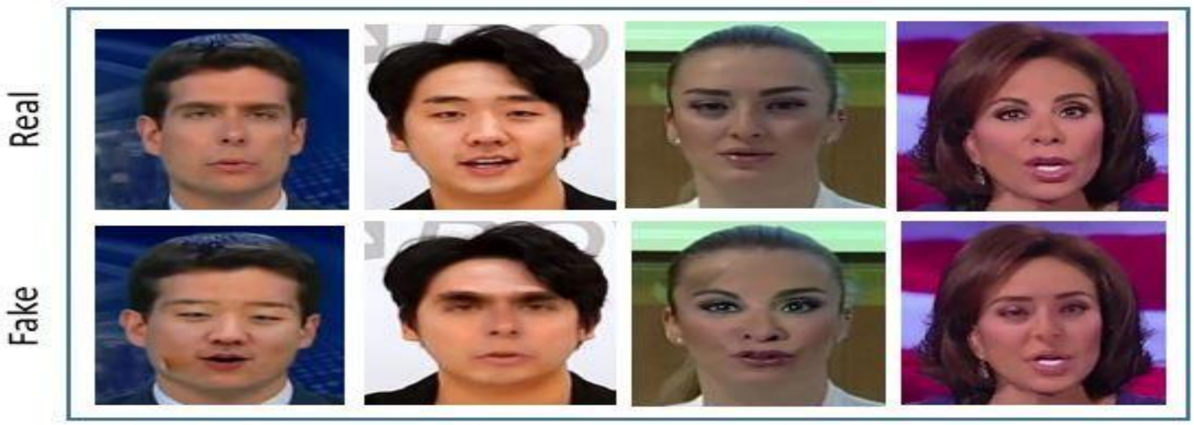


Fig 1: Samples from the FaceForensics++ dataset depicting both unaltered and manipulated facial images.

II.LITERATURE SURVEY

The values of democracy, fairness, and public confidence are seriously threatened by the deepfake videos' explosive growth and illegal usage. The mitigation is therefore rising. In the subject of deepfake detection, several strategies that concentrate on different facets of the phenomenon have been put forth.

For instance, "ExposingDF Videos by Detecting Face Warping Artifacts" [1] presented a technique to detect artefacts in deepfake videos by utilising a specialised Convolutional Neural Network (CNN) model to compare the generated facial portions with their surrounding regions. The study highlighted the presence of distinct facial artifacts within deepfake content, underscoring the limitations of current deepfake algorithms in generating images with constrained resolutions. These images require additional transformations to align with the faces being replaced in the original video, thereby creating identifiable patterns that can be exploited for detection purposes.

In "Exposing AI-Created Fake Videos by Detecting Eye Blinking" [2], A false face videos made with deep neural network models is suggested. The method depends on the ability to identify eye blinking in the videos a physiological signal that is not well replicated in artificially created videos. The method shows promising performance in recognising movies generated by Deep Neural network-based software for generating fake content through evaluation against benchmarks of eye-blinking detection datasets.

While their approach solely relies on the absence of blinking as an indicator for detection, it's imperative to account for additional parameters such as teeth alignment, facial wrinkles, and others to effectively identify deepfake content. Our proposed method encompasses the consideration of all these parameters to enhance detection accuracy.

"Using capsule networks to detect forged images and videos" [3] employs a technique leveraging capsule networks to identify manipulated images and videos across various contexts, including replay attack detection and the detection of computer-generated videos.

In their approach, the utilization of random noise during the training phase is considered suboptimal. While their model demonstrated effectiveness within their dataset, When used on real-time data, its effectiveness could be reduced since noise is incorporated into the training process. On the other hand, our approach is made to be trained on real-time and noiseless datasets in order to guarantee reliable results in real-world scenarios.

The approach outlined in "Detection of Synthetic Portrait Videos using Biological Signals" [5entails obtaining biological signals from face areas in video pairings of both real besides phoney portraits. By transforming, these signals compute temporal consistency and spatial coherence, which are then captured in feature sets and PPG maps. The next step is to train a CNN and a probabilistic SVM to combine authenticity probabilities and identify if the video is real or false.

