Forensic Science International: Digital Investigation 42-43 (2022) 301452

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

LBPNet: Exploiting texture descriptor for deepfake detection

Staffy Kingra*, Naveen Aggarwal, Nirmal Kaur

University Institute of Engineering and Technology, Panjab University, Chandigarh, India

ARTICLE INFO

Article history:
Received 5 August 2021
Received in revised form
7 September 2022
Accepted 10 September 2022
Available online 4 October 2022

Keywords:
Local Binary Pattern
Deepfake
LBP
Face swap
Texture
Deep network

ABSTRACT

Recent AI advancements made it significantly easier to generate high-quality synthesized faces, referred as deepfakes. It can make people saying and doing things that never happened actually. Owing to the vast usage of social media, manipulated content is susceptible to spread unrest in the society. With the continuous evolvement of AI-enabled deepfake generators, the research on deepake detection has started progressing in the last few years. From analyzing eye-blinking in the previous generation deepfakes to deep learning based models for advanced deepfakes, the existing deepfake detectors have analyzed numerous artefacts. This paper proposes a novel detection approach, LBPNet, to distinguish deepfaked faces from genuine ones by means of exploiting inconsistencies in texture information. In particular, Local Binary Pattern (LBP) of deepfaked and pristine faces have been investigated through a CNN based model. Moreover, the proposed LBPNet technique is evaluated on more advanced and diverse deepfaked datasets such as Celeb-DF, DFDC, and DeeperForensics, which provided a detection accuracy of 92.38%, 80% and 86% respectively. Comprehensive analysis on different benchmark datasets and comparison with state-of-art endorse the superior performance of the proposed method. Thorough experimentations also reveal the robustness of LBPNet against different compression levels and tampering types.

1. Introduction

Creation and sharing of manipulated media have become a common place in today's digital age. Digital images and videos possessing crucial information are prone to a great risk of being manipulated. Such digital data serves as a proof of evidence for crime investigations as well as to shape public perceptions. However, increasing ease of manipulating techniques make it hard to trust the integrity of multimedia content besides promoting the fake propaganda. Many multimedia manipulation methods such as copy-move, splicing, FRUC (Frame Rate Up Conversion) emerged in the early 90's to obscure reality. Tremendous advancement in the field of AI resulted in the emergence of GAN (Generative Adversarial Network) that later became the source of deepfake generation. Deepfake multimedia generated using deep architectures provided more realism to the fabricated content. The need for a large amount of data for deepfake generation through GAN was also alleviated by MocoGAN (Tulyakov et al., 2018). In addition, development of easy-to-use and sophisticated deepfake creation tools (e.g. REFLECT,[1] REFACE, MyHeritage,[2] FaceApp[3]) provided access of fake media generators to novice users too. On one side, these innovations brought appreciation for the research community while on the other hand, malicious usage of the same has raised serious concerns for an individual and society at large.

Since inception, deepfakes are evolving continuously, and are being generated through different mechanisms. Through deepfakes, target person is intended to re-enact things said by source person either by swapping the face or by direct mapping of expressions. Face swapping (Rössler et al., 1901), on one hand, replaces the face of source person with that of target person in an image or video retaining source person's expressions. On the other hand, face re-enactment (Rössler et al., 1803) directly alters face expressions of target person. Face manipulation can also be performed by employing some external attributes like spectacles, makeup, etc. on target person's image. Another kind of deepfake, Lip-sync deepfake (Suwajanakorn et al., 2017), intends to transform

* Corresponding author.
E-mail addresses: staffysk@gmail.com (S. Kingra), navagg@gmail.com (N. Aggarwal), nirmaljul19@gmail.com (N. Kaur).

[1] https://reflect.tech/.
[2] https://www.myheritage.com/.
[3] https://www.faceapp.com/.

https://doi.org/10.1016/j.fsidi.2022.301452
2666-2817/

lip movements of target person with respect to a particular audio recording. Apart from facial manipulation, whole body deepfakes (Chan et al., 2019) also called puppet-master deepfakes have emerged, wherein the whole body movements of source person are mapped to target person. Recently emerged audio deepfakes intends to clone the voice of target person through a deep learning based software (Jr, 2019).

Deepfakes became popular in 2017 when deepfaked pornographic videos of some popular hollywood actresses were shared in the cyberspace (Cole, 2017). Another deepfaked video went viral in 2018, wherein former U.S. President Barack Obama was purported to insult the president at that time (Vincent, 2018). Afterwards, a deepfaked video of Mark Zuckerberg was shared on social media wherein he was intended to say things that were not said by him (Posters, 2018). Recently, a puppet-master deepfake of Queen Elizabeth went viral by mapping the body movements of a source person to her (Rahim, 2020). Other than visual deepfakes, CEO of a European company was tricked by an audio deepfake in 2019 that resulted in a fraudulent transfer of $243000. In view of these state of affairs, it is pertinent that deepfakes can be as harmful to defame one's reputation by assassinating his/her character to create unrest in the society by faking celebrity speeches, to blackmail individuals for financial benefits, and to influence public opinion by spreading false information. Availability of various user-friendly deepfaking applications have worsened the situation even more. As per the 2020 report, deepfaked content on internet has doubled in mere six months (Hofesmann, 2020).

Considering the risk of digital information and deepfake generation softwares being misused, some deepfake detection techniques have been developed. The strategy of deepfake detection technique is to pick up a peculiar imperfection in the generated tampered content that helps to differentiate it from the real one. As advances in the deepfake detection artefact opens a new path for improvement, there is still a sufficient space to develop more efficient techniques capable enough to differentiate deepfaked content from an authentic one. Many datasets are proposed by researchers to evaluate the efficacy of deepfake detection techniques, majority of which were focused on face swap deepfake (Rössler et al., 1901; Li et al., 1909; Dolhansky et al., 2006; Yang et al., 2019a). In these datasets, face of source person in an image or a video is swapped with the face of target person using autoencoder based GAN architecture.

1.1. How deepfakes are created?

Deepfakes are the fake digital media which are generated using various deep learning architectures either by swapping face or by mimicking lip movement. Numerous GAN (Nirkin et al., 2019a, 2019b; Yan et al., 2018; Li et al., 1912) have been proposed for swapping a face of source person with that of target one, and thereby utilized to develop deepfake specified datasets. The first attempt to create face swap deepfake was performed using an autoencoder architecture in which a pair of autoencoders is trained with faces of source and target person. Here, source and target images are inputted to a common encoder to extract individual face artefacts. However, decoders are kept separate that takes encoded output as input and reconstructs a respective face. For deepfake generation, decoder trained on target face is attached at the end of common encoder. Here, source image is taken as an input for a shared encoder while a target-specific decoder maps the facial attributes of target person onto a reconstructed image. Fig. 1a demonstrates this deepfake generation process. Most of the open-source deepfake generation tools, e.g. Faceswap[4] and various published datasets (Rössler et al., 1901; Dolhansky et al., 2006) utilized this mechanism for deepfake creation. Another mechanism of deepfake generation, led by GAN's, utilizes a pair of adversarial neural networks as shown in Fig. 1b. One network from a pair is Generator that is used to generate deepfake images, and another network named as Discriminator is trained to differentiate generated images from pristine. Generator network gets random noise as input and outputs a real-looking image. Afterwards, generated images and real images are fed into discriminator for classification. Generator keeps on improving its output until discriminator starts misclassifying generated images as real.

Most of the deepfake generation process such as Faceswap (Rössler et al., 1901) tends to produce ghost artefacts, blurriness, and irregular contour around facial regions in the generated image. Many deepfake detection techniques are proposed in literature to investigate such artefacts. One of the approaches (Kohli and Gupta, 2021) analyzed face in frequency domain through CNN based model that provided an accuracy of 85.24% and 66.50% on deepfake videos of FF++ (Rössler et al., 1901) and Celeb-DF (Li et al., 1909) dataset. However, the performance was not found good for Neural-Textured videos of FF++ dataset. Then, PRRNet (Shang et al., 2021) was introduced that analyzed pixel-wise and region-wise similarity to exploit any local discrepancies in suspected image. These inconsistencies were analyzed at different resolutions through a pretrained network, HRNet (Sun et al., 1904). The model provided an accuracy of 95.63% and 80.01% on deepfake and neural textured videos respectively. In addition, the technique also performed well on Celeb-DF (Li et al., 1909) and DFDC-P dataset with an accuracy of 99.80% and 97.78% respectively.

One of the major limitations of state-of-art techniques is the lack of their evaluation on advanced deepfake datasets such as DeeperForensics (Kohli and Gupta, 2021; Shang et al., 2021; Khalil et al., 2021) and DFDC (Kohli and Gupta, 2021; Shang et al., 2021; Khalil et al., 2021). These advanced datasets exhibit more realistic deepfaked videos with very few visual inconsistencies. Deepfake detection technique for such videos is in great demand. However, prior works (Nguyen et al., 2021; Caldelli et al., 2021) have shown better performance on early generation datasets, such as FF++, as they exhibit visual artefacts that is easy to analyze. Also, research in deepfake detection always advanced in ameliorating the network complexity to improve its performance. To manage the complexity of deepfake detection model, training should be focused on some peculiar artefacts only. One such artefact can be the texture pattern detail of facial region that gets easily disrupted during deepfaking procedure, that is taken into consideration in this paper.

Recently, texture based LBP-coded histograms (Khalil et al., 2021) for deepfake detection is developed that reported an AUC of 76.8% on DFDC-P (Hofesmann, 2020) and 77.1% on Celeb-DF dataset (after fine-tuning). Along with LBP generated histograms, researchers also utilized features extracted from a modified version of HRNet (Sun et al., 1904), which increased the model complexity. Thereby, accuracy of this model was not found at par with its complex architecture. Although some other researchers (Arini et al., 2022; Wang et al., 2021; Akhtar and Dasgupta, 2019) proposed LBP-based technique for deepfake detection but were not found to be efficient. The technique proposed in (Arini et al., 2022) utilized a combination of Gaussian filtered and LBP-image to train Xception and ResNet50, and provided an accuracy of 79% on a subset of Celeb-DF dataset. Researchers in (Wang et al., 2021) proposed a new model by introducing some LBP-layers in between layers of ResNet and GramNet models while SVM classification was performed on different handcrafted features including LBP in (Akhtar and Dasgupta, 2019). However, these approaches are not tested on benchmