Detection of Deepfake Videos

Dr. Mamatha H R
Chairperson, Dept. of CSE
PES University, Bangalore, India
mamathahr@pes.edu

Abbu Bucker Siddique
Dept. of CSE
PES University, Bangalore, India
abbubucker124@gmail.com

Hithesh D Nayak
Dept. of CSE
PES University, Bangalore, India
hitheshdnayak@gmail.com

Mohammed Anas Danish
Dept. of CSE
PES University, Bangalore, India
mohammedanasdanish41@gmail.com

Thejas N U
Dept. of CSE
PES University, Bangalore, India
thejasnu@gmail.com

***Abstract*—In our pursuit to counter the looming challenge of deepfake videos, we've engineered a robust architecture that serves as a sentinel of truth. Our deepfake model commences with video input, which undergoes a meticulous transformation journey. First, frames are meticulously extracted, laying the groundwork for data preprocessing. Through strategic manipulation, noise reduction, and resolution enhancement, our preprocessing stage refines the data for subsequent analysis.**

As we ascend, feature extraction becomes our compass. This step is pivotal, as it distills the essence of each frame, capturing nuanced patterns that can discern genuine from manipulated content. Our model's core strength resides in the preprocessing and feature extraction unlike the existing CNN based models, where we extract the features before passing it on to the neural network. Also the differentiating factor is that our model has been trained on combined dataset which includes 3 datasets, so that the model is robust against different deepfake generation methods.

When confronted with a video frame, our model undertakes a lightning-quick assessment, sifting through layers of learned insights. The culmination of this meticulous analysis results in a binary classification - a decision that definitively classifies the frame as real or deepfake .

This efficient process underscores our dedication to upholding authenticity in an era where technology tests the boundaries of reality.

***Index Terms*—prepossessing, feature extraction,CNN,Binary classification**

I. Introduction

The deepfake Techniques refer to the forgery methods that are used on images or Videos of a person using the deep learning methods like Attention Nets,deep-CNN,deep-RNN,Transformers,etc.There are many cases where this innovative Idea is Used on Images or Videos or celeb reties or any other famous personal, and is used to Spread rumors on the person and disturb his life. Therefore there has been many tries to detect these life-critical or mission-critical deepfakes.In This paper we will be concentration on Video form of Deepfakes.

Currently, there are three major types of Deep Fake videos.

- Head puppetry involves creating a video in which a person's entire head and upper shoulders are seamlessly combined from a source individual's video to make the target person appear to mimic the behaviors of the source.
- Face swapping entails the creation of a video featuring the subject, wherein their original facial features are substituted with artificially generated faces from the source, all while preserving the identical facial expressions.
- Lip syncing involves the fabrication of a video by exclusively altering the movements of the lips, thereby making it seem as if the subject is saying something they never actually uttered.

Initially This was tried only on Images, it was binary classification and then multi-class-classification.The concept of how to convert a video into frames and then use them as sequentially linked individual images and then process using a CNN,RNN combination was introduced,the later papers have also used the newly recognized Landmark Detection Algorithms and use only the landmarks to classify the image. Using both the idea of main-landmarks and closely connected landmarks, Attention model was introduced which performed very well. The Main principle of Attention model was to concentrate on closely related features of the input may it be sentences or image and then use this information in the future.

By Now most of the papers had few common research Gaps which were Orientation of Face in images did not provide accurate results , Occlusions of images as well as the video quality was a main cause for drop in Accuracy and in this paper we would be Concentrating more on the Orientation of Images and Create a Architecture with various technologies and components such as bi-directional LSTM, Pre-trained ResNet weights, which would provide us with a model which is less prone to orientation of Face in images.

In our approach to detect deepfake videos , we have used 3 explicit well defined and famous feature extraction techniques to capture nuanced patterns,that will help us differentiate the real from the fake one's.The 3 feature extractors used are **HOG** (Histogram of Oriented Gradient) , **SIFT** (Scale-Invariant Feature Transform), **LBP** (Local Binary Patterns) After the three feature extractors extract features parallely, we combine them to 32 features since all the 3 feature extractor return with different number of features. Later these are combined to 96 weights which undergoes a linear layer with a RELU activation function to reduce it further to 48,then to 24 and then

finally undergoes the sigmoid activation function to generate one single weight.Later this is passed to the neural network to predict the final output.

II. Related Work

A. Image deepfake detectors Based on number of outputs:-

Binary Classifiers:-Deepfakes could take either a video form or image form, initially the deepfake detection was done on images only, and many such methods used the binary classification on the images (real or fake)[13].most of the models use a CNN based architecture which will encode the image into a long 1D vector and then use each of the neuron value to be used in a Sigmoid Logistic regression unit to output either 1 or 0.Next the deepfakes we forged with very high resolution and were of a very good quality,Using a Meso-Net architecture would be helpful as the dense hidden layers will capture minute information which is underlayed in the potential forged image[4], this was also a Binary Classifier.

Multi-Class Classifiers:-But as obvious as it seems, deepfake images in training sets are different from the deepfakes in real world as they come from different distribution and may have undergone different forging techniques.For this reason many papers have done a multi-class-classification on data.

B. Landmark Detection Algorithms Used in Deepfake Detection Algorithms:-

Traditional approach:-Several survey papers have discussed the utilization of landmark detection algorithms in conjunction with the selection of critical facial landmarks, such as the eyes, nose, and cheeks, to serve as input for various neural networks [5]. This approach is motivated by the observation that the majority of significant facial changes primarily occur within the aforementioned regions, while landmarks like hair, forehead, and chin tend to undergo comparatively fewer alterations. Consequently, this strategy offers a distinct advantage, as it allows for the reduction of potential over fitting of the model to the training data set, by focusing solely on the major landmarks that exhibit substantial variations.

State-of-art Approach:-SIFT[22] is one of the widely used models which is based on the landmark detection feature. Unlike the tradition landmark detection algorithm it does not mark eyes,nose and others explicitly but rather learn which all are the main landmarks and in SIFT these important landmarks are called key-Points.during testing only the learned landmarks are located in the test image and then classified based on those.

Data-Augmentation Approach:-Here the landmark Detection algorithm is used to mark the landmarks on training data faces and then add another image to the set without the landmark[10], by doing this to all the training data we would reduce the over fitting in the training.

C. Unsupervised Learning Algorithms:-

Clustering techniques are used the most based on the photo-Response Non-Uniformity and the noise-Print Feature.

D. Multiple Training(Ensemble):-

We Notice that some models perform better on one set of conditions and do not perform well on other. But a model with different hyper-parameters can perform better on the latter with a trade-off with the former, such cases will have a very small efficient region but Some papers have gone further and ensemble different models to perform well on their respective tasks and would have a larger efficient region where in the model can be used.[3][9][12][19][20][18]

efficient-Net and Vision transformers for DFD:- Convolution cross ViT builds upon Efficient Vit and multi-scale transformer architecture. It features 2 branches , S branch and L branch so that it can focus on both local and global details.

face-identification-learning and face-context-learning:- It leverages two recognition networks, Ef and Ec, to extract identity information from the face and context images.The training process is multi-stage. Initially, the individual classifiers (Es, Er, Ef, and Ec) are pre-trained on relevant datasets. Later, the weights of Ef and Ec are frozen to ensure that identity cues remain dominant. The final classification network, D, is trained using the concatenated feature vector, with fine-tuning of Es and Er as needed.

Sharp Multiple Instance Learning for DFD:- spatial and temporal instances are extracted with corresponding encoding branches to form spatial-temporal bags with different temporal kernel sizes. S-MIL is performed on these bags to get the final fake scores of all bags, which can derive a final fake score for the whole video.

Transformer-Based Feature Compensation and Aggregation for Deep-Fake Detection:-Multi-head Clustering Projection (MCP) and Frequency-guided Fusion Module (FFM) to hierarchically aggregate features from all layers.A Locality Compensation Block (LCB) is proposed to compensate local features for transformers, where global transformer and local convolutional features are fused using a Global-Local Cross-Attention (GLCA)

hybrid technique based on Inception Res-net V2 and Xception:-The residual connections have been utilized to handle the degradation problem and have even helped reduce the training time by fifty percent. The output from the inception model is added to the current input connections of the residual network. The Inception network has three modules, A, B, and C, to form the entire network. The pooling layer will be replaced by a one-by-one convolution layer along with the residual connection network.

E. Sequence Models:-

Understanding that frames of a video would not be independent but related both in previous and next frames, led to creation of models with LSTMs GRUs ,specifically bi-directional[17].

F. Attention Models:-

A new Model was created which took the Correlation between different parts of the image along with the usual image features.This model concentrates more on the closely

related parts of the input rather than all combinations or groups which may include fairly unrelated components.

Spatial and temporal Attention mechanisms:-[1] The spatial attention component is designed to capture spatial disharmony and focus on shallow features. It re-calibrates shallow feature maps using attention maps generated by the mechanism.Temporal attention focuses on capturing temporal inconsistencies between consecutive frames in deepfake videos. It guides relatively high-level semantic features.

multi-attention mechanisms:-[6] Data-Scientists went further and found that along with the above mentioned Long-distance attention mechanism and also some extra not closely related parts of the image the model would perform better, Using multiple parts which are related to a region of image not only which comes in the first rank in correlation but few other top ranks were used.

Meta-Deepfake Detection:-[2]using Efficient-Net-B0 as the backbone architecture, it is combined with the Meta-learning which uses a Pairwise Attention Loss. It aims to improve the generalization of the model, making it effective in detecting deepfakes across various datasets and unseen domains, which is essential for addressing the evolving challenges posed by deepfake technology. The objective of this model is to create a versatile solution capable of directly addressing novel, previously unseen domains without necessitating model updates. The MDD algorithm accomplishes this by assigning varying weights to facial images from diverse domains. Specifically, MDD employs a meta-weight learning approach to transfer knowledge from source domains to target domains through meta-optimization steps. This process aims to enable the model to generate effective representations for both source and target domains. To construct multi-domain datasets, a meta splitting strategy is employed, resulting in the creation of meta-train and meta-test sets. The model then utilizes these sets to determine gradient descent and perform back propagation. The gradients from both inner and outer loops are combined to update the model, thereby improving its ability to generalize. By introducing pair-attention loss and average-center alignment loss, the system's detection capabilities are significantly enhanced. Furthermore, to assess its effectiveness and compare its generalization against several baselines, we conducted evaluations using benchmarks derived from various well-known deepfake datasets.

III. METHODOLOGY

The intent of the paper is to concentrate more on classifying the video as deepfake or not, even if the video has faces with different orientations (person seeing left or right,etc). Most of the surveys have mentioned in their limitations that the model would loose the efficiency if there is orientation factor in the image ,So we have tried to reduce that dependency for better results in Deepfake classification ,using various components of the architecture which are as follows:

A. Data-Set Collection and Merging

To ensure the quality and prediction accuracy of deep learning models, proper dataset preparation is crucial. In our research, we leverage 3 top datasets **DFDC**, **Celeb-DF** and **FaceForensics++**, which comprise both authentic and manipulated videos, along with corresponding authenticity labels. These videos are combined and then sampled to extract individual frames.

The **DFDC**(Deepfake Detection Challenge) is a is a dataset for deepface detection consisting of more than 100,000 videos.The full dataset comprises over 124k videos featuring 8 facial modification algorithms. These are sourced from 3,426 paid actors, produced with several Deepfake, GAN-based, and non-learned methods.It contains diverse subjects, backgrounds, and lighting conditions, making it a valuable resource for training and testing deepfake detection algorithms.

The **Celeb-DF** data set has 5,639 high-quality Deep Fake videos of celebrities, Which is a plus as most of the deepfakes are used to disturb the dignity of famous people. This Data set is generated using the improved synthesis process[22]. From an input video, faces of the target are detected, from which facial landmarks are further extracted. The landmarks are used to align the faces to a standard configuration. The aligned faces are then cropped and fed to an auto-encoder to synthesize faces of the donor.

FaceForensics++ is a forensic dataset that encompasses 1000 original video sequences. The dataset is compiled from 977 YouTube videos, with each video featuring a clearly visible mostly frontal face without any obstructions. This characteristic facilitates the generation of convincing forgeries through automated tampering methods.The manipulations techniques include four state-of-the-art methods, namely, Face2Face, Face Swap, Deep Fakes, and Neural Textures:

1) Face2Face- is a system for facial reenactment that preserves the identity of the target person while transferring the expressions from a source video. The system relies on two video input streams and involves manual selection of key frames from both videos. These selected frames are then employed to create a detailed facial reconstruction, enabling the re-synthesis of the face with different expressions and lighting conditions.
2) Face Swap- is a graphics-oriented technique that involves transferring the facial region from a source video to a target video by utilizing sparsely detected facial landmarks to extract and adapt the face region. This adaptation is achieved through the application of a 3D template model and blend shapes.
3) Deep Fakes - Deepfakes, both a term and a specific manipulation method, involve replacing a face in a target video with one from a source video or image collection using shared encoder autoencoders. The process includes face detection, alignment, and Poisson image editing. The FaceSwap GitHub implementation is used for dataset creation, with some modifications, and pre-trained models are provided to simplify the process. This

enables additional manipulations of individuals with post-processing.

4) NeuralTextures - exemplified by Thies et al.[23], utilizes original video data to train a neural texture and rendering network for a target person. The training process involves photometric and adversarial losses, employing a patch-based GAN-loss similar to Pix2Pix. Tracked geometry, generated using Face2Face's tracking module, is employed, with modifications focused on the mouth region for facial expression adjustments.

Combining the datasets gave a regularizing effect as the frames of Videos come from different datasets, the model won't be over fitting to one particular data set.The other advantage that we found from combining the frames from the different datasets is that there are now significantly large amount of frames in the resultant data set and from the survey we understood that increasing the number of images or videos in data set would increase the accuracy, we also learnt that increasing the data set size excessively would result in reduced overall accuracy.Also the randomness is introduced while selecting frames for the training phase as we wanted our model to be robust and work on most of the datasets without biasinf for a single dataset.

B. Frame-Extraction

The Architecture takes in input in image or frame format, thus this stage is used to convert the input Video into frames. This Stage uses the Dlib Library of python, for frames extraction.for few videos the number of frames is very high and for few its small number so we have also implemented a Helper function to select almost equal number of frames per video and to reduce the total number of frames per video.

C. Data-Prepossessing

The Surveys tell that the image quality which is going in as input must be improved, and some papers have also put this in their limitation section, therefore the architecture uses a preprocessing stage which will enhance the image quality and also do down-sampling for regularization.The steps involved here are converting the color image to gray scale, down-sampling the pixels to (64,64).

D. Feature Extraction

We have introduced this particular stage to answer our needs on the orientation of face in frames.A deep neural Network has a characteristic of doing feature Extraction implicitly, but from the survey this type of implementation would have the above mentioned research gap, so We are getting arrays which will have Features of the frame and then pass this instead of the image. Experimentally we understood that combining more than one feature extraction would give a better result, hence we used ,**HOG**(Histogram of Oriented Gradients),**LBP**(Local Binary Patterns),**SIFT**(Scale-Invariant Feature Transform).

HOG - is a feature extraction technique used in computer vision for object detection and image classification.The HOG algorithm computes gradient information from the image to create histograms of oriented gradients in local regions.The key parameters for HOG feature extraction include the size of the cell, the size of the block, and the number of bins in the histograms. HOG computes a feature vector for each frame based on these histograms.By using HOG as an explicit feature extraction technique, we can capture the image's gradient information and create a feature vector that represents the distribution of gradients across the frame. This feature vector combines with the other 2 and is used as input to our model for deepfake detection.there are certain limitations and challenges when using Histogram of Oriented Gradients (HOG) for deepfake detection. Here are a few:

- *Limited to Local Gradient Information*: HOG primarily captures local gradient information within an image. While this information can be useful for detecting certain patterns and structures, it may not capture higher-level semantic information that could be relevant for deepfake detection. Deepfake techniques often involve subtle manipulations that may not be fully captured by HOG alone.
- *Vulnerability to Image Transformations*: HOG is sensitive to image transformations such as rotation, scale changes, and affine transformations. This sensitivity can make the feature extraction process less robust when deepfake videos involve different transformation techniques. Augmenting the dataset with transformed versions of the frames or applying additional preprocessing steps may help mitigate this limitation.
- *Fixed Grid Structure*: HOG divides the image into fixed-size cells and computes gradients within each cell. This fixed grid structure may not be optimal for capturing local patterns that vary in scale or size. Deepfake techniques can involve manipulations at different scales, making it challenging for HOG to capture fine details in all cases.
- *Limited Discriminative Power*: While HOG can capture local texture and edge information, it may struggle to distinguish between real and fake frames when the differences are subtle or when deepfake videos are of high quality. Deepfake techniques are continuously evolving, and sophisticated methods can generate visually compelling results that are difficult to detect using simple local gradient features.

To overcome these limitations and enhance deepfake detection,we have combined HOG with SIFT and LBP feature extraction techniques. By integrating multiple feature extraction methods and incorporating temporal analysis, it is possible to improve the overall performance and robustness of deepfake detection systems.

SIFT - is a feature extraction method which will convert the frame to a latent representation which will have information about key points of the frame.The Key points are the points which tell the characteristics of the face in the frame that is those are unambiguous set of points, the extractor will discard all the other ambiguous points.There are few steps included in the SIFT extractor.

1) *Scale space extrema detection* - Scale space extrema

detection in the SIFT detector involves identifying key points at varying scales by searching for local minima and maxima in the difference-of-Gaussian (DoG) pyramid. The algorithm convolves the image with Gaussian filters at different scales and subtracts adjacent scales to create the pyramid. Key points are detected as extremal values in this pyramid, enabling the selection of features that are invariant to scale changes.

2) *Keypoint localization* - Among all the choosen points further filteration is done here. The number of points from the previous step is still large, so here one point is choosen from each neighbourhood with predefined loacality.This process is called keypoint localization. Candidates are chosen from extrema detection and low contrast outlier and poorly localized candidate along an edge are rejected in initial outlier rejection.
3) *Orientation assignment* - By establishing a consistent orientation for each key point based on local image characteristics, the key point descriptor can be expressed relative to this orientation. The key point's scale is employed to select the Gaussian smoothed image L at the most appropriate scale, ensuring that all computations are carried out in a manner that is invariant to scale.
4) *Keypoint descriptor* - In this step the gradient magnitudes and orientations are calcultated and then the weighted histogram is created for each neighbourhood, and then take the highest peak as the orientation of interest points.

LBP:

- the Local Binary Pattern classifier is used in deepfake detection as a texture analysis technique that captures texture patterns within an image, encodes them into a feature vector, and then uses machine learning to distinguish real images from manipulated ones.
- Its ability to handle variations in texture and illumination makes it a valuable component in the detection of deepfake content.
- After calculating LBP codes for all pixels, a histogram is constructed. Each unique LBP pattern represents a bin in the histogram, and its count (frequency) in the image serves as the value in that bin. The LBP histogram is typically used as a feature vector.
- This feature vector is invariant to changes in illumination, which makes it particularly useful for detecting inconsistencies or artifacts introduced by deepfake techniques.

A drawback of the conventional LBP operator lies in its incapacity to capture prominent characteristics characterized by extensive structural patterns within its limited 3×3 neighborhood. To address this limitation and effectively handle textures across various scales, the operator was subsequently extended to accommodate neighborhoods of varying sizes.

E. Equalizing the number of features from each feature extractor

From the above feature extractor components we have now got 3 feature arrays. It So happens that the sizes of the feature array from the three feature extractors are different, which makes different feature extractors require different neural networks. To make it uniform after few layers we stop at (32*1), so the backtracking becomes easier too. We pass these on Linear Sequential Layers with ReLU activation function. The reason we are taking the same feature array size is that we need no biasing between features from different feature extractors.

- HOG - output feature array size - 1728*1
- SIFT - output feature array size - 254*1
- LBP - output feature array size - 10*1

To achieve this we have three pipelines for each of the feature extractor output which will have linear Sequential layers or a Vanilla Neural Network which will take in input as the output feature array and will output a 32*1 feature array. In between of the the input and output layers we will have experimentally tested number of hidden layers as follows.

Every Linear Layer (LL) is a LinearLayer-ReLU-BatchNorm()-Dropout(0.5).

HOG :-

1) feature array - 1728*1
2) LL(ReLU-Activation)-2048 Weights
3) LL(ReLU-Activation)-1024 Weights
4) LL(ReLU-Activation)-512 Weights
5) LL(ReLU-Activation)-256 Weights
6) LL(ReLU-Activation)-128 Weights
7) LL(ReLU-Activation)-64 Weights
8) LL(ReLU-Activation)-32 Weights

SIFT :-

1) feature array - 254*1
2) LL(ReLU-Activation)-128 Weights
3) LL(ReLU-Activation)-64 Weights
4) LL(ReLU-Activation)-32 Weights

LBP :-

1) feature array - 10*1
2) LL(ReLU-Activation)-32 Weights

This will now be passed to the combining unit.

F. Combining weights from different layers of different feature extractors

Now we have passed the features arrays on linear layers to three arrays of smaller size and equal size, then we will be merging all the three to form a 96*1 matrix and then passing it to the next component in our architecture.

G. Neural Network for the combined feature matrix

After getting the merged feature matrix we will pass that to a neural network which will have input as the combined array and then will have a number of hidden layers and then will have the the output layer with only one output.which will give the desiered output whether the input image is a real image or a deepfake image.

1) combined feature array - 96*1
2) LL(ReLU-Activation)-48 weights
3) LL(ReLU-Activation)-24 weights

4) LL(Sigmoid-Activation)-1 weight

The Last Layer here is the output Layer

H. Optimizing Function and Loss Function

Optimizing Function :- Adam Optimizer with Learning Rate = 0.01. The Adam Optimizer is used because of two reasons ,one being theoretical reason which is that adam is combination of the Momentum and also the RMSprop Optimizer.Momentum will reduce the vertical movement of the convergence graph and increase the horizontal movement of the graph and the RMSprop will change the Learning rate as the epoch increases that is it adapts to the conditions.The second reason is experimental, we had got good results using this optimizer.

$$
\begin{aligned}
m &= \beta_1 \cdot m + (1 - \beta_1) \cdot \nabla J(\theta) \\
v &= \beta_2 \cdot v + (1 - \beta_2) \cdot (\nabla J(\theta))^2 \\
m_{\text{corrected}} &= \frac{m}{1 - (\beta_1)^t} \\
v_{\text{corrected}} &= \frac{v}{1 - (\beta_2)^t} \\
\theta &= \theta - \alpha \cdot \frac{m_{\text{corrected}}}{\sqrt{v_{\text{corrected}}} + \epsilon}
\end{aligned}
$$

Where:

- θ are the parameters being optimized.
- $\nabla J(\theta)$ is the gradient of the objective function J with respect to θ.
- α is the learning rate.
- β_1 and β_2 are exponential decay rates for moment estimates.
- m and v are the first and second moment estimates.
- t is the timestep.
- ϵ is a small constant to prevent division by zero.

Loss Function :- The Loss function used in out Architecture is the Binary Cross Entropy Loss. We have chosen to work with this particular loss function because the data that we are working with is a labeled data that we are doing supervised learning, this means we will have the correct label y and the expected value $\hat{y}$. The other reason why we have chosen this loss function is that the output is binary output that is either 0 or 1(binary Classification), that is y takes either 0 or 1 and $\hat{y}$ takes any real number between 0 and 1.

$$-(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}))$$

Where:

- y is the true label (either 0 or 1),
- $\hat{y}$ is the predicted probability of the positive class (value between 0 and 1).

I. overall Architecture

1)

TABLE I
TABLE TYPE STYLES

Architecture	Results
No feature Extraction	62 % Accuracy
Face Pose Estimator Features on NN	71 % Accuracy
SIFT Features on NN	73 % Accuracy
Local Binary Patterns Features on NN	75 % Accuracy
Preprocessed images directly on Deep VGG like CNN architecture	85 % Accuracy
Preprocessed images directly on Resnet	82 % Accuracy
Combined architecture of images on Resnet and LBP data on NN	84 % Accuracy
Capsule Network	85%
Mesonet	needs 53.16GB
SIFT+LBP+HOG on NN	96% accuracy