Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7766132

SPOTTING FAKE VIDEOS: A SIMPLE SOLUTION FOR DETECTING DEEPFAKES USING DEEP LEARNING

G Sri Devi[1], B Hemanth Reddy[2], B K R Mohan Kumar[3], G Bharath Kumar[4]
sridevigadde.85@gmail.com,19981A0522@raghuenggcollege.in
19981A0516@raghuenggcollege.in,19981A0548@raghuenggcollege.in
[1]Assistant professor, Department of Computer Science and Engineering, Raghu Engineering College,[2,3,4] B.Tech Students , Department of Computer Science and Engineering, Raghu Engineering

Abstract
The emergence and progression of deepfake have caused widespread concern, as it can create misleading content that can potentially manipulate public opinion. People are finding it harder and harder to tell what is real and what is a deep fake because deep fakes may modify images and videos. The human eye is not always able to detect the results of deepfake. To address this issue, a deep learning-based method has been developed that can efficiently differentiate fake videos. The proposed method for detecting manipulated videos uses a pre-trained Res-Next convolutional neural network to extract features from frames of the input video. These features are subsequently utilized to train a long short-term memory (LSTM) network, which can classify whether the input video has been tampered with or not. To enhance the model's performance on real-time data, it has been trained with a balanced dataset. The Deepfake Detection Challenge dataset was used to train the model. The model that has been proposed has yielded promising results by accurately predicting the output. This model is expected to reduce the potential harm that deep fakes pose to society.

Keywords: Deepfakes, Deep learning, Res-Next Convolutional Neural Network, LongShort-Term Memory (LSTM), Deepfake Detection Challenge Dataset.

I. Introduction

Detecting deep fakes holds significant implications for multiple fields, including politics, journalism, and entertainment. Deepfake can be misused in politics to manipulate speeches or videos of political figures, causing public confusion and a loss of trust. In journalism, deepfakes can be utilised to generate false news stories, leading to misinformation and confusion among the public. Similarly, in entertainment, deepfakes can be used to produce fabricated celebrity videos, which can harm their reputation and create chaos among their fans.

The legal system can also greatly benefit from a deep-fake detection system. Deepfakes have the potential to create fabricated evidence or tamper with video footage, resulting in erroneous convictions. By utilizing deep-fake detection, fabricated evidence can be identified, ensuring that justice is served and wrongful convictions are prevented.

One of the best-known examples of deepfake is a video of former President Barack Obama [12], which was manipulated to show him delivering a speech that he never actually gave. This video was created by taking footage of Obama and using DeepFake to manipulate his facial

expressions and mouth movements to sync with the words of the fake speech. The result was a highly convincing video that appeared to be genuine at first glance. This example highlights the potential dangers of deepfake, as it can be used to spread false information and manipulate public opinion.

To tackle the increasing threat posed by deepfake videos, it is crucial to develop effective methods for detecting them. Therefore, a new approach based on deep learning has been proposed that can accurately distinguish between artificially generated fake videos and authentic ones. The importance of identifying and preventing the spread of fake videos cannot be overstated, and the proposed method aims to achieve this by leveraging the power of deep learning. To detect manipulated videos, the method uses a pre-trained Res-Next convolutional neural network to extract features from the input video's frames, which are then utilized to train a long short-term memory (LSTM) network. The LSTM network can classify whether the input video has been tampered with or not. This technology can be employed to identify fake videos and prevent the potential harm caused by the dissemination of misinformation and fake news on the internet.

Fig 1. Couple ofreal images and deepfakes

II. Literature Review

The authors provide a detailed account of the dataset, baselines, and challenge structure used in the competition, as well as the metrics used to assess the performance of the models. The authors utilized a dataset of 5,639 videos to train and evaluate their models. They experimented with various baseline models, including SVMs with feature extraction methods such as HOG, LBP, and CNN features. They evaluated the model's performance using the AUC metric. The study's findings reveal that the top-performing SVM-based model achieved an AUC of 62.9% for video-based deepfake detection, while the best-performing SVM-based model for image-

based deepfake detection achieved an AUC of 78.8%. The results illustrate that SVM-based models can achieve reasonable accuracy for detecting some types of deepfakes [1].

The authors propose a new method for identifying deepfake videos generated through face swapping. The method involves using aRNN model with a two-stream architecture that processes both the original and manipulated videos to extract features. The extracted features are then used by an RNN-based classifier to identify whether the video is a deepfake or not. The researchers evaluated their approach using the DFDC dataset, a large dataset of manipulated videos used for evaluating deep-fake detection algorithms. The proposed method achieved an accuracy of 76.3% on the DFDC dataset, which is lower than the top-performing models in the DFDC competition [2].

The authors proposed an approach that leverages a model based on a deep MLP-CNN to detect and uncover deep fakes. This approach is novel in the sense that it combines the strengths of both neural network architectures, allowing for the extraction of high-level features from input images and the identification of spatial patterns and dependencies between these features. The authors' primary objective was to develop a deep learning-based model to detect fake images, and to achieve this, they collected a dataset of 1,500 real and 1,500 fake images generated using a face-swapping algorithm. Using the TensorFlow and Keras libraries, the authors designed a MLP-CNN model that integrates the strengths of both neural network architectures. Specifically, the MLP component of the model extracts high-level features from the input images, while the CNN component identifies the spatial patterns and dependencies between these features. The effectiveness of the model was evaluated by the authors using multiple performance metrics, such as accuracy. Additionally, to evaluate the model's performance, a separate dataset containing 200 real and 200 deep-fake images was utilized [3].

The authors propose that the advancement of deepfake technology has posed a significant threat to the authenticity and trustworthiness of audio and visual information. The fake videos have been used to spread disinformation, manipulate public opinion, and even blackmail individuals. In this regard, the author suggests a deep learning approach that utilises mouth movement and transfer learning to enhance the precision of identifying fakes. The CNN is used to extract features from video frames, while the RNN models the temporal relationship between the frames. The VGG-Face model is pre-trained and used to extract facial features, which are then fed into the RNN for classification. The method incorporates mouth movement information by extracting the mouth region using a separate mouth region extractor and using a CNN to extract mouth features, which are concatenated with the facial features and fed into the RNN. The pre-trained VGG-Face CNN model is fine-tuned using a large dataset, and data augmentation techniques are used to increase the model's robustness [4].

The authors of this paper presented a deep fake detection model that utilizes a CNN architecture. The proposed model takes a single video frame as input, and through the use of a CNN, it extracts important features from the image that are then used for binary classification, determining whether the video is a deepfake or real. The authors also explored different training strategies, including data augmentation and transfer learning, which significantly improved the performance of the model [5].

The authorsused a pre-trained CNN model called Inception-V3, which has been trained on a large dataset of images. They adapted this model for video analysis by using a technique called

Two-Stream Inception-V3. This technique involves using two separate streams, one for spatial analysis and one for temporal analysis. The two streams are then combined to generate the final feature representation of the video. After extracting the spatio-temporal features from the videos, the authors trained a classifier using a SVM algorithm. The authors evaluated the performance of their approach using various evaluation metrics. Furthermore, they evaluated the effectiveness of their method by comparing it with other currently advanced deepfake detection techniques. Overall, the authors used a combination of deep learning and machine learning algorithms to extract and classify spatio-temporal features from videos to detect deep fakes [6].

This method, designed to locate and extract images that reveal Convolutional Traces (CT) during image formation, is built upon the Expectation-Maximization algorithm and has shown remarkable discriminative power and effective outcomes. While successful results have been achieved, further exploration can be conducted on renowned real-image forensics datasets like DRESDEN, UCID, or VISION [7].

In this study, the author proposes a model for detecting facially modified videos from a set perspective, utilizing a distinct architecture called the set convolutional neural network (SCNN). The SCNN network incorporates features from multiple video frames and learns feature maps of the altered facial images. This innovative method can simultaneously process multiple input video frames, enhancing the accuracy of the detection process [8].

The authors of this paper put forward a deepfake detection model that employs disentangled representations and takes into account various modalities of a video, such as RGB frames, optical flow, and audio spectrograms. The features of each modality were extracted using a disentangled representation learning method, which captures the fundamental attributes of the data. The model combines the disentangled representations via a multi-modal fusion network that adjusts the weighting of each modality for the final prediction. To enable the network to learn distinctive features for deepfake detection, the authors employed a contrastive loss function. [9].