On the other hand, regardless of the video's quality, resolution, or source, "Fake Catcher" detects fraudulent content with excellent accuracy. Nevertheless, the loss of biological signal preservation occurs from its lack of a discriminator. It is problematic to formulate loss function that is consistent with the suggested signal processes.

III. PROPOSED SYSTEM

Numerous tools exist for generating deepfake (DF) content, yet reliable detection tools for both videos and images are scarce. The proposed approach to DF detection, encompassing both images and videos, represents a significant contribution aimed at curtailing DF dissemination online. Envisioned is a user-friendly web-based platform enabling the upload of both images and videos for classification as fake or real. Scalability of this research could lead to the development of a browser plugin for automatic DF detection. Furthermore, this method might be integrated by well-known apps like Facebook and WhatsApp for user convenience. A major goal is to assess user acceptance and performance in terms of security, correctness, dependability, and ease of use. The approach focuses on three different forms of DF: replacement, retrenchment, and interpersonal. The architecture of the scheme is seen in Figure 2.

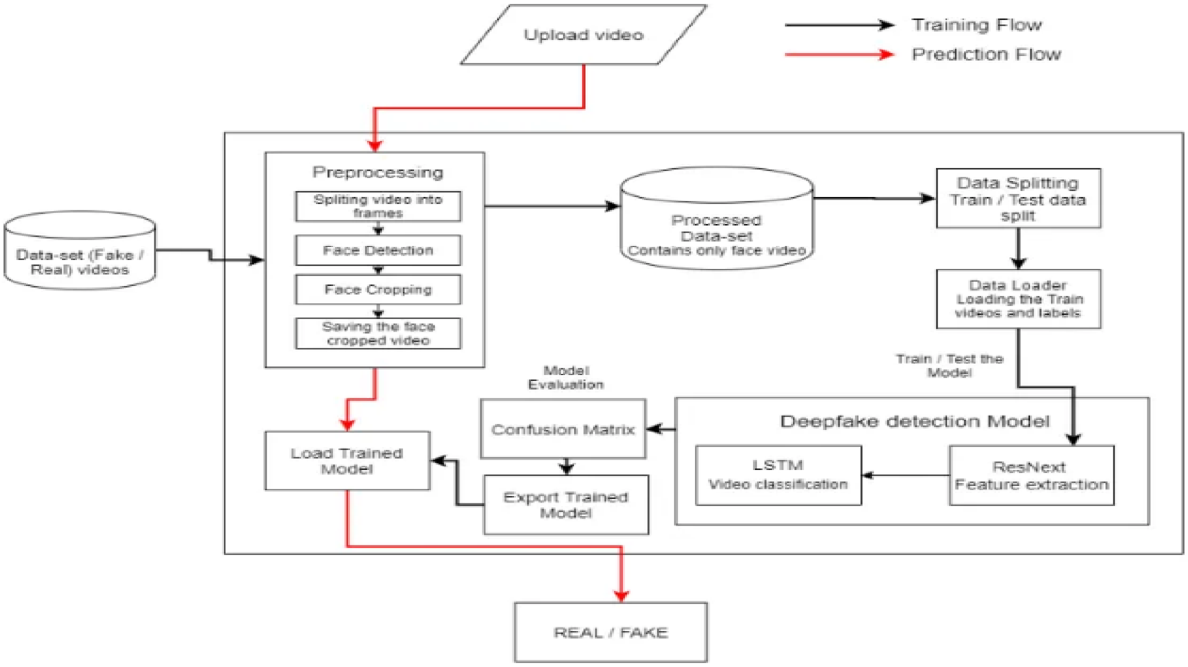

Upload video
Training Flow
Prediction Flow
Preprocessing
Spliting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Data-set (Fake / Real) videos
Processed Data-set Contains only face video
Data Splitting Train / Test data split
Data Loader Loading the Train videos and labels
Train / Test the Model
Model Evaluation
Confusion Matrix
Deepfake detection Model
LSTM Video classification
ResNext Feature extraction
Load Trained Model
Export Trained Model
REAL / FAKE

Fig. 2: Scheme Architecture

A. Dataset:

Our dataset is a blend sourced from various repositories, including CelebDF for images and DFDC for video detection, complemented by FaceForensics++ for validation. We've meticulously curated this dataset to feature an equal distribution of original and manipulated content, with 70% allocated for training and 30% for testing.

B. Preprocessing:

First, videos must be segmented into frames. Next, face detection and cropping must be done. To ensure uniformity in frame count, we calculate the mean frames per video and create a new dataset with frames matching this mean. Frames lacking detected faces are excluded. Due to computational constraints, we propose using only the first 100 frames of each 10-second video for training purposes.

C. Model:

ResNext50_32x4d makes up our model architecture, which is followed by an LSTM layer. The preprocessed videos are loaded and divided into sets by the Data Loader. Frames from videos that have been analysed are then supplied into the model in small batches for testing and training.

D. CNN Layers for Fake Image Detection:

For detecting fake images, our model employs multiple CNN layers. These layers are specifically designed to analyze image data and extract intricate features indicative of manipulation or authenticity. Figure 3 shows the structure of CNN layers for Fake Image Detection.

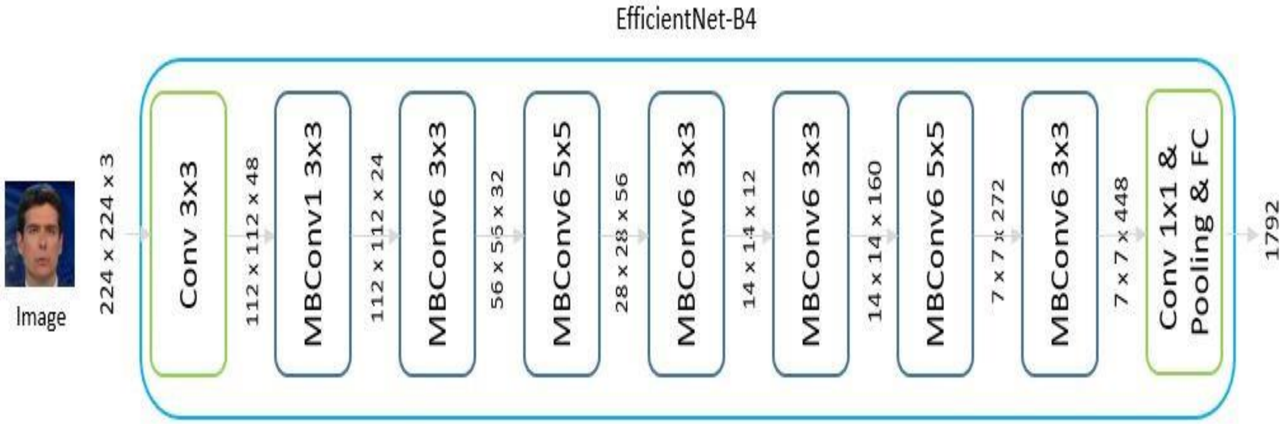


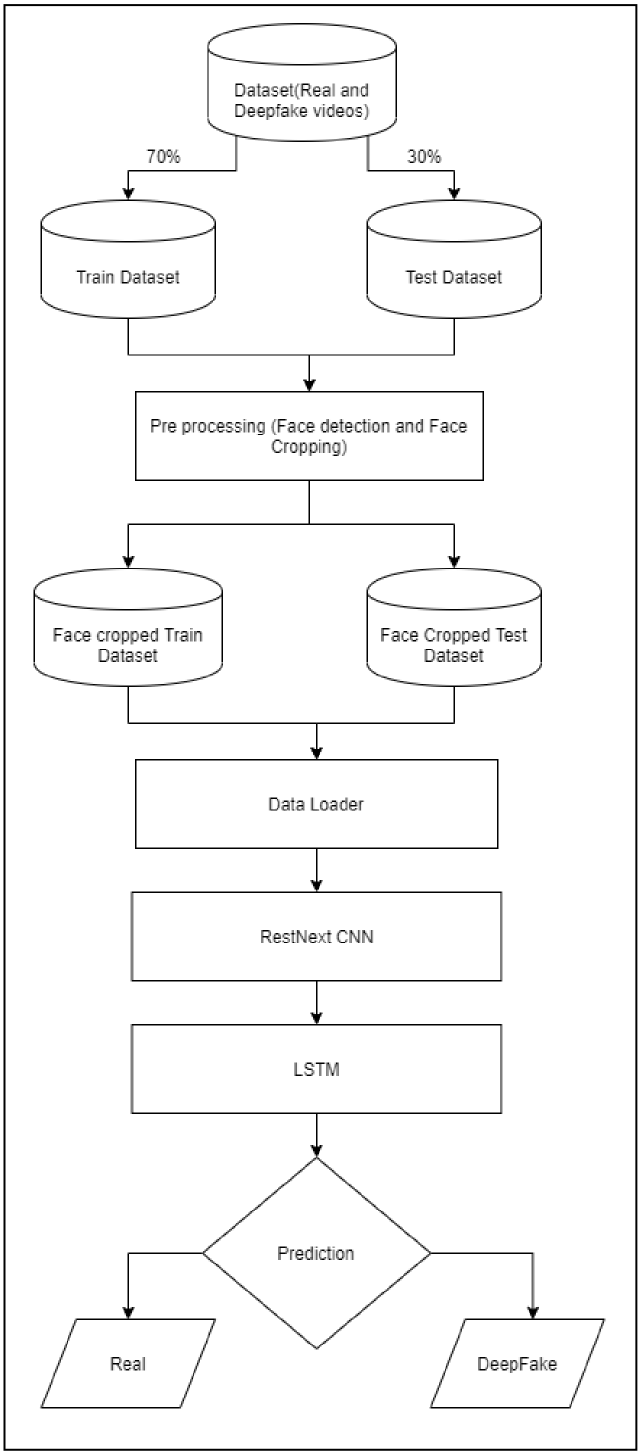
Fig. 3: Fake Image DetectionCNN layers

E. Preprocessing with ResNext CNN and Detection with LSTM for Video:

Video detection, on the other hand, follows a different approach. We first preprocess the videos using ResNext CNN, which LSTM network is used to carry out the real detection. This combination allows for a comprehensive analysis of temporal patterns within the videos, aiding in the accurate identification of deepfake content.

F. Prediction:

A fresh video is given to the trained algorithm for forecasting. In addition, the format of the learnt model is imported by preprocessing a new video. The faces in the divided video are cropped once it has been divided into frames, and instead of being kept locally, the cropped frames are transmitted directly to the trained model for detection. Figure 5 shows the prediction procedure.


Dataset(Real and Deepfake videos)
70%
30%
Train Dataset
Test Dataset
Pre processing (Face detection and Face Cropping)
Face cropped Train Dataset
Face Cropped Test Dataset
Data Loader
RestNext CNN
LSTM
Prediction
Real
DeepFake

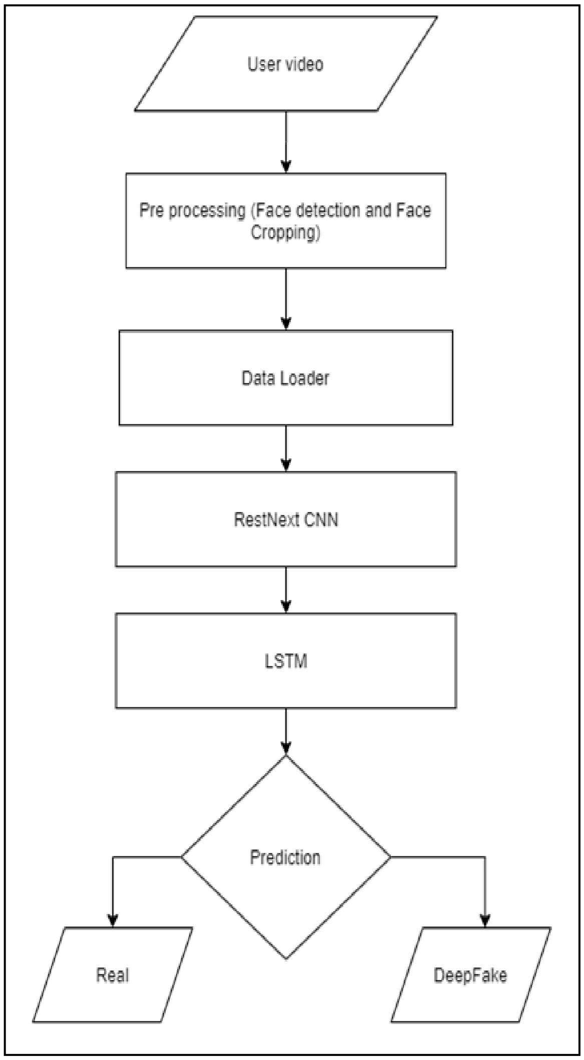


Fig. 5: Predictions Flow

Fig. 4: Training Flow

IV. RESULT

A. Training Progress:

The project's training progress is demonstrated in Figure 6, depicting four (4) distinct configurations:

— Configuration 1: Utilizing the ResNext perfect and freezing all convolution parameters.

— Configuration 2: Utilizing ResNext model and retraining the last 8 MB-ConvBlocks with the last two layers (convolution layer and classifier).

— Configuration 3: Utilizing the ResNext model and retraining the last 16 MB-ConvBlocks with the last two layers (convolution layer and classifier).

— Configuration 4: Utilizing the pre-trained ResNext model and integrating a Spatial Transformer Network after MBConvBlock 20.

B. Performance Analysis:

Here's a comparative summary illustrating the performance across different configurations:

Modèle	**Acc(train)**	**Acc(valid)**	**Acc(test)**
Configuration 1	83.64	83.64	83.44
Configuration 2	89.69	87.84	86.94
Configuration 3	92.99	90.34	90.08
Configuration 4	92.05	89.67	91.10

TABLE I: Comparative Analysis of Results

C. Attention Mechanism Insights:

The results of the LSTM model variant incorporating the attention mechanism (Fig. 8) are not included here; this model was solely utilized to enhance our understanding of the attention mechanism.

We analyzed the map calculated on the faces of our dataset. We selected the output of the Sigmoid layer within the attention block (Fig. 3), forming a 2D map sized 32 x 32. This map was then upscaled to align with the dimensions of the input face (128 x 128) and overlaid onto it.

Based on observations from Fig. 8, it appears that this simple attention mechanism favours important face features as the mouth, nose, ears, and eyes. Conversely, level areas (with weak gradients) contribute minimally to the network. Notably, artifacts from deepfake generation methods primarily cluster around facial attributes, aligning convincingly with real-world scenarios.

D. Video Detection Process:

In the video detection process (Fig. 4), we initiated a preprocessing step to extract frames from the videos, followed by face detection and cropping. These frames were subsequently fed into a ResNext CNN model for feature extraction. Afterward, a LSTM network was employed for sequence processing to analyze the temporal aspects of the video. This comprehensive approach ensured the accurate detection of deepfake content in videos.

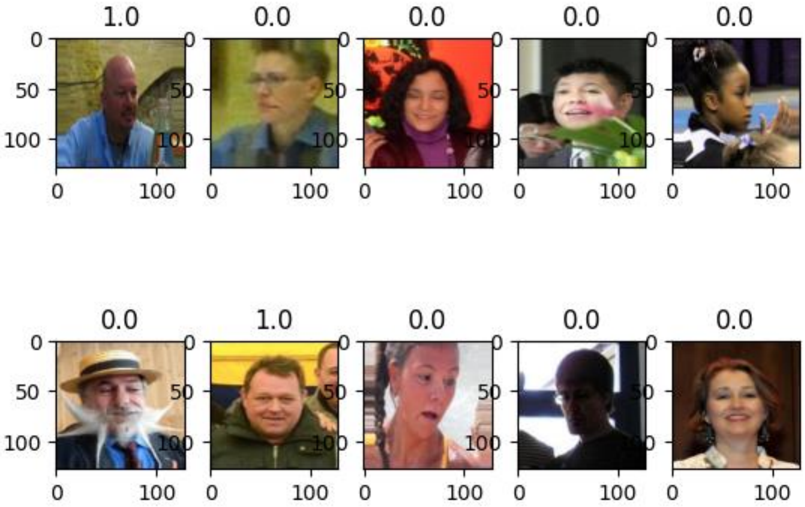


Fig. 8: Attention Mechanism

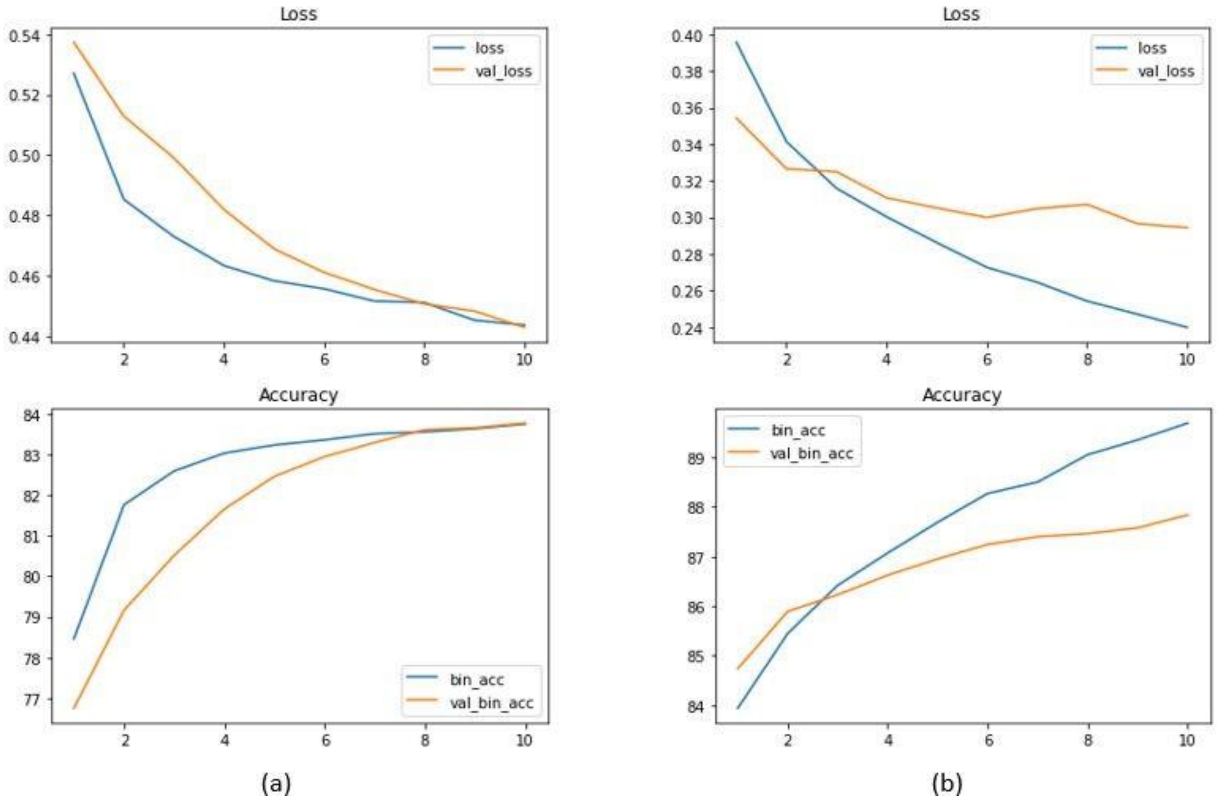
The representation's production will indicate whether the images or video is authentic or a deepfake, as well as the model's confidence level.

V. DISCUSSION OF RESULTS

In this section, we delve into the outcomes presented earlier, discussing the encountered challenges, the applied solutions, the anticipated improvements, and the research avenues identified throughout this project.

A. Interpretation of Results

Let's highlight the significant differences among our various model configurations:



(a)

(b)

Fig. 6: Training History of Configurations 1 (a) and 2(b)

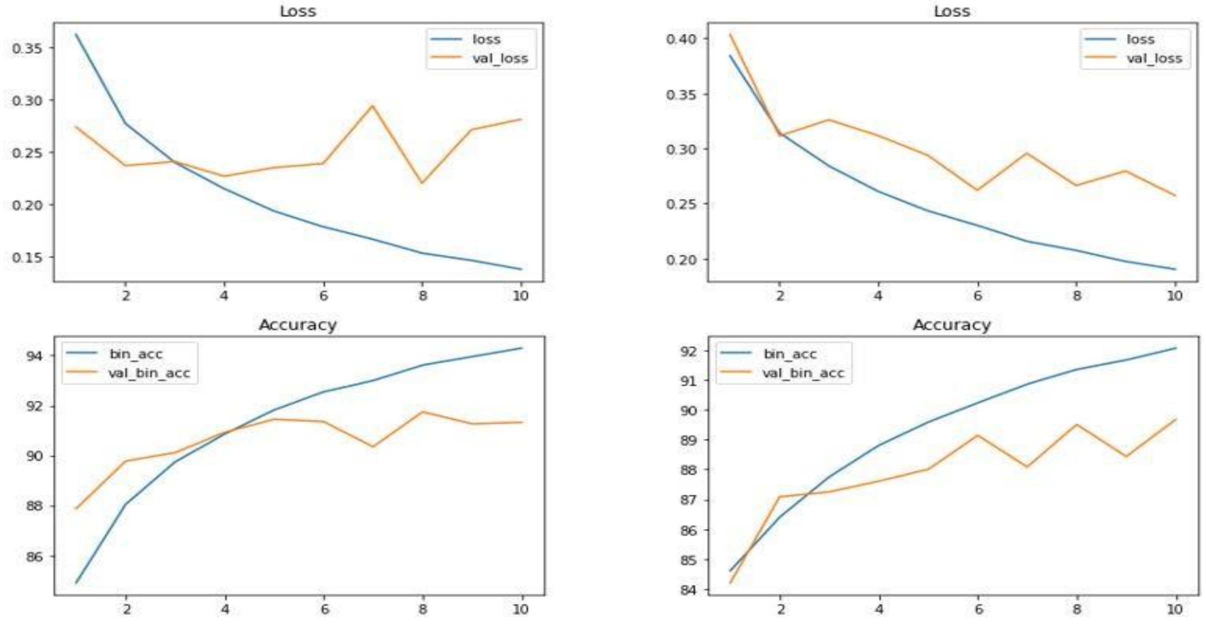


Fig. 7: Training History of Configurations 3 (a) and 4 (b)

— Configuration 1: Freezing all convolution parameters while utilizing the pre-trained model ensures the preservation of robust features learned by the ResNext model.

— Configurations 2 and 3: Freezing convolution parameters in specific blocks alongside the classifier involves training unfrozen layers on new images, particularly the higher layers. This approach prevents the model from memorizing differences among real and fake faces by encouraging it to seek useful artifacts for fake face finding.

— Configuration 4: The integration of a Spatial Transformer Network after MBConvBlock 20 applies spatial transformation to features during forward propagation, enhancing model performance.

Overall, our models perform well, with observed precision gains using the Spatial Transformer Network in the last configuration. However, further training and hyperparameter adjustments could lead to even better performance.

B. Encountered and Resolved Issues

Initially, accessing the dataset storage from Google Colab posed a challenge due to storage limitations. Additionally, dataset preprocessing proved time-consuming, requiring careful consideration to ensure code generality for future challenges in deepfake video detection. Face detection technology evaluation led us to choose Blazeface from multiple options.

Overfitting during initial experiments was a concern, which we addressed by adjusting the number of images extracted per video to 15, striking a balance to mitigate overfitting.

C. Improvements

One improvement we plan to implement is ensemble methods, combining predictions from different classifiers based on CNNs to enhance stability and performance. Additionally, combining our approach with the temporal pipeline method proposed by authors in [10] could further enhance detection accuracy.

D. Discussions and Research Perspectives

The rising quality of deepfakes necessitates improved detection methods. Deepfake Challenge Dataset and other reference datasets are regularly updated to support the development and validation of detection techniques, especially deep learning-based ones. Taking into account the difficulties presented by adversarial environments, future research should concentrate on introducing more reliable, scalable, and generalificable techniques.

An intriguing research direction is integrating detection methods into social media platforms to combat the widespread impact of deepfakes effectively. This integration, coupled with digital watermarking and potentially blockchain technology, could provide immutable authenticity details for multimedia content, addressing the growing concerns surrounding deepfake proliferation. Despite its effectiveness in various domains, minimal research exists on deepfake detection leveraging blockchain technology.

VI. CONCLUSION

Our project introduces a neural network approach to differentiate between real and fake images and videos, along with the confidence of the proposed model. We were inspired by how deepfakes are made using GANs with the assistance of Autoencoders. Our method analyzes each frame of the content using ResNext CNN for images and RNN with LSTM for videos. By integrating IoT devices for continuous data collection and real-time analysis, our model can process data from various sources dynamically. By considering specific parameters, our model accurately identifies whether the content is authentic or manipulated. We expect our IoT-integrated method to perform well when applied to real-world data in real-time scenarios.

VII. LIMITATIONS

Our project is that we have not taken audio into account. Therefore, our method is unable to detect audio deepfakes. We plan, however, to address this limitation in future research to achieve the discovery of audio, bottomless fakes!!!

VIII.REFERENCES

[1] Yuezun Li, Ming-Ching Chang and Siwei Lyu “Exposing AI Created Fake Videos by Detecting Eye Blinking” in arxiv.

[2] Huy H. Nguyen , Junichi Yamagishi, and Isao Echizen“ Using capsule networks to detect forged images and videos ”.

[3] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari and Weipeng Xu “Deep Video Portraits” in arXiv:1901.02212v2.

[4] Umur Aybars Ciftci, ˙Ilke Demir, Lijun Yin “Detection of Synthetic Portrait Videos using Biological Signals” in arXiv:1901.02212v2.

[5] A. M. Almars, “Deepfakes Detection Techniques Using Deep Learning: A Survey,” *J. Comput. Commun.*, 2021, doi:10.4236/jcc.2021.95003.

[6] S. Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-Generated Images Are Surprisingly Easy to Spot.. For Now,” 2020,
doi: 10.1109/CVPR42600.2020.00872.

[7] J. C. Dheeraj, K. Nandakumar, A. V. Aditya, B. S. Chethan, and G. C. R. Kartheek, “Detecting Deepfakes Using Deep Learning,” 2021,doi:10.1109/RTEICT52294.2021.9573740.

[8] M. Li, B. Liu, Y. Hu, and Y. Wang, “Exposing deepfake videos by tracking eye movements,” 2020, doi:
10.1109/ICPR48806.2021.9413139.

[9] Guera, 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) IEEE.

[10] Y. Al-Dhabi and S. Zhang, “Deepfake Video Detection by Combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN),” 2021, doi: 10.1109/CSAIEE54046.2021.9543264.

[11] A. Badale, L. Castelino, and J. Gomes, “Deep Fake Detection using Neural Networks,” vol. 9, no. 3, pp. 349–354, 2021.

[12] Y. Li, M. C. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI created fake videos by detecting eye blinking,” 2019, doi:
10.1109/WIFS.2018.8630787.

[13] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting world leaders against deep fakes,” 2019.

[14] M. Nagao, “Natural language processing and knowledge,” in *2005 International Conference on Natural Language Processing and Knowledge Engineering*, 2005, pp. 1-, doi: 10.1109/NLPKE.2005.1598694.

[15] G. Jaiswal, “Hybrid Recurrent Deep Learning Model for DeepFake Video Detection,” in *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2021, pp. 1–5, doi: 10.1109/UPCON52273.2021.9667632.

[16] Long Short-Term Memory: From Zero to Hero with Pytorch: https://blog.floydhub.com/long-short-term-memory-from-zero-to-hero-with-pytorch/

[18] Tiago de Freitas Pereira, Andr´e Anjos, Jos´e Mario De Martino, and S´ebastien Marcel, “Can face anti spoofing countermeasures work in a real world scenario?,”in ICB. IEEE, 2013.

[19] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, “Distinguishing computer graphics from natural images using convolution neural networks,” in WIFS. IEEE, 2017.

[20] F. Song, X. Tan, X. Liu, and S. Chen, “Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients,” Pattern Recognition, vol. 47, no. 9, pp. 2825–2838, 2014.