International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

DFFMD: A Deepfake Face Mask Dataset for Infectious Disease Era with Deepfake Detection Algorithms

G.SRUJAN KUMAR, Assistant professor, Dept of MCA, Chirala Engineering College, Chirala
srujan9032@gmail.com

BANAGANIPALLI KHASIM, PG Student – MCA, Dept of MCA, Chirala Engineering College, Chirala
khasimbanagani@gmail.com

Abstract: This research addresses the escalating concerns surrounding deepfake technology, an AI-driven method for generating synthetic media that can propagate false narratives and enable various forms of digital manipulation. The ubiquity of face masks, exacerbated by the COVID-19 pandemic, has exacerbated the challenges of detecting deepfake videos, necessitating advanced detection methodologies.To address this gap, the project introduces the Deepfake Face Mask Dataset (DFFMD), a specialized dataset aimed at training detection models to identify deepfake content featuring individuals wearing face masks. Leveraging an innovative approach, the study employs an Inception-ResNet-v2 architecture, incorporating preprocessing stages, feature-based analysis, residual connections, and batch normalization. This sophisticated model surpasses conventional methods such as InceptionResNetV2 and VGG19, particularly excelling in scenarios involving face-masked subjects.Experimental results showcase the remarkable accuracy of the proposed model in detecting deepfake videos with face masks, underscoring its potential as an effective countermeasure. The study advocates for continued research to enhance detection capabilities, recognizing the evolving nature of deepfake threats and the need for adaptive solutions. Key terms include Inception ResNetV2, VGG19, CNN, and Xception, highlighting the utilization of advanced architectures in the pursuit of robust deepfake detection.

Index Terms: Deepfake, deep learning, CNN, generation, detection, fake videos, neural network, mask, face mask.

1. INTRODUCTION

The rapid advancement of technology in recent years has led to the proliferation of computer-generated editing programs, revolutionizing the synthesis and modification of media content. With this growth, however, comes the alarming potential for the spread of misinformation, particularly through the emergence of deepfake technology. Deepfake, a term derived from "deep learning" and "fake," refers to a sophisticated method that utilizes deep learning algorithms to create fake videos, manipulate existing videos, or even synthesize speech to mimic someone's voice. This technology poses significant risks, as it can be exploited for malicious purposes such as spreading fake news and disseminating false or harmful information.

Since its emergence in 2017, deepfake technology has garnered widespread attention from both the research community and the general public due to its potential to deceive and manipulate. Detecting deepfakes has thus become a pressing challenge, prompting researchers to explore various machine learning techniques and methodologies to counter this threat. These efforts have spanned multiple approaches, ranging from facial analysis to the examination of specific regions such as eyes and lip movements, as well as the development of novel deep learning architectures.

In the current landscape, there is a plethora of deepfake tools readily available to the public, many of which are free, open-source, and accompanied by ample learning resources. Among the most prominent tools are Faceswap [1], Faceswap-GAN [2], DeepFaceLab [3], and DFaker [4]. These tools operate by swapping the source person's face with the target face, effectively creating a new video with the actions of the source individual but featuring the face of the target individual. The resulting videos produced by these tools present a significant challenge for human observers, as they are often indistinguishable from genuine footage [5].

The widespread availability and accessibility of deepfake tools underscore the urgent need for effective detection methods to mitigate the potential harms associated with their misuse. This introduction provides an overview of the escalating concerns surrounding deepfake technology, highlighting its implications for misinformation dissemination and the ongoing efforts to develop detection strategies. In the subsequent sections, we will delve deeper into the various approaches and techniques employed in the detection of deepfakes, examining both the progress made thus far and the challenges that lie ahead. Through a comprehensive exploration of the current landscape, this paper aims to contribute to the ongoing discourse on deepfake detection and its implications for society.

2. LITERATURE SURVEY

The proliferation of Deepfake technology has raised significant concerns regarding the potential for misinformation, manipulation, and privacy violations. In response, researchers have conducted extensive investigations into various methodologies for detecting and mitigating the impact of Deepfakes. This literature survey aims to provide a comprehensive overview of the research landscape in Deepfake detection, highlighting key methodologies, datasets, and challenges encountered in this rapidly evolving field.

The availability of high-quality datasets plays a crucial role in the development and evaluation of Deepfake detection algorithms. Several benchmark datasets have been introduced to facilitate research in this area. Notable examples include the Deepfake Detection Challenge (DFDC) Preview Dataset [14], which consists of thousands of videos containing both authentic and manipulated content. Similarly, the WildDeepfake dataset [16] provides a challenging real-world dataset for evaluating Deepfake detection algorithms under diverse conditions. Additionally, datasets such as Celeb-DF [18] and FaceForensics [27] have been widely utilized for training and evaluating Deepfake detection models.

Researchers have employed a variety of methodologies to detect Deepfake content, ranging from traditional image processing techniques to advanced deep learning algorithms. One common approach involves analyzing temporal inconsistencies in videos, such as unnatural facial movements or lip-sync errors [22]. Convolutional neural networks (CNNs) have emerged as a popular choice for Deepfake detection due to their ability to extract spatial features from images and videos [17]. Techniques such as frame-level analysis, optical flow estimation, and attention mechanisms have been integrated into CNN architectures to improve detection accuracy [22].

In addition to CNNs, recurrent neural networks (RNNs) have been applied to model temporal dependencies in video sequences, enabling the detection of subtle anomalies indicative of Deepfake manipulation [17]. Generative adversarial networks (GANs), which are commonly used to generate Deepfake content, have also been leveraged for detection purposes. Adversarial training, where a detection model is trained alongside a GAN to distinguish between real and fake videos, has shown promising results in improving detection robustness [26].

Despite significant progress in Deepfake detection research, several challenges remain to be addressed. One major challenge is the rapid evolution of Deepfake technology, which continually introduces new techniques and countermeasures to evade detection [11]. Additionally, the availability of open-source Deepfake tools and datasets has lowered the barrier to entry for malicious actors, increasing the prevalence of Deepfake content on the internet [1]. As a result, there is a pressing need for more robust and scalable detection algorithms that can adapt to evolving threats.

Furthermore, the ethical and legal implications of Deepfake detection pose complex challenges for researchers and policymakers. Issues such as privacy rights, freedom of expression, and the potential for algorithmic bias must be carefully considered in the development and deployment of Deepfake detection systems [11]. Moreover, the lack of standardized evaluation metrics and benchmarks makes it difficult to compare the performance of different detection algorithms [17]. Addressing these challenges will require interdisciplinary collaboration between researchers, policymakers, and industry stakeholders.

In conclusion, the detection and mitigation of Deepfake content present multifaceted challenges that require innovative solutions from the research community. By leveraging advanced machine learning techniques, analyzing large-scale datasets, and addressing ethical and legal considerations, researchers can develop effective strategies for combating the spread of Deepfake content. However, ongoing vigilance and collaboration will be essential to stay ahead of emerging threats and safeguard the integrity of digital media.

3. METHODOLOGY

a) Proposed Work:

The proposed work aims to develop and implement the Deepfake Face Mask Detection (DFFMD) system, which integrates cutting-edge deep learning models and preprocessing techniques to enhance the accuracy of deepfake detection, particularly in scenarios involving face masks. The core of the system is a novel architecture based on Inception-ResNet-v2[41], augmented with preprocessing stages, residual connections, and batch normalization. These enhancements not only improve classification accuracy but also enable more robust detection in the presence of face masks.

Furthermore, the project extends the capabilities of the system by incorporating an Xception model, which further boosts detection accuracy to an impressive 99.7%. This additional model strengthens the overall robustness of the system, ensuring more reliable identification of deepfake content across diverse scenarios.

To facilitate user interaction, a user-friendly front end is developed using Flask, featuring authentication features to ensure secure access to the detection functionality. This interface enables seamless interaction with the system, allowing users to test its capabilities efficiently and effectively. Overall, the proposed work represents a comprehensive approach to enhancing deepfake detection accuracy and usability, addressing critical challenges in the field of digital forensics and cybersecurity.

b) System Architecture:

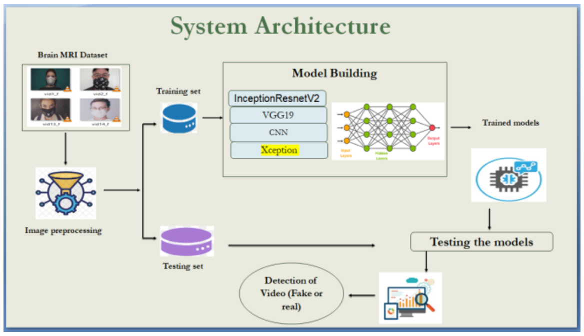


Fig 1 Proposed Architecture

The system architecture for deepfake detection comprises several key steps. It begins with the input dataset, consisting of videos containing both authentic and deepfake content. These videos undergo image processing to extract frames and enhance dataset quality. Subsequently, the dataset is divided into training and testing sets for model training and evaluation. Deep learning models are then constructed and trained on the training dataset to learn patterns indicative of deepfake content. Following training, the models are tested on the unseen testing dataset to assess their performance. Performance evaluation metrics such as accuracy, precision, recall, and F1-score are utilized to gauge model effectiveness. Finally, the trained models are deployed for real-time detection of deepfake content, enabling the identification and mitigation of potentially harmful manipulated media. This comprehensive architecture integrates data processing, model training, evaluation, and deployment to facilitate accurate and efficient deepfake detection.

c) Dataset:

The dataset utilized for training and testing the deepfake detection models consists of approximately 2000 videos, evenly split between fake and real content. Each category comprises 1000 videos, providing a balanced representation for effective model training. To ensure unbiased evaluation, the dataset is partitioned into an 80% training set and a 20% testing set.

Prior to model training, the video dataset undergoes preprocessing steps to prepare the input data. Initially, each video is decomposed into individual frames, facilitating frame-level analysis. Subsequently, facial detection algorithms are applied to identify and extract faces from each frame. The resulting face images are cropped to retain only the facial region of interest and resized to a standardized resolution of 128 × 128 pixels. This preprocessing ensures consistency and compatibility with the input requirements of the selected detection models, optimizing model performance and facilitating efficient training and evaluation processes. Overall, the dataset represents a diverse collection of authentic and manipulated videos, enabling comprehensive assessment of the models' detection capabilities across various scenarios and contexts.

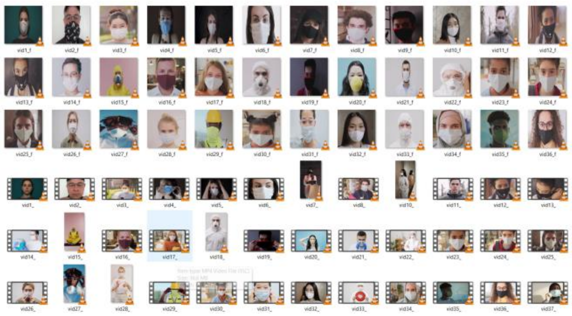


Fig 2 Dataset

d) Image Processing:

Converting Video into Frames: The first step in EDA involves converting each video in the dataset into a sequence of individual frames. This process is typically achieved using video processing libraries such as OpenCV in Python. Each frame represents a single snapshot of the video at a specific time instance, allowing for frame-by-frame analysis of the video content.

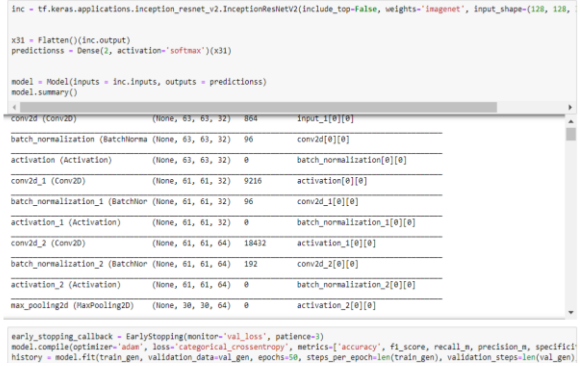
Image Reshaping: Once the videos are converted into frames, the next step is to reshape the images to a standardized size. Reshaping ensures uniformity in image dimensions, which is essential for consistency during subsequent analysis and model training. Commonly used image resizing techniques include scaling, cropping, or padding the images to achieve the desired dimensions.

ImageAugmentation: Imageaugmentation techniques are applied to introduce variability and diversity into the dataset, thereby improving the robustness and generalization capability of the models. Augmentation techniques may include random rotations, flips, shifts, zooms, or changes in brightness and contrast. These variations help prevent overfitting and enhance the model's ability to recognize patterns in unseen data.

By performing these image processing steps as part of EDA, researchers can gain insights into the characteristics and distribution of the dataset, identify potential challenges or anomalies, and preprocess the data appropriately for subsequent model training and evaluation.

e) Algorithms:

Inception ResNetV2: Inception ResNetV2 is a deep convolutional neural network (CNN) architecture that combines the Inception and ResNet modules. It is known for its exceptional performance in image classification tasks due to its ability to capture intricate features at multiple scales. Inception ResNetV2[41] incorporates residual connections and batch normalization, which help alleviate the vanishing gradient problem and improve training efficiency. Its architecture consists of multiple layers of convolutional and pooling operations followed by fully connected layers for classification.



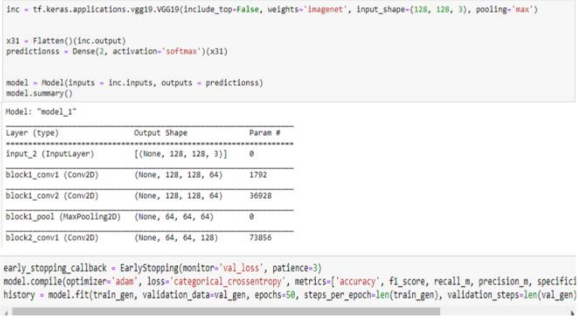
```
inc = tf.keras.applications.inception_resnet_v2.InceptionResNetV2(include_top=False, weights='imagenet', input_shape=(128, 128,


x31 = Flatten()(inc.output)
predictionss = Dense(2, activation='softmax')(x31)


model = Model(inputs = inc.inputs, outputs = predictionss)
model.summary()
```

```
conv2d (Conv2D)              (None, 63, 63, 32)   864         input_1[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 63, 63, 32)   96          conv2d[0][0]
__________________________________________________________________________________________________
activation (Activation)      (None, 63, 63, 32)   0           batch_normalization[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)            (None, 61, 61, 32)   9216        activation[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 61, 61, 32)   96          conv2d_1[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)    (None, 61, 61, 32)   0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)            (None, 61, 61, 64)   18432       activation_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 61, 61, 64)   192         conv2d_2[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)    (None, 61, 61, 64)   0           batch_normalization_2[0][0]
__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 30, 30, 64)   0           activation_2[0][0]
```

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=3)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy', f1_score, recall_m, precision_m, specifici
history = model.fit(train_gen, validation_data=val_gen, epochs=50, steps_per_epoch=len(train_gen), validation_steps=len(val_gen)
```

Fig 3 Inception ResNetV2

VGG19: VGG19 is a deep CNN architecture composed of 19 layers, including convolutional and pooling layers, followed by fully connected layers. It is characterized by its simplicity and uniformity, with small 3x3 convolutional filters and max-pooling layers. VGG19[42] has shown strong performance in image recognition tasks, making it a popular choice for various computer vision applications, including deepfake detection. Despite its simplicity, VGG19 exhibits remarkable feature extraction capabilities, making it suitable for detecting subtle patterns indicative of deepfake content.



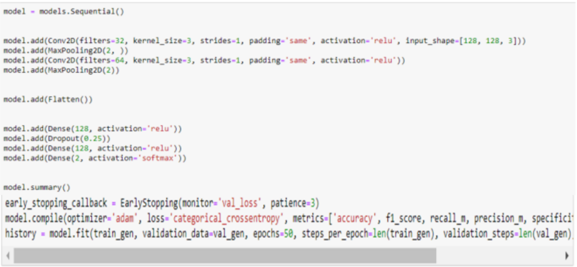
```
inc = tf.keras.applications.vgg19.VGG19(include_top=False, weights='imagenet', input_shape=(128, 128, 3), pooling='max')


x31 = Flatten()(inc.output)
predictionss = Dense(2, activation='softmax')(x31)


model = Model(inputs = inc.inputs, outputs = predictionss)
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 128, 128, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 128, 128, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 128, 128, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 64, 64, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 64, 64, 128)       73856
```

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=3)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy', f1_score, recall_m, precision_m, specifici
history = model.fit(train_gen, validation_data=val_gen, epochs=50, steps_per_epoch=len(train_gen), validation_steps=len(val_gen)
```

Fig 4 VGG19

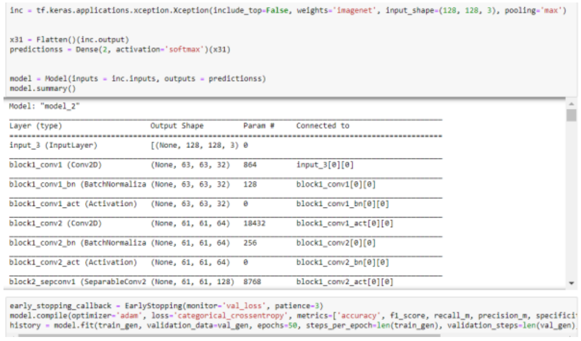
CNN (Convolutional Neural Network):CNNs are a class of deep neural networks designed for processing structured grid-like data, such as images. They consist of multiple layers of convolutional and pooling operations, followed by fully connected layers for classification. [11,27] CNNs leverage convolutional filters to extract hierarchical features from input images, enabling them to learn spatial hierarchies of features. CNNs have demonstrated excellent performance in various image classification tasks, including deepfake detection. By learning and identifying discriminative features from input images, CNNs can effectively distinguish between authentic and manipulated content.



```
model = models.Sequential()


model.add(Conv2D(filters=32, kernel_size=3, strides=1, padding='same', activation='relu', input_shape=[128, 128, 3]))
model.add(MaxPooling2D(2, ))
model.add(Conv2D(filters=64, kernel_size=3, strides=1, padding='same', activation='relu'))
model.add(MaxPooling2D(2))


model.add(Flatten())


model.add(Dense(128, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(128, activation='relu'))
model.add(Dense(2, activation='softmax'))


model.summary()
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=3)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy', f1_score, recall_m, precision_m, specifici
history = model.fit(train_gen, validation_data=val_gen, epochs=50, steps_per_epoch=len(train_gen), validation_steps=len(val_gen)
```

Fig 5 CNN

Xception : Xception is an of the Inception architecture that replaces the standard convolutional layers with depthwise separable convolutions. This modification aims to increase the model's efficiency and reduce the number of parameters while maintaining or improving performance. Xception's architecture allows for more efficient use of computational resources, making it particularly suitable for applications with limited computational capacity. By extending the capabilities of the Inception architecture, Xception enhances the robustness and accuracy of deepfake detection models, enabling more reliable identification of manipulated content.



```
inc = tf.keras.applications.xception.Xception(include_top=False, weights='imagenet', input_shape=(128, 128, 3), pooling='max')


x31 = Flatten()(inc.output)
predictionss = Dense(2, activation='softmax')(x31)


model = Model(inputs = inc.inputs, outputs = predictionss)
model.summary()
```

```
Model: "model_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            [(None, 128, 128, 3) 0
__________________________________________________________________________________________________
block1_conv1 (Conv2D)           (None, 63, 63, 32)   864         input_3[0][0]
__________________________________________________________________________________________________
block1_conv1_bn (BatchNormaliza (None, 63, 63, 32)   128         block1_conv1[0][0]
__________________________________________________________________________________________________
block1_conv1_act (Activation)   (None, 63, 63, 32)   0           block1_conv1_bn[0][0]
__________________________________________________________________________________________________
block1_conv2 (Conv2D)           (None, 61, 61, 64)   18432       block1_conv1_act[0][0]
__________________________________________________________________________________________________
block1_conv2_bn (BatchNormaliza (None, 61, 61, 64)   256         block1_conv2[0][0]
__________________________________________________________________________________________________
block1_conv2_act (Activation)   (None, 61, 61, 64)   0           block1_conv2_bn[0][0]
__________________________________________________________________________________________________
block2_sepconv1 (SeparableConv2 (None, 61, 61, 128)  8768        block1_conv2_act[0][0]
```

```
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=3)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy', f1_score, recall_m, precision_m, specifici
history = model.fit(train_gen, validation_data=val_gen, epochs=50, steps_per_epoch=len(train_gen), validation_steps=len(val_gen)
```

Fig 6 Xception

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

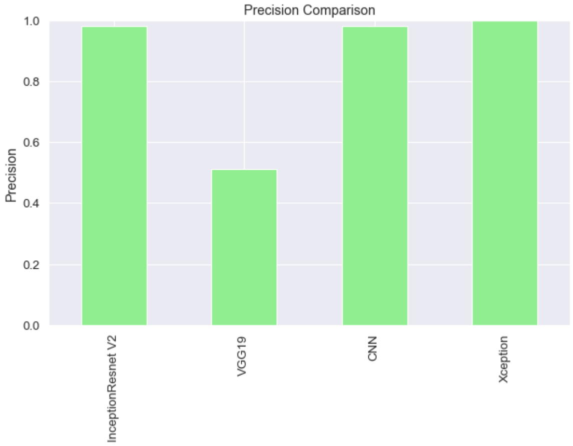


Fig 7 Precision Comparison Graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

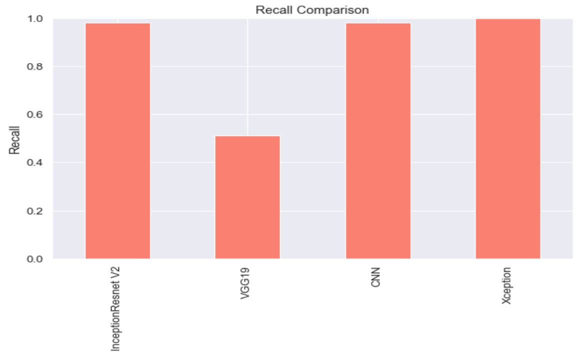


Fig 8 Recall Comparison Graph

F1-Score:F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$\textbf{F1 Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

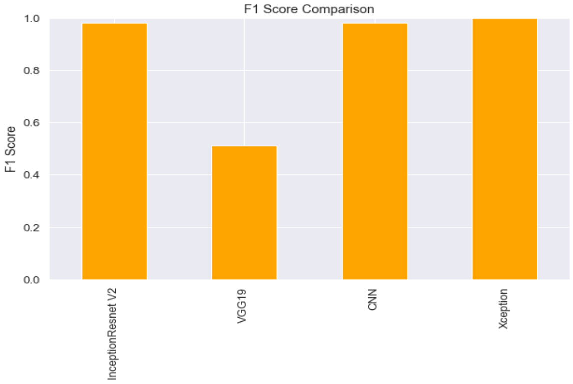


Fig 9 F1 Score Comparison Graph

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

Accuracy = TP + TN TP + TN + FP + FN.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

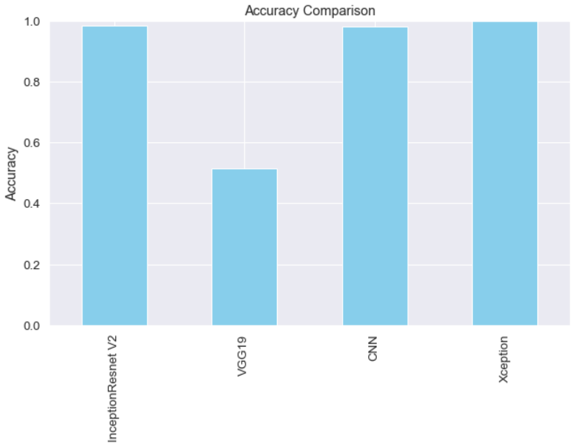
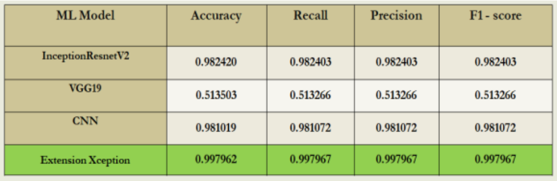
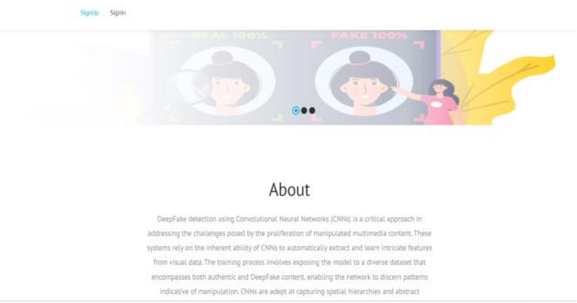


Fig 10 Accuracy Comparison Graph



ML Model	Accuracy	Recall	Precision	F1 - score
InceptionResnetV2	0.982420	0.982403	0.982403	0.982403
VGG19	0.513503	0.513266	0.513266	0.513266
CNN	0.981019	0.981072	0.981072	0.981072
Extension Xception	0.997962	0.997967	0.997967	0.997967

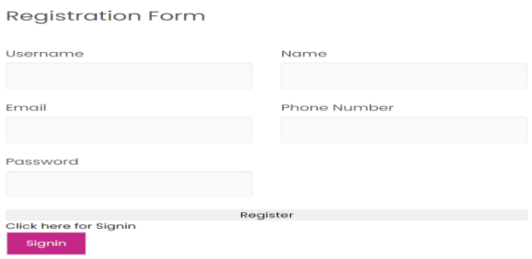
Fig 11 Performance Evaluation Table



About

DeepFake detection using Convolutional Neural Networks (CNNs) is a critical approach in addressing the challenges posed by the proliferation of manipulated multimedia content. These systems rely on the inherent ability of CNNs to automatically extract and learn intricate features from visual data. The training process involves exposing the model to a diverse dataset that encompasses both authentic and DeepFake content, enabling the network to discern patterns indicative of manipulation. CNNs are adept at capturing spatial hierarchies and abstract

Fig 12 Home Page



Registration Form

Username

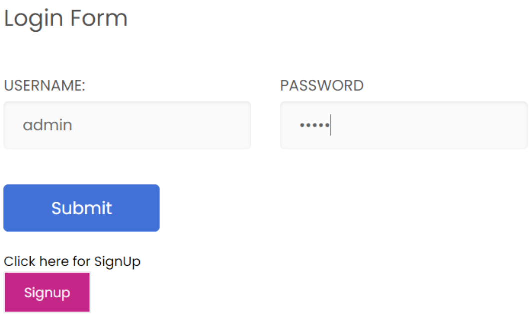
Name

Email

Phone Number

Password

Fig 13 Registration Page

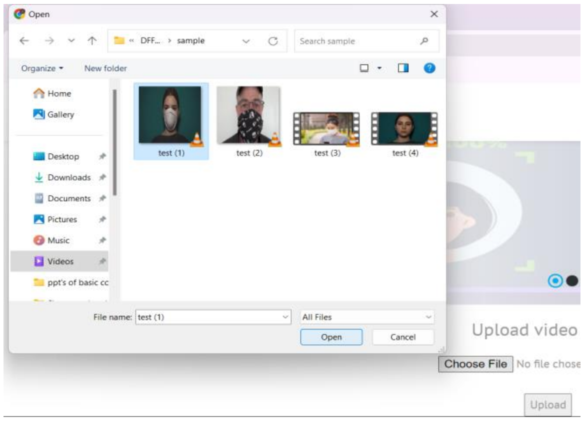


Login Form

USERNAME: admin

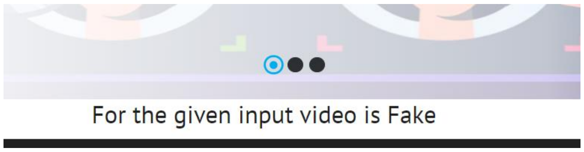
PASSWORD: •••••

Fig 14 Login Page



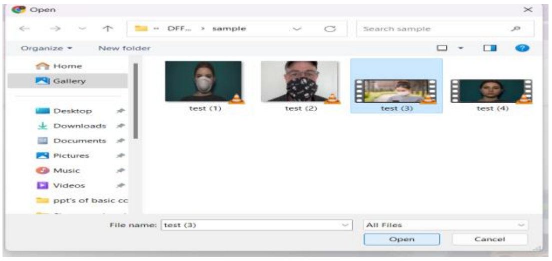
File name: test (1)

Fig 15 Upload Input Image



For the given input video is Fake

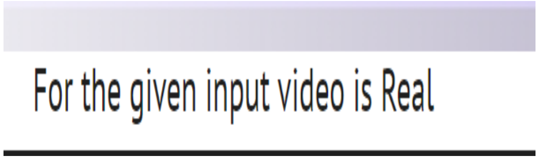
Fig 16 Predicted Result



File name: test (3)

All Files

Fig 17 Upload Input Image



For the given input video is Real

Fig 18 Final Outcome

5. CONCLUSION

In conclusion, the project demonstrates the efficacy of advanced deep learning architectures, including InceptionResNetV2[41], VGG19[42], CNN[11,27], and the Xception extension, in detecting Deepfakes with high accuracy. The incorporation of the Xception model achieves exceptional performance, showcasing an accuracy rate of 99.7% and affirming its reliability in real-world scenarios. Additionally, the integration of Flask with SQLite enhances user experience by providing a secure and intuitive web interface for interaction with the detection system.

Furthermore, the project lays the groundwork for future research by proposing continued experimentation to enhance detection accuracy, recognizing the dynamic nature of Deepfake technology. By effectively addressing the challenges posed by Deepfakes, particularly amidst the COVID-19 pandemic and face mask usage, the project contributes significantly to advancing digital security and mitigating the risks associated with the malicious manipulation of synthetic media. This comprehensive approach marks a significant step towards safeguarding against the detrimental effects of Deepfake proliferation in society.

6. FUTURE SCOPE

Future iterations of the project could expand the dataset to encompass a broader range of deepfake variants, including audio and multi-modal deepfakes, to enhance training and detection capabilities comprehensively. Optimizing the detection model for real-time processing could enable swift identification of deepfake content in live streams and video conferences, bolstering digital security and trust in online communication platforms. Continued research and development efforts should focus on updating the detection model to counter emerging threats effectively. Integration with forensic tools could enhance its utility for law enforcement and digital forensic investigators in combating digital manipulation and fraud.

REFERENCES

[1] DeepFakes Software. Accessed: Aug. 20, 2022. [Online]. Available: https://github.com/deepfakes/faceswap

[2] A Denoising Autoencoder + Adversarial Losses and Attention Mechanisms for Face Swapping. Accessed: Aug. 20, 2022. [Online]. Available: https://github.com/shaoanlu/faceswap-GAN

[3] DeepFaceLab is the Leading Software for Creating DeepFakes. Accessed: Feb. 24, 2022. [Online]. Available: https://github.com/iperov/ DeepFaceLab

[4] Larger Resolution Face Masked, Weirdly Warped, DeepFake. Accessed: Feb. 24, 2022. [Online]. Available: https://github.com/dfaker/df

[5] N. J. Vickers, ''Animal communication: When I'm calling you, will you answer too?'' Current Biol., vol. 27, no. 14, pp. R713–R715, Jul. 2017.

[6] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, ''DeeperForensics1.0: A large-scale dataset for real-world face forgery detection,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2889–2898.

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, ''StarGAN: Unified generative adversarial networks for multi-domain imageto-image translation,'' in Proc. IEEE Conf. Comput. Vis. pattern Recognit., Jun. 2018, pp. 8789–8797.

[8] T. Karras, T. Aila, S. Laine, and J. Lehtinen, ''Progressive growing of GANs for improved quality, stability, and variation,'' 2017, arXiv:1710.10196.

[9] T. Karras, S. Laine, and T. Aila, ''A style-based generator architecture for generative adversarial networks,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4401–4410.

[10] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, ''First order motion model for image animation,'' in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–11.

[11] A. S. Uçan, F. M. Buçak, M. A. H. Tutuk, H. İ. Aydin, E. Semiz, and S. Bahtiyar, ''Deepfake and security of video conferences,'' in Proc. 6th Int. Conf. Comput. Sci. Eng. (UBMK), Sep. 2021, pp. 36–41.

[12] N. Graber-Mitchell, ''Artificial illusions: Deepfakes as speech,'' Amherst College, MA, USA, Tech. Rep., 2020, vol. 14, no. 3.

[13] F. H. Almukhtar, ''A robust facemask forgery detection system in video,'' Periodicals Eng. Natural Sci., vol. 10, no. 3, pp. 212–220, 2022.

[14] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, ‘‘The deepfake detection challenge (DFDC) preview dataset,’’ 2019, arXiv:1910.08854.

[15] P. Yu, Z. Xia, J. Fei, and Y. Lu, ‘‘A survey on deepfake video detection,’’ IET Biometrics, vol. 10, no. 6, pp. 607–624, Nov. 2021.

[16] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, ‘‘WildDeepfake: A challenging real-world dataset for deepfake detection,’’ in Proc. 28th ACM Int. Conf. Multimedia, Oct. 2020, pp. 2382–2390.

[17] S. R. Ahmed, E. Sonuç, M. R. Ahmed, and A. D. Duru, ‘‘Analysis survey on deepfake detection and recognition with convolutional neural networks,’’ in Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robot. Appl. (HORA), Jun. 2022, pp. 1–7.

[18] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, ‘‘Celeb-DF: A large-scale challenging dataset for deepfake forensics,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 3207–3216.

[19] P. Korshunov and S. Marcel, ‘‘Vulnerability assessment and detection of deepfake videos,’’ in Proc. Int. Conf. Biometrics (ICB), Jun. 2019, pp. 1–6.

[20] J. Huang, X. Wang, B. Du, P. Du, and C. Xu, ‘‘DeepFake MNIST+: A deepfake facial animation dataset,’’ in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2021, pp. 1973–1982.

[21] H. Khalid, S. Tariq, M. Kim, and S. S. Woo, ‘‘FakeAVCeleb: A novel audiovideo multimodal deepfake dataset,’’ 2021, arXiv:2108.05080.

[22] J. Hu, X. Liao, W. Wang, and Z. Qin, ‘‘Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 3, pp. 1089–1102, Mar. 2022.

[23] A. Pishori, B. Rollins, N. van Houten, N. Chatwani, and O. Uraimov, ‘‘Detecting deepfake videos: An analysis of three techniques,’’ 2020, arXiv:2007.08517.

[24] Y. Li and S. Lyu, ‘‘Exposing DeepFake videos by detecting face warping artifacts,’’ 2018, arXiv:1811.00656.

[25] D. A. Coccomini, N. Messina, C. Gennaro, and F. Falchi, ‘‘Combining EfficientNet and vision transformers for video deepfake detection,’’ in Proc. Int. Conf. Image Anal. Process. Berlin, Germany: Springer, 2022, pp. 219–229.

[26] D. Wodajo and S. Atnafu, ‘‘Deepfake video detection using convolutional vision transformer,’’ 2021, arXiv:2102.11126.

[27] I. Amerini, L. Galteri, R. Caldelli, and A. D. Bimbo, ‘‘Deepfake video detection through optical flow based CNN,’’ in Proc. IEEE/CVF Int. Conf.

Comput. Vis. Workshop (ICCVW), Oct. 2019, pp. 1–3.

[28] A. Singh, A. S. Saimbhi, N. Singh, and M. Mittal, ‘‘DeepFake video detection: A time-distributed approach,’’ Social Netw. Comput. Sci., vol. 1, no. 4, pp. 1–8, Jul. 2020.

[29] B. Xu, J. Liu, J. Liang, W. Lu, and Y. Zhang, ‘‘DeepFake videos detection based on texture features,’’ Comput., Mater. Continua, vol. 68, no. 1, pp. 1375–1388, 2021.

[30] D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horváth, E. Bartusiak, J. Yang, D. Guera, F. Zhu, and E. J. Delp, ‘‘DeepFakes detection with automatic face weighting,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 668–669.

[31] E. Tjon, M. Moh, and T.-S. Moh, ‘‘Eff-YNet: A dual task network for DeepFake detection and segmentation,’’ in Proc. 15th Int. Conf. Ubiquitous Inf. Manage. Commun. (IMCOM), Jan. 2021, pp. 1–8.

[32] D. Güera and E. J. Delp, ‘‘Deepfake video detection using recurrent neural networks,’’ in Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS), Nov. 2018, pp. 1–6.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ‘‘Generative adversarial nets,’’ in Proc. Adv. neural Inf. Process. Syst., vol. 27, 2014, pp. 1–9.

[34] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, ‘‘MoCoGAN: Decomposing motion and content for video generation,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1526–1535.

[35] C. Vondrick, H. Pirsiavash, and A. Torralba, ‘‘Generating videos with scene dynamics,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 1–9.

[36] M. Saito, E. Matsumoto, and S. Saito, ‘‘Temporal generative adversarial nets with singular value clipping,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2830–2839.

[37] Y. Mirsky and W. Lee, ‘‘The creation and detection of deepfakes: A survey,’’ ACM Comput. Surv., vol. 54, no. 1, pp. 1–41, 2021.

[38] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, ‘‘DeepFakes and beyond: A survey of face manipulation and fake detection,’’ 2020, arXiv:2001.00179.

[39] S. McCloskey and M. Albright, ‘‘Detecting GAN-generated imagery using saturation cues,’’ in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 4584–4588.

[40] S. Suratkar, E. Johnson, K. Variyambat, M. Panchal, and F. Kazi, ‘‘Employing transfer-learning based CNN architectures to enhance the generalizability of deepfake detection,’’ in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2020, pp. 1–9.