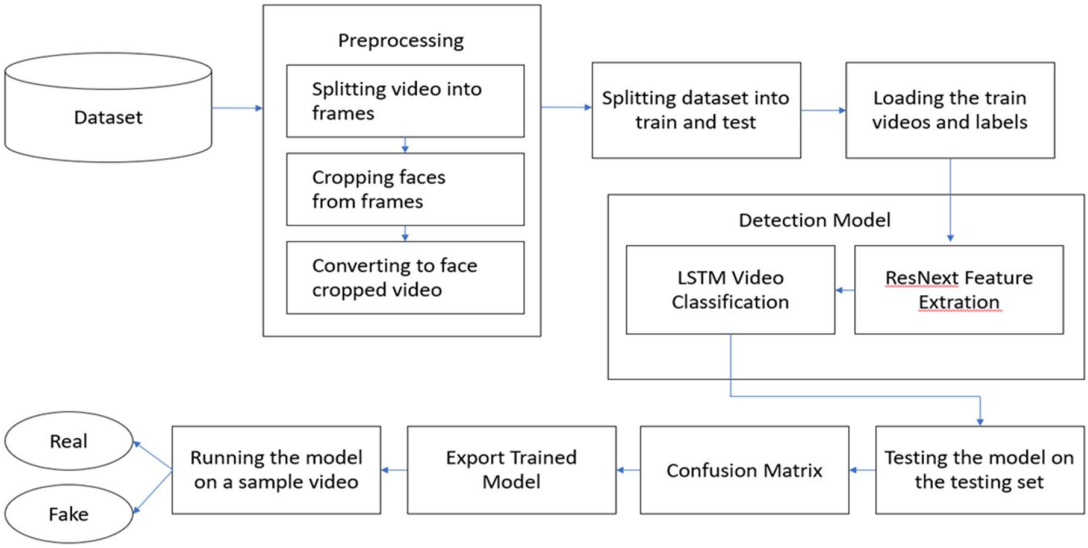
The authors introduced a deepfake detection model that integrates RGB, depth, and thermal data. The model utilizes a CNN architecture to extract features from each type of data and then applies a dynamic fusion method to combine the features. This fusion method calculates weights for each modality's contribution to the final prediction based on the input data. Furthermore, the authors incorporated a dropout regularization technique and a focal loss function to enhance the model's performance [10].

For classification purposes, the proposed approach utilizes feature engineering as a pre-filtering strategy, followed by the use of CNN-based deep learning frameworks. The author's method relies on traditional frequency analysis of photographs, which identifies different behaviors at higher frequencies. [11]

III. Methodology

The present method for deepfake detection is developed using a combination of CNN and RNN. The model under consideration receives a video as input from the user, and its objective is to ascertain if the video has been manipulated or is authentic. The first step involves subjecting the input video to several preprocessing stages, resulting in a face-cropped video.

This face-cropped video is then forwarded to a ResNext CNN model that has been pre-trained to extract features. Then a LSTM network is developed for classifying the input video as real or fake.



3.1 Proposed Algorithm

1 Import the necessary libraries and define any constants or hyperparameters.
2 Define the Model class, which inherits from nn.Module, and implement its architecture in the __init__ method and the forward pass in the forward method.
3 Instantiate the Model class with the desired number of classes and other hyperparameters.
4 Define the loss function (criterion) and the optimizer (optimizer).
5 Load the training and validation datasets using a DataLoader.
6 For each epoch, do the following:
a. Set the model to training mode using model.train().
b. Iterate through each batch of the training data:
i. Load the batch of data and move it to the GPU (if available).
ii. Compute the model's output and loss for the batch.
iii. Compute the accuracy for the batch.
iv. Backpropagate the loss and update the model parameters using the optimizer.
v. Update the average loss and accuracy for the epoch.
c. Set the model to evaluation mode using model.eval().
d. Iterate through each batch of the validation data:
i. Load the batch of data and move it to the GPU (if available).
ii. Compute the model's output and loss for the batch.
iii. Compute the accuracy for the batch.
iv. Update the average validation loss and accuracy for the epoch.
e. Print the epoch number, training loss, training accuracy, validation loss, and validation accuracy.
f. Save the model state using torch.save() if desired.

7 After training is complete, evaluate the model on the test set if desired.

3.2Dataset Gathering

The study utilized a dataset obtained from Kaggle, specifically the DFDC (Deepfake Detection Challenge) dataset, which is openly available for anyone to use. This dataset is the most comprehensive publicly available dataset for training the proposed method. The entire dataset is over 470 GB in size, comprising more than fifteen thousand videos, and is provided as a single file that is further divided into 50 smaller files, each of which is 10 GB in size. For this study, one of the smaller files, which includes 802 files with extensions of mp4, json, and csv, was used for development and training. The dataset contains two main folders, namely train and test. The train folder also includes a metadata.json file that provides the filename and label of each video.

3.3Preprocessing

To detect the presence of deepfakes, the initial step involves preprocessing the dataset to ensure that only relevant data is present and to reduce the workload of subsequent steps. This involves converting the input video into frames, detecting and cropping the face in each frame, and then combining the face-cropped frames into a new video to create a new dataset that only contains face data. Frames without faces are discarded. Since a 10-second video at 30 frames per second contains 300 frames, processing all of these frames requires significant computational power. To classify deepfakes more efficiently, frames are processed sequentially and inputted into the long short-term memory.

3.4Model Training

3.4.1 ResNext50 32x4d

ResNext50 32x4d is a CNN architecture that has been pre-trained on large datasets such as ImageNet for image classification. Instead of developing the feature extraction code from scratch, a pre-trained CNN model is utilized in this study to extract frame-level features from the input video. To achieve improved performance with deeper neural networks, an optimised residual network is used. Specifically, a resnext50_32x4d model [13] is utilised, consisting of 50 layers with 32x4 dimensions. The model is fine-tuned by adding additional layers and selecting an appropriate learning rate to ensure the convergence of gradient descent. At the end of the ResNext model's last pooling layer, a 2048-dimensional feature vector is generated. This feature vector is sequentially inputted into the LSTM network for further processing.

stage	output	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 [1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax
# params.		**25.0**×10^6

Fig 3. ResNext Architecture

3.4.2 Sequential Layer

The sequential layer is a frequently used layer type in neural networks for defining a sequential order of layers. It is a fundamental component of deep learning models that allows for the stacking of multiple layers to create a complex architecture. In this architecture, each layer takes the output produced by the preceding layer as its input. Our model utilises this layer type, where the ResNext layer's output feature vector is saved within the Sequential layer and then passed as input to the LSTM network.

3.4.3 LSTM Model

The output feature vector obtained from ResNext, which has a dimensionality of 2048, is used as input for an LSTM network. The proposed method uses a single LSTM layer with 2048 hidden units and a dropout rate of 0.4 to improve the performance of the model. Temporal analysis is carried out by the LSTM, which sequentially processes the frames. Each frame with a duration of t seconds is compared with a frame of t-n seconds. An average pooling layer is used to obtain the target output size, and a RELU activation function is employed in the model. To process the frames sequentially, a sequential layer is utilized. Finally, a softmax layer is used to obtain the prediction confidence. The below figure gives an internal view of the LSTM architecture.

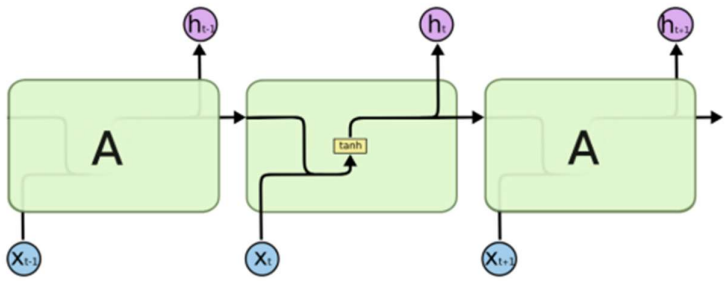


Fig 4. Internal LSTM Architecture

3.4.3.1 Dropout Layer

To prevent overfitting of deep learning models to training data, a common technique used is called dropout. This method involves randomly dropping out a certain proportion of neurons during training to encourage the remaining neurons to learn more robust and generalizable features. For instance, a dropout rate of 0.4 indicates that 40% of the neurons within the LSTM layer will be randomly dropped out during training.

3.4.3.2 ReLU

The ReLU is an activation function that has become popular in neural networks for its effectiveness and ease of implementation. It works by taking an input value x and returning the maximum of that value and zero. ReLU has the advantage of not having any backpropagation errors. which is represented as

$ReLU(x) = max(0,x)$.

3.4.3.3 Average Pooling Layer

The average pooling layer partitions the input feature map into non-overlapping regions known as pooling regions, which are generally of a fixed size, usually 2x2 or 3x3. In the current study, a two-dimensional adaptive average pooling layer is employed to minimize computational complexity and variance. This type of layer functions similarly to an average pooling layer, but with the added capability of computing the appropriate kernel size necessary to generate an output with the desired dimensionality from the provided input.

3.4.3.4 Softmax Layer

The Softmax function is a squashing function that restricts the function's output within the range of 0 to 1, making it directly interpretable as a probability. In our case, the Softmax layer has two output nodes, namely REAL or FAKE, and it provides the prediction's confidence or probability.

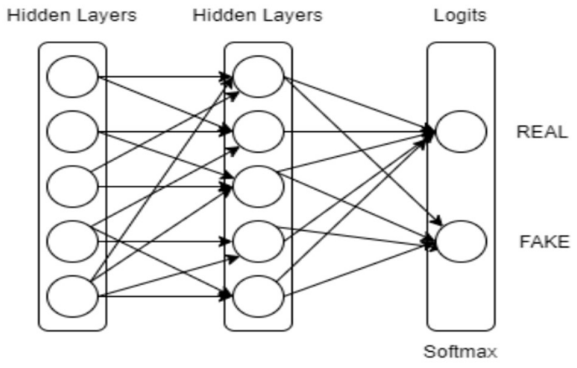


Fig 5. Softmax Layer

3.5 Hyperparameter Tuning

Selecting appropriate hyperparameters is crucial for developing a ML model that is both efficient and accurate. To optimize the learning rate, an adaptive learning rate algorithm such asthe Adam optimizer is used in conjunction with the model parameters. In order to achieve better convergence during gradient descent, a learning rate of 0.00001 and weight decay are

employed. To maximize computational efficiency, batch training is utilized. Since the current problem is a classification problem, a cross entropy loss function is used.

IV. Results

The proposed model was implemented using a dataset consisting of 800 videos from the Deepfake Detection Challenge, resulting in 636 training videos and 160 testing videos. Of the training videos, 319 were real and 317 were fake, while the testing videos contained 72 real and 88 fake. During training, 70% of the total dataset was utilised, and the model was observed to accurately decrease losses over time, as demonstrated by the accuracy and loss graph.

