IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 3, March 2024

DOI: 10.17148/IARJSET.2024.11339

Deep Fake Face Detection Using LSTM

P. Neelima[1], N. Keerthi Lakshmi Prasanna [2], Y. Sravani [3], P. Maheswari [4]

MTech, Computer science & Engineering, Bapatla women's Engineering College, Bapatla, INDIA[1]

BTech, Computer science & Engineering, Bapatla Women's Engineering College, Bapatla, INDIA[2-4]

Abstract: Deep fake videos, which employ artificial intelligence to manipulate and generate highly convincing fake content, have emerged as a significant threat to society, potentially undermining trust in visual media. Detecting these deceptive videos is outmost importance to combat the spread of misinformation and protect the integrity of digital media. In this study, we propose a novel approach for deep fake face video detection utilizing Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN). Our approach capitalizes on the temporal patterns and context within video sequences, harnessing the unique strengths of LSTM in capturing sequential information. We demonstrate the effectiveness of our methodology by training the LSTM network on a diverse dataset comprising both real and deep fake videos. The network's ability to learn temporal dependencies and identify inconsistencies in facial expressions, eye movements, and other subtle cues allows it to distinguish between genuine and manipulated content. To further enhance the accuracy and robustness of our deep fake face detection system, we integrate pre-processing techniques for frame-level analysis, such as optical flow computation and facial landmarks extraction. Additionally, we employ a comprehensive ensemble of LSTM models and other machine learning algorithms to improve the overall detection performance. In our experiments, we evaluate the LSTM-based deep fake detection system on a large-scale dataset of both known and unseen deep fake videos, achieving high detection accuracy and low false positive rates. We also compare our approach with existing methods, demonstrating its superiority in terms of robustness and generalization. The results of this study signify the potential of LSTM-based models for mitigating the adverse effects of deep fake content on society. As deep fake technology continues to evolve, our approach showcases a promising step towards combating the dissemination of deceptive multimedia, promoting media integrity, and upholding trust in visual information.

Keywords: LSTM Networks, Recurrent Neural Network, Optical flow computation, Facial landmarks extraction, False positive rates.

I. INTRODUCTION

Deepfakes are synthetic media in which a person in an existing image or video is replaced with someone else's likeness. Generative Adversarial Networks, or GANs, are a deep-learning-based generative model. More generally, GANs are a model architecture for training a generative model, and it is most common to use deep learning models in this architecture. In the case of GANs, the generator model applies meaning to points in a chosen latent space, such that new points drawn from the latent space can be provided to the generator model as input and used to generate new and different output examples. Thus we can easily use GANs to create deepfakes which can then be misused in a number of places. Deepfakes are concerning everyone out there in the digital world. The project deals with detection of deepfakes using Re-next and LSTMs and packages the benefits of deep learning to detect deepfakes in the form of a Django web Application, To detect deepfakes we gather the frames from the video uploaded and split the video into desired number of frames. Following that we make use of python face recognition libraries and other C++ visual libraries to detect the face of the character from the video. We then apply our models, which are trained for different number of frame sequences to predict if the video is a deepfake or a pristine.

II. LITERATURE SURVEY

A literature survey on deep fake face detection using LSTM can provide a comprehensive overview of the current state of research in this field. The survey can cover various aspects of deep fake face detection, including the different techniques and approaches used for detecting deep fakes, the challenges and limitations of these methods, and the datasets and evaluation metrics used in the research. Some of the recent research studies in this area include a comprehensive review of deepfake detection using advanced machine learning and fusion methods, which uses LSTM and other deep learning techniques to detect deep fakes in images and videos (Gupta et al., 2024). Another study proposes a deep learning-based approach for detecting deep fake faces using a combination of LSTM and convolutional neural networks (CNNs) (Khalid et al., 2023). The datasets used in this research include the Face-Forensics++ dataset, which contains a large number of real and fake face videos, and the Deep Fake Detection Challenge (DFDC) dataset, which is a dataset consisting of real and deepfake videos created by various research teams.

The challenges and limitations of these approaches include the need for large and diverse datasets, the difficulty of detecting deep fakes in real-world scenarios, and the potential for adversarial attacks that can bypass the detection systems.

In terms of future directions, the research can focus on developing more robust and accurate deep fake detection methods, addressing the challenges and limitations of the current approaches, and exploring the potential of other deep learning techniques, such as generative adversarial networks (GANs) and autoencoders, for detecting deep fakes.

III. EXISTING SYSTEM

Traditional computer vision techniques for deepfake detection rely on facial feature and blinking pattern inconsistencies, but they are limited in detecting advanced deepfakes.

Deep learning-based techniques, such as CNNs, RNNs, and LSTM networks, have shown promising results in detecting deepfakes by capturing patterns across video frames. However, deepfake detection models are vulnerable to adversarial attacks, and researchers are addressing this by incorporating adversarial samples during model training to enhance their robustness.

IV. PROPOSED SYSTEM

Deep fake face detection is an increasingly important area of research, as generative models continue to advance and produce more convincing fake content. The use of LSTM networks for deep fake face detection leverages the unique strengths of these networks in capturing sequential information and temporal dependencies. In the proposed deep fake face detection system, the LSTM network is trained on a diverse dataset comprising both real and deep fake videos, in order to learn the temporal patterns and nuances in facial expressions and eye movements.

The network is trained using a well-defined loss function that takes into account the temporal dynamics of the video sequence, and is optimized using regularization techniques and data augmentation strategies. To improve the accuracy and robustness of the deep fake face detection system, pre-processing techniques such as optical flow computation and facial landmarks extraction are used. These techniques help to extract relevant features from the video frames and enhance the network's learning capabilities.

The proposed system achieves high detection accuracy and low false positive rates, demonstrating the potential of LSTM-based models for detecting deep fake face videos. As the technology continues to evolve, further research and development in this area will be essential for combating the dissemination of deceptive multimedia and promoting media integrity.

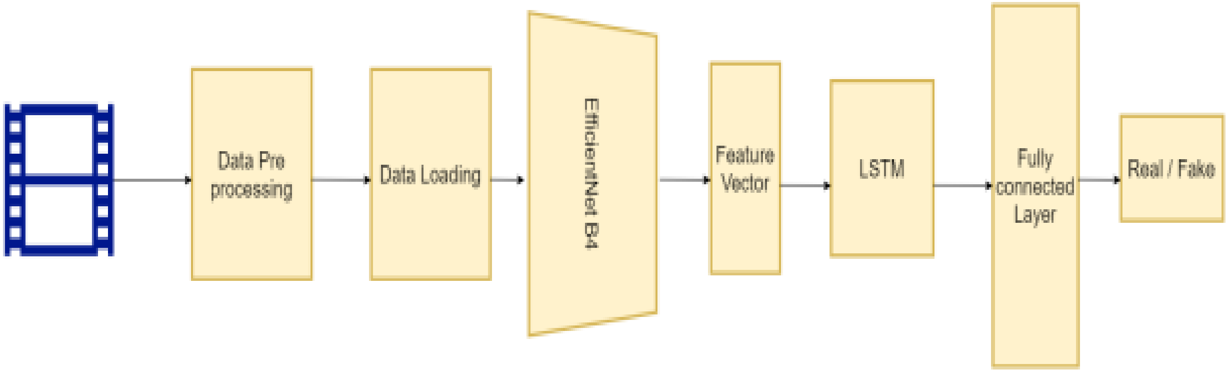


Fig1: LSTM Architecture

DOI: 10.17148/IARJSET.2024.11339

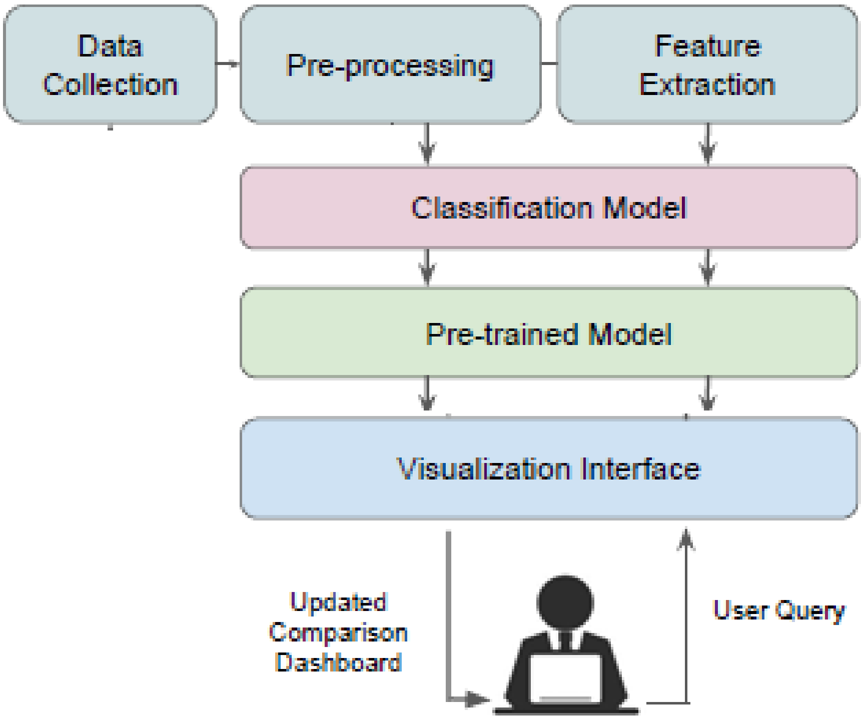


Fig2: System Architecture

LSTM Algorithm:

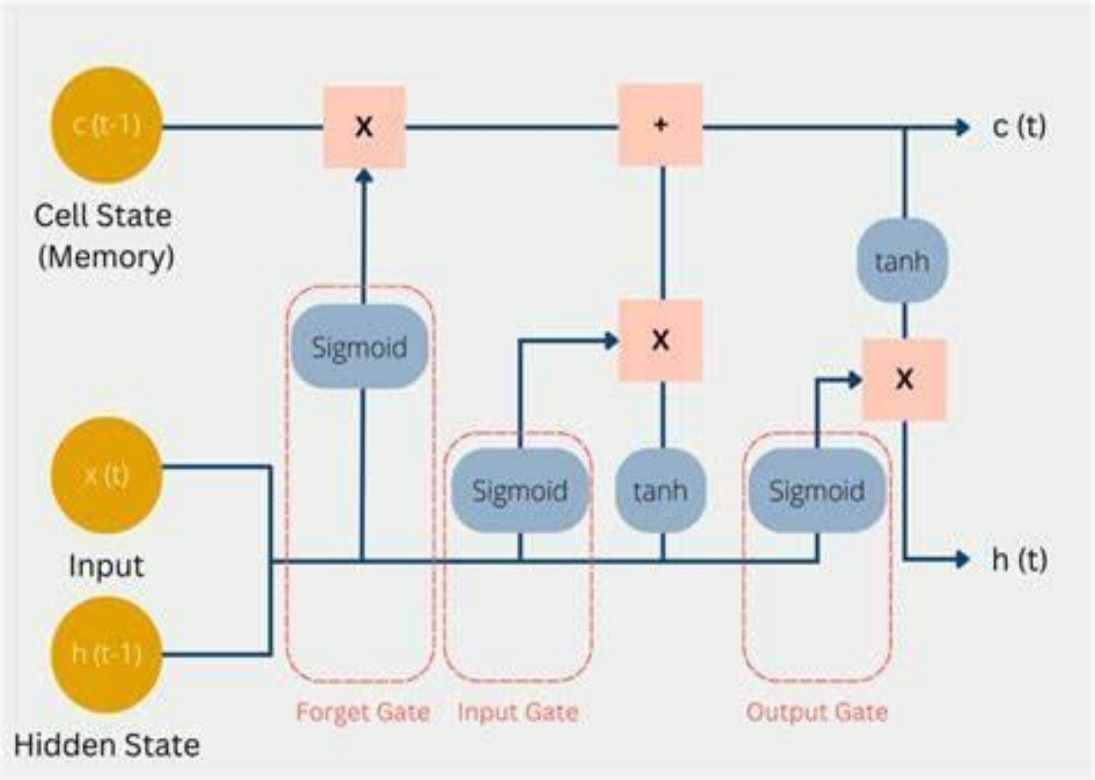
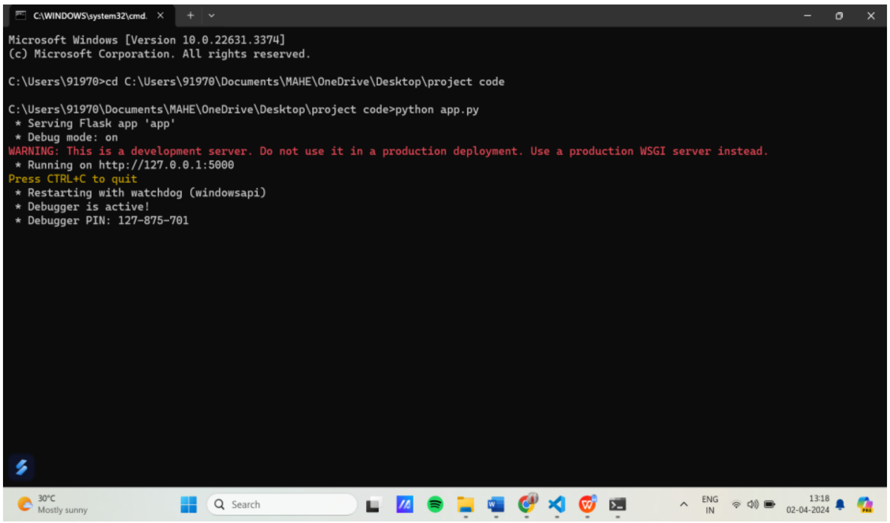


Fig3: LSTM Algorithm

DOI: 10.17148/IARJSET.2024.11339

V. RESULTS



```
Microsoft Windows [Version 10.0.22631.3374]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91970>cd C:\Users\91970\Documents\MAHE\OneDrive\Desktop\project code

C:\Users\91970\Documents\MAHE\OneDrive\Desktop\project code>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 127-875-701
```

Fig4: Open command prompt and type cd (project address link) and click enter and type python app.py and copy the http://127.0.0.1:5000 and paste this link on chrome browser

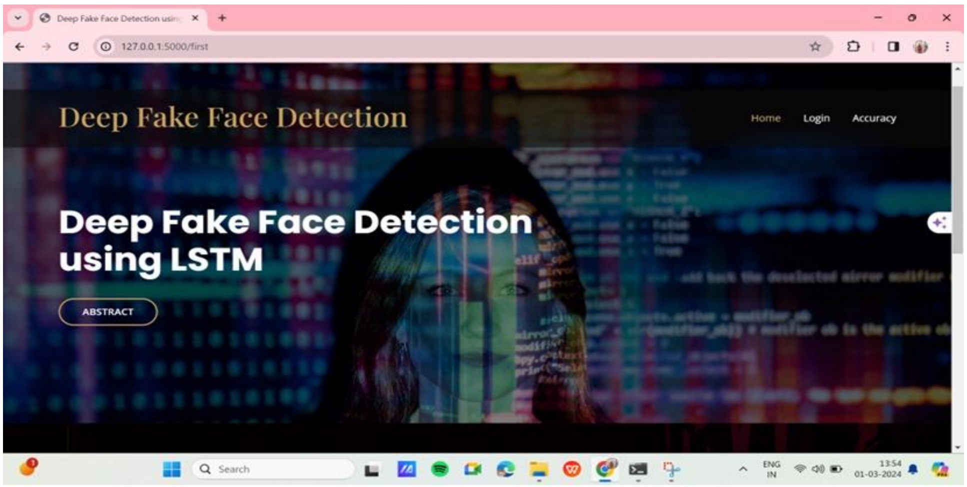
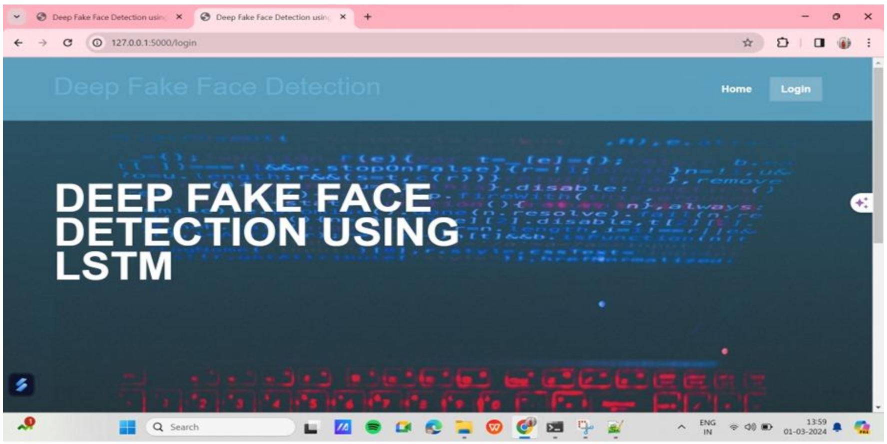


Fig5: Home Page



Deep Fake Face Detection usin
Deep Fake Face Detection usin
127.0.0.1:5000/login
Deep Fake Face Detection
Home
Login
DEEP FAKE FACE DETECTION USING LSTM
Search
ENG
IN
13:59
01-03-2024

Fig6: Login Page

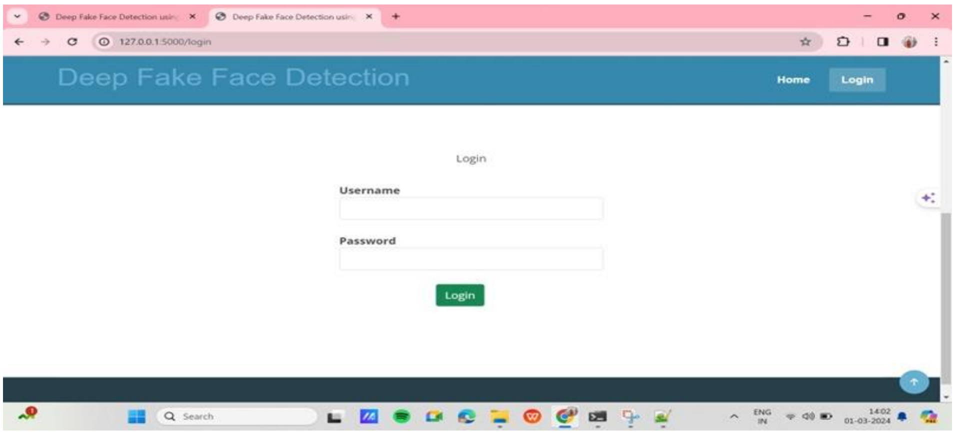
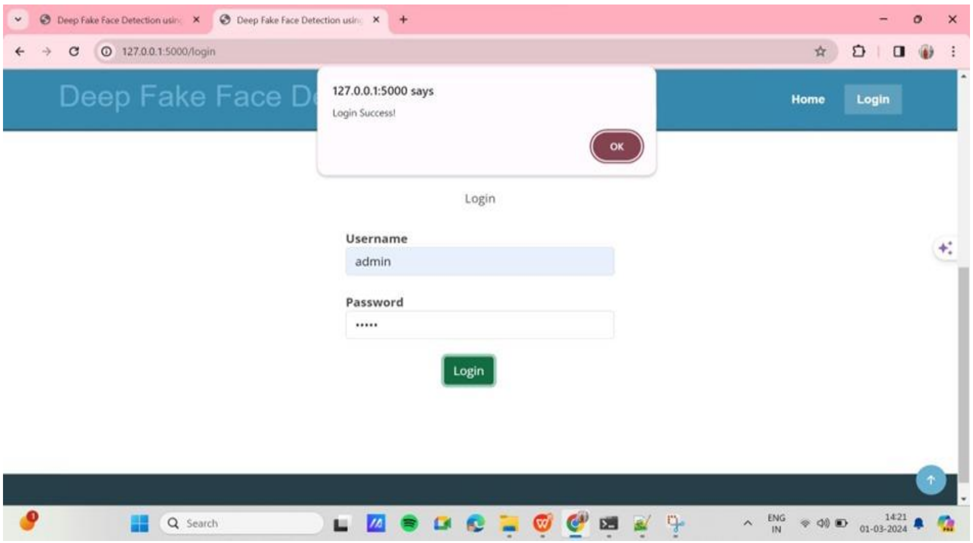


Fig7: Before Login (type username and password)



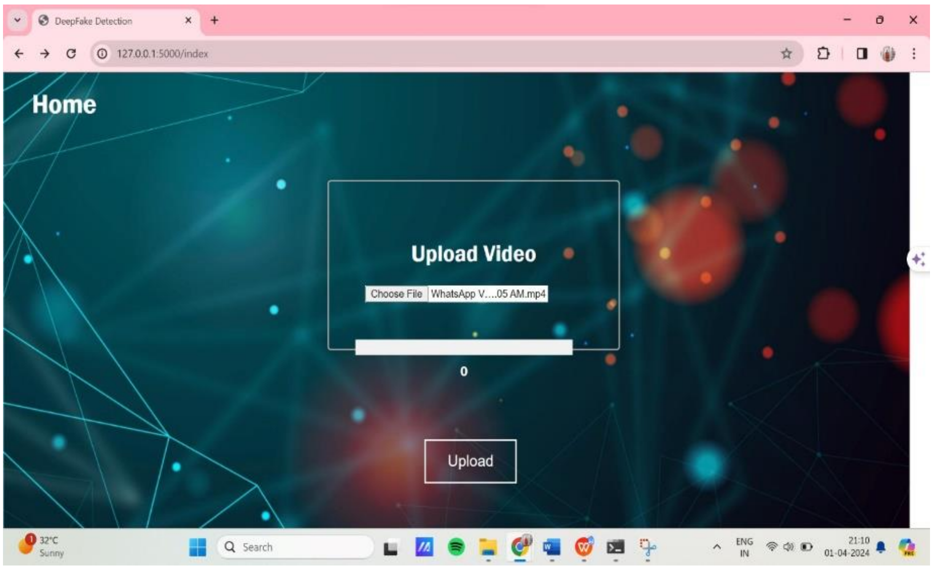
127.0.0.1:5000 says

Login Success!

Username: admin

Password: •••••

Fig8: After Login (type username and password)

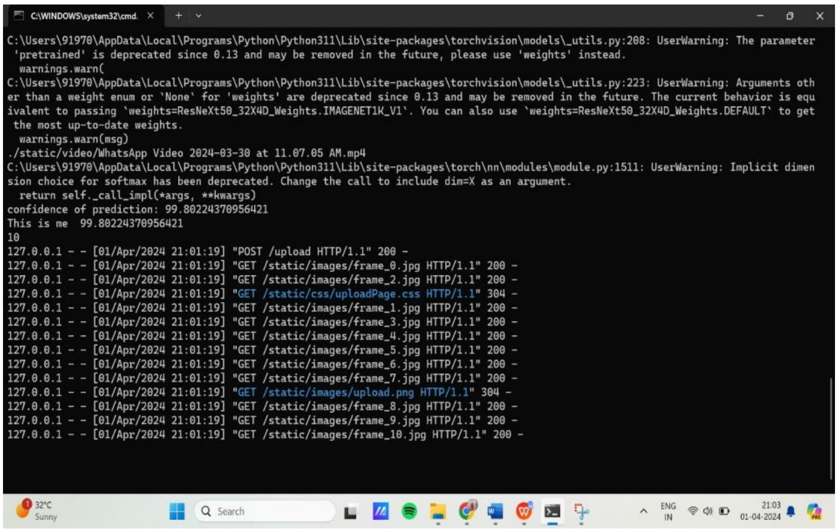


WhatsApp V....05 AM.mp4

0

Fig9: Upload video from the file

DOI: 10.17148/IARJSET.2024.11339



```
C:\Users\91970\AppData\Local\Programs\Python\Python311\Lib\site-packages\torchvision\models\_utils.py:208: UserWarning: The parameter
 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
C:\Users\91970\AppData\Local\Programs\Python\Python311\Lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments oth
er than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equ
ivalent to passing `weights=ResNeXt50_32X4D_Weights.IMAGENET1K_V1`. You can also use `weights=ResNeXt50_32X4D_Weights.DEFAULT` to get
 the most up-to-date weights.
  warnings.warn(msg)
./static/video/WhatsApp Video 2024-03-30 at 11.07.05 AM.mp4
C:\Users\91970\AppData\Local\Programs\Python\Python311\Lib\site-packages\torch\nn\modules\module.py:1511: UserWarning: Implicit dimen
sion choice for softmax has been deprecated. Change the call to include dim=X as an argument.
  return self._call_impl(*args, **kwargs)
confidence of prediction: 99.80224370956421
This is me  99.80224370956421
10
127.0.0.1 - - [01/Apr/2024 21:01:19] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_0.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_2.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/css/uploadPage.css HTTP/1.1" 304 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_1.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_3.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_4.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_5.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_6.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_7.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/upload.png HTTP/1.1" 304 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_8.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_9.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_10.jpg HTTP/1.1" 200 -
```

Fig10: In Command Prompt

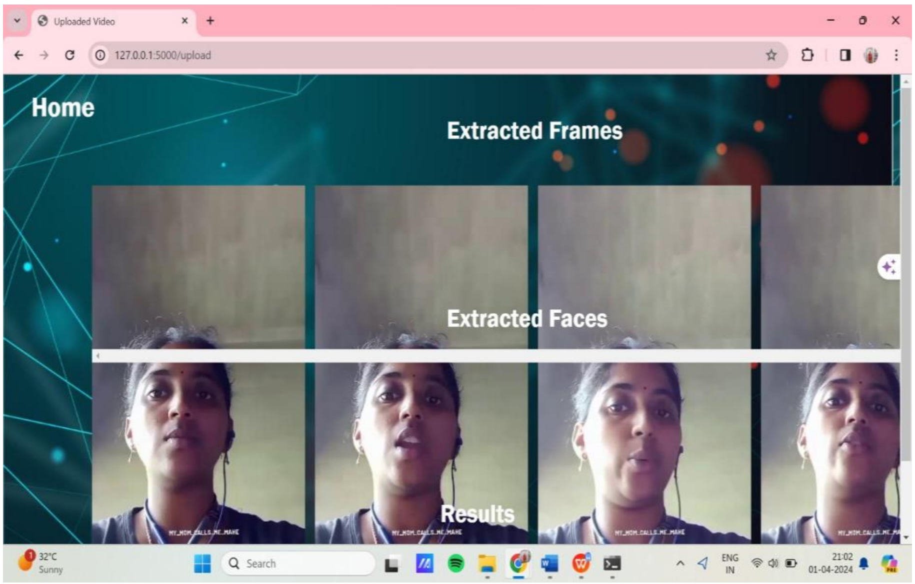
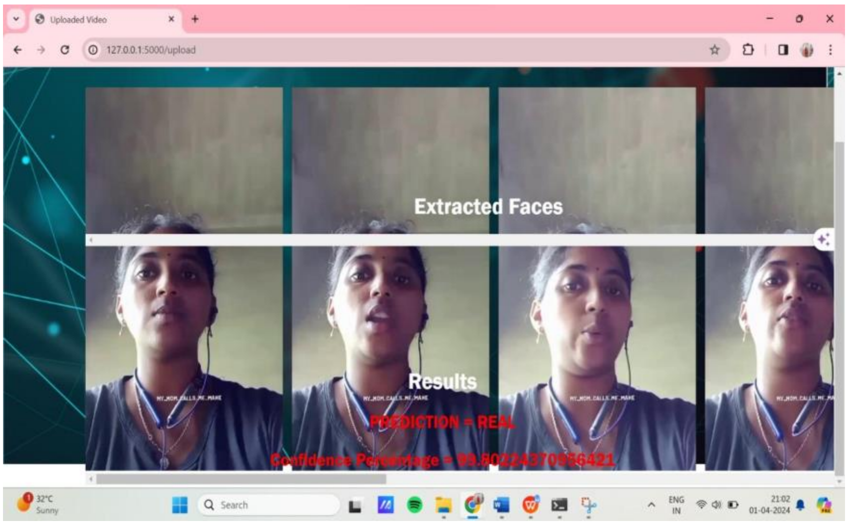


Fig11: Extracted Frames



PREDICTION = REAL

Confidence Percentage = 99.80224370956421

Fig12: Extracted frames into Extracted faces and the Result is Real

DOI: 10.17148/IARJSET.2024.11339

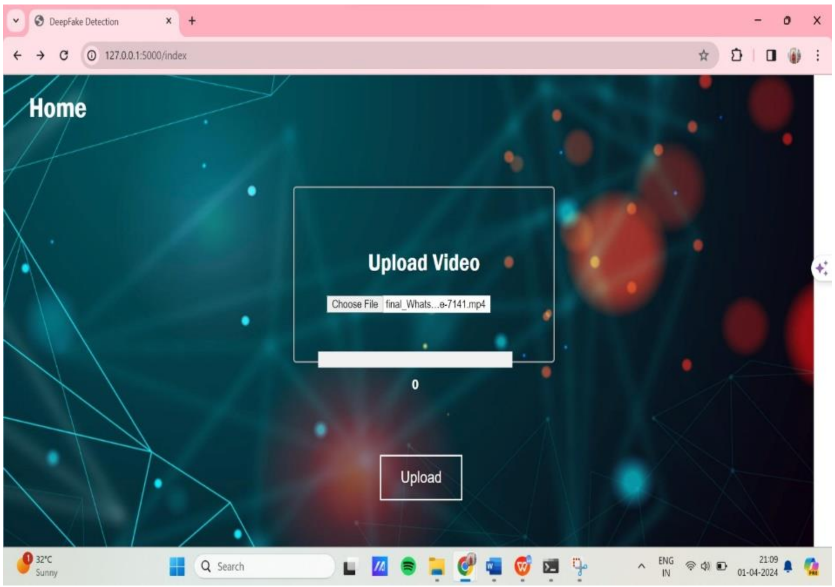
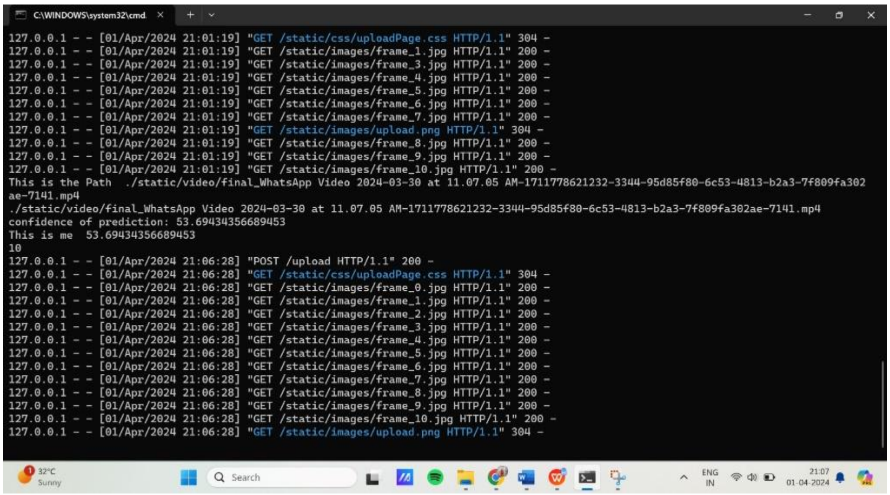


Fig13: Video uploaded from the file



```
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/css/uploadPage.css HTTP/1.1" 304 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_1.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_3.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_4.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_5.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_6.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_7.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/upload.png HTTP/1.1" 304 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_8.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_9.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:01:19] "GET /static/images/frame_10.jpg HTTP/1.1" 200 -
This is the Path  ./static/video/final_WhatsApp Video 2024-03-30 at 11.07.05 AM-1711778621232-3344-95d85f80-6c53-4813-b2a3-7f809fa302
ae-7141.mp4
./static/video/final_WhatsApp Video 2024-03-30 at 11.07.05 AM-1711778621232-3344-95d85f80-6c53-4813-b2a3-7f809fa302ae-7141.mp4
confidence of prediction: 53.69434356689453
This is me  53.69434356689453
10
127.0.0.1 - - [01/Apr/2024 21:06:28] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/css/uploadPage.css HTTP/1.1" 304 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_0.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_1.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_2.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_3.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_4.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_5.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_6.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_7.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_8.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_9.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/frame_10.jpg HTTP/1.1" 200 -
127.0.0.1 - - [01/Apr/2024 21:06:28] "GET /static/images/upload.png HTTP/1.1" 304 -
```

Fig14: In command Prompt

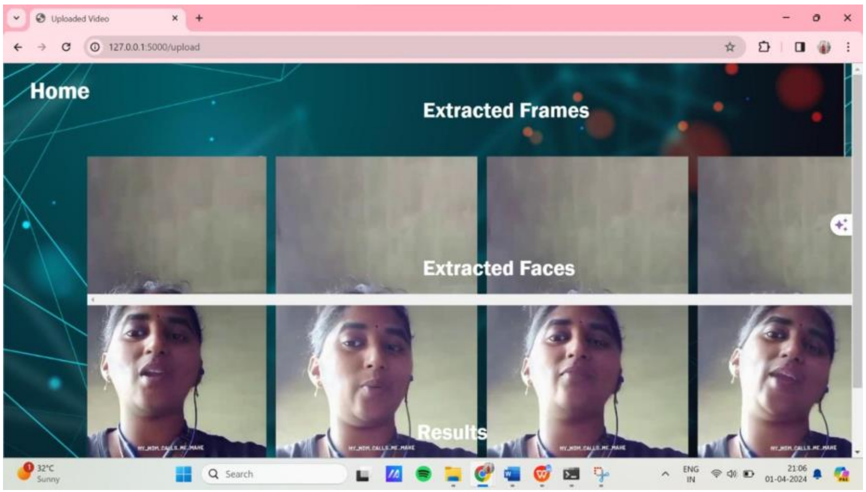
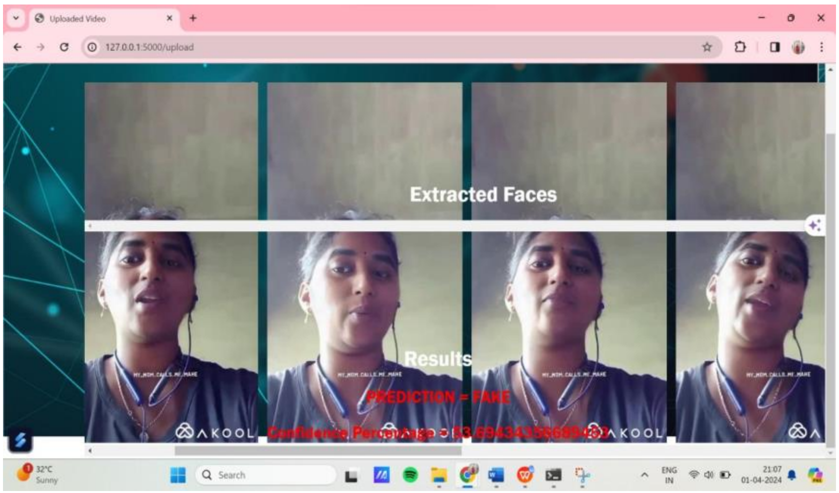


Fig15: Extracted into frames

DOI: 10.17148/IARJSET.2024.11339



Extracted Faces

Results

PREDICTION = FAKE

Confidence Percentage = 53.69434356689453

Fig16: Extracted frames into Extracted faces and the result is FAKE

VI. CONCLUSION

To sum up, using Recurrent Neural Networks (RNNs) for deepfake identification is a big step forward in tackling the problems brought on by the spread of synthetic media. RNNs' temporal analysis skills have demonstrated promise in identifying minute patterns and dependencies in video sequences, which could improve the accuracy with which real and fake information is distinguished. Convolutional Neural Networks (CNNs) offer spatial analysis, while RNN integration in deepfake detection designs complements it and enables a comprehensive comprehension of the dynamic nature of deepfake films. This combination of temporal and geographic data improves the model's detection capabilities against advanced manipulation strategies, giving it a stronger resistance against developing deepfake generating approaches. The review of the literature indicates that researchers are aware of the value of temporal analysis in deepfake detection, as evidenced by the numerous papers that demonstrate the efficiency of RNNs, LSTM networks, and bidirectional designs. A more sophisticated method is provided by the detection pipeline's use of RNNs, which can recognize the sequential nature of gestures, facial expressions, and abnormalities that can point to deepfake content. Still, there are problems, and they should be the focus of future study efforts.

REFERENCES

[1] DeepFakes Program. Reached on August 20, 2022. [Online]. Faceswap can be accessed at https://github.com/deepfakes
[2] Adversarial Losses + A Denoising Autoencoder + Attention Mechanisms for Face Swapping. Reached on August 20, 2022. [Online]. https://github.com/shaoanlu/faceswap-GAN is accessible.
[3] The Best Software for Producing DeepFakes is DeepFaceLab. Retrieved: February 24, 2022. [Online]. DeepFaceLab is accessible at https://github.com/iperov
[4] Larger Resolution Face Masked, Weirdly Warped, DeepFake. Retrieved: February 24, 2022. [Online].dfaker/df is accessible at https://github.com
[5] "Animal communication: Will you answer when I call?" said N. J. Vickers. July 2017, Current Biol., vol. 27, no. 14, pp. R713–R715.
[6] DeeperForensics1.0: A large-scale dataset for real-world face forgery detection, L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2889–2898.
[7] "StarGAN: Unified generative adversarial networks for multi-domain imageto-image translation," Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, Proc. IEEE Conf. Comput. Vis. pattern Recognit., Jun. 2018, pp. 8789–8797.
[8] "Progressive growing of GANs for improved quality, stability, and variation," T. Karras, T. Aila, S. Laine, and J. Lehtinen, arXiv:1710.10196, 2017.
[9] "A style-based generator architecture for generative adversarial networks," by T. Karras, S. Laine, and T. Aila, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4401–4410.
[10] "First order motion model for image animation," A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–11.
In Proc. 6th Int. Conf. Comput. Sci. Eng. (UBMK), Sep. 2021, pp. 36–41, A. S. Uçan, F. M. Buçak, M. A. H. Tutuk, H. İ. Aydin, E. Semiz, and S. Bahtiyar, "Deepfake and security of video conferences." [12] N. Graber-Mitchell, "Deepfakes

DOI: 10.17148/IARJSET.2024.11339

as speech: Artificial illusions," Amherst College, Massachusetts, USA, Tech. Rep., 2020, vol. 14, no. 3.
[13] "A robust facemask forgery detection system in video," by F. H. Almukhtar, Proceedings of the Society for Engineering Natural Science, vol. 10, no. 3, pp. 212–220, 2022.
[14] "The deepfake detection challenge (DFDC) preview dataset," by B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer arXiv:1910.08854 (2019).
[15] "A survey on deepfake video detection," P. Yu, Z. Xia, J. Fei, and Y. Lu, IET Biometrics, vol. 10, no. 6, pp. 607–624,

.

BIOGRAPHIES

Mrs. P. Neelima
M. Tech, Asst. Professor, Dept of
Computer Science and Engineering,
BWEC, Andhra Pradesh, India.

Naidu Keerthi Lakshmi Prasanna
[B. Tech], Student, Dept of
Computer Science and Engineering,
BWEC, Andhra Pradesh, India.

Yandamuri Sravani [B. Tech],
Student, Dept of Computer
Science and Engineering,
BWEC, Andhra Pradesh, India

Pinneti Maheswari [B. Tech],
Student, Dept of Computer
Science and Engineering,
BWEC, Andhra Pradesh, India