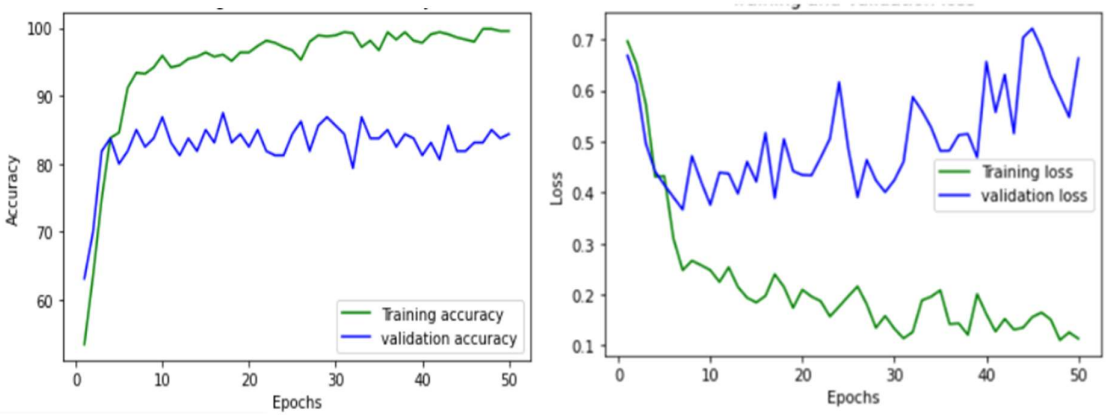


Fig 6. Accuracy Graph Fig 7. Loss Graph

The proposed model's prediction results are summarized using a confusion matrix.This matrix provides an overview of the errors made by the classification model and illustrates the ways in which the proposed model is confused during the prediction process. The confusion matrix is a tool used to evaluate and estimate model's performance. Below is a depiction of the confusion matrix that was generated upon implementing the proposed model. This matrix provides an overview of the performance of the classification model, highlighting the number of correct and incorrect predictions made by the model. By analyzing the errors made by the model, we can gain insights into the ways in which the model is confused while making predictions. Ultimately, the confusion matrixserves as a valuable tool for evaluating the effectiveness of the proposed model in detecting and classifying deepfake videos.

Accuracy = (TP + TN) / (TP + FP + TN + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

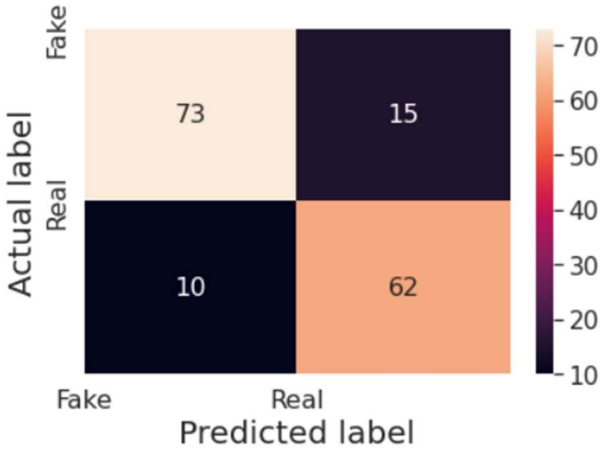


Fig 8. Confusion Matrix

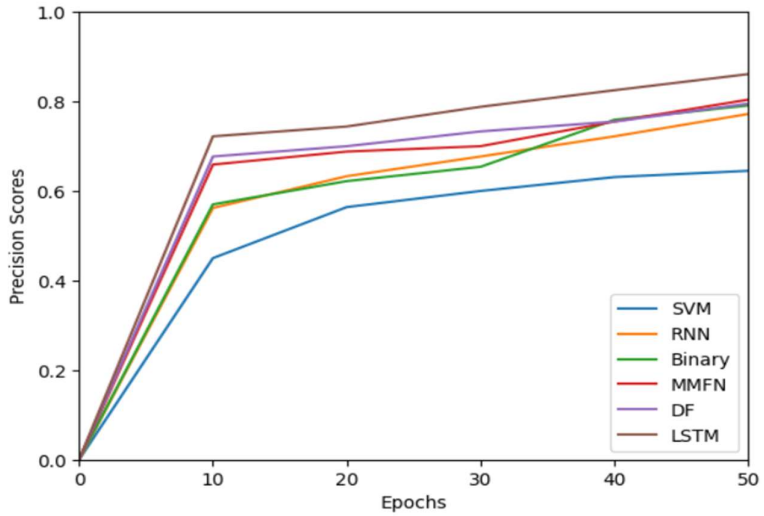


Fig 9. Precision Graph

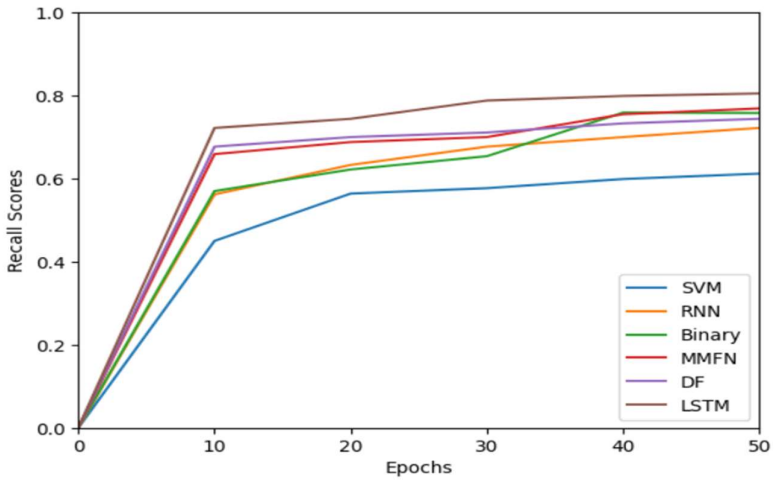


Fig 10. Recall Graph

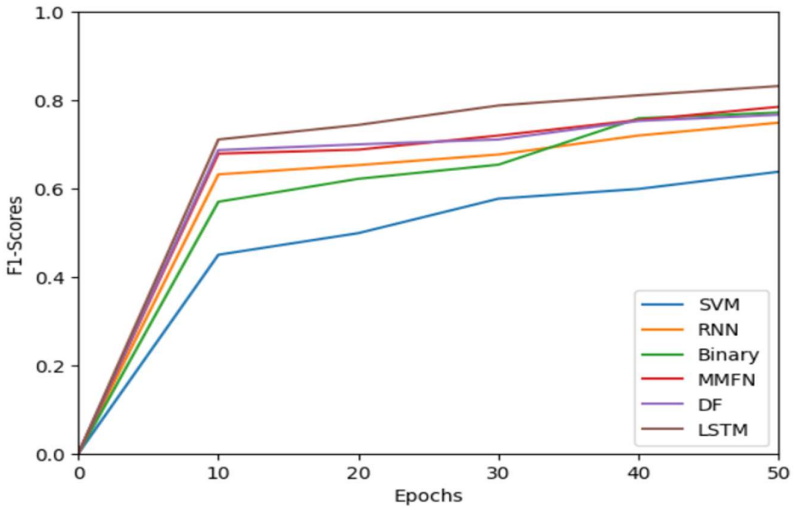


Fig 11.F1-Score Graph

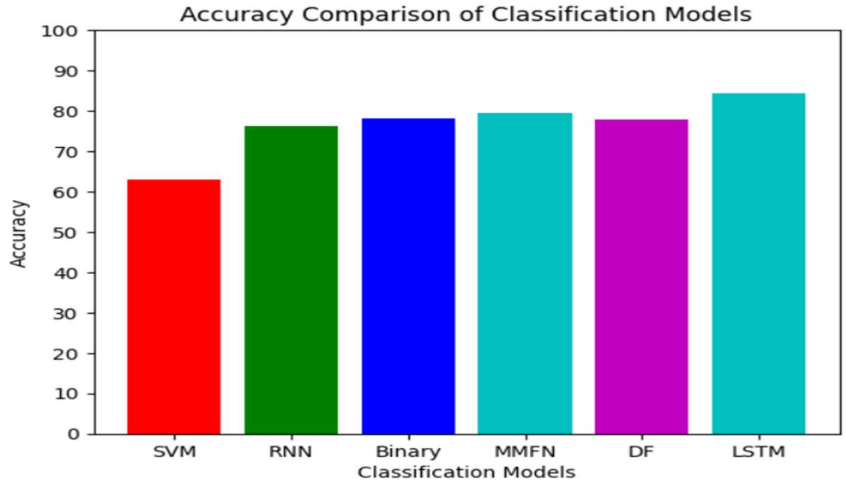


Fig 12. Comparison Graph

Classification Model	Accuracy	Precision	Recall	F1-Score
SVM Based Model	0.629	0.645	0.612	0.638
RNN	0.763	0.772	0.722	0.749
Binary Classification Model	0.782	0.791	0.758	0.772
Multi-Model Fusion Network	0.796	0.804	0.769	0.785
Dynamic Fusion	0.778	0.795	0.744	0.767
LSTM	0.843	0.861	0.805	0.832

Table 1: Classification Model Performance

We have taken up various classification models like the SVM-based model, the RNN, binary classification, the multi-model fusion network, and dynamic fusion, and their accuracies are 62.9%, 76.3%, 78.2%, 79.6%, and 77.8%, respectively. So, we used the LSTM network model as it predicts the deepfakes more accurately when compared to other models.

V. Conclusion

With the rapid advancement of technology, deepfakes have become a significant threat to both ordinary individuals and celebrities. The misuse of deepfake technology has caused widespread panic and even resulted in loss of life. Therefore, in the present study, a deep learning model that combines CNN and RNN is developed to identify whether a given video is original or a deepfake. The input video is first preprocessed and then passed to a pretrained ResNext model to extract features. The output of the ResNext CNN model is then fed into the LSTM network for accurate classification. The proposed model is evaluated using a confusion matrix, and the results demonstrate that it efficiently distinguishes between real and manipulated videos. This improvement can prevent the spread of fake videos through social media and other platforms, thereby avoiding panic among the public.Currently, the algorithm is capable of detecting only Face Deep Fakes, but it can be further developed to identify full-body deep fakes as well.

VI. References

[1]. Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2019). *Deepfake detection challenge (DFDC) preview:Dataset, baselines and challenge structure*.arXiv preprint arXiv:1910.08854.

[2]. T. Nguyen, H. Nguyen, and T. T. Nguyen, *Deepfake Video Detection Using Recurrent Neural Networks*,in Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2020, pp. 1-8. DOI: 10.1109/AVSS48556.2020.9224615.

[3]. Kolagati, S., Priyadharshini, T., & Rajam, V. M. A. (2022). *Exposing deepfakes using a deep multilayer perceptron–convolutional neural network model*. International Journal of Information Management Data Insights, *2*(1), 100054.

[4]. Elhassan, A., Al-Fawa'reh, M., Jafar, M. T., Ababneh, M., & Jafar, S. T. (2022). DFT-MF: *Enhanced deepfake detection using mouth movement and transfer learning*. SoftwareX, *19*, 101115.

[5]. M. K. Khan, S. Z. Hussain, H. Tariq, and A. B. Javed, "*DeepFake Detection Using a Convolutional Neural Network*," in Proceedings of the 10th International Conference on Intelligent Human Computer Interaction (IHCI), 2020, pp. 1-6.

[6]. Nguyen, X. H., Tran, T. S., Nguyen, K. D., & Truong, D. T. (2021*). Learning spatio-temporal features to detect manipulated facial videos created by the deepfake techniques*. Forensic Science International: Digital Investigation, *36*, 301108.

[7]. Guarnera, L., Giudice, O., & Battiato, S. (2020). *Fighting deepfake by exposing the convolutional traces on images*. IEEE Access, *8*, 165085-165098.

[8]. Xu, Z., Liu, J., Lu, W., Xu, B., Zhao, X., Li, B., & Huang, J. (2021). *Detecting facial manipulated videos based on set convolutional neural networks*. Journal of Visual Communication and Image Representation, *77*, 103119.

[9]. M. Sabir, S. Khan, S. Iqbal, M. A. S. Butt, and F. S. Baig, "*Multi-Modal Deep Fake Detection Using Disentangled Representations*," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 3137-3141.

[10]. Jaiswal, S., Gupta, A., & Gaur, M. S. (2021*). DeepFake Detection using Convolutional Neural Networks and Dynamic Fusion of RGB, Depth and Thermal Data*. 2021 11th International Conference on Cloud Computing,DataScience & Engineering (Confluence), 31-35. doi: 10.1109/CONFLUENCE51747.2021.9394058

[11]. Burroughs, S. J., Gokaraju, B., Roy, K., & Khoa, L. (2020, October). *DeepFakes Detection in Videos using Feature Engineering Techniques in Deep Learning Convolution Neural Network Frameworks.* In *2020* IEEE Applied Imagery Pattern Recognition Workshop (AIPR) (pp. 1-4). IEEE.

[12]. https://www.creativebloq.com/features/deepfake-examples

[13]. ResNext Model : https://pytorch.org/hub/pytorch_vision_resnext/ accessed on 30March 2023

[14]. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep residual learning for image recognition. In CVPR*, 2016

[15]. Sequence Models And LSTM Networks https://pytorch.org/tutorials/beginner/nlp/sequence_mod els_tutorial.html

[16]. https://www.kaggle.com/c/deepfake-detection-challenge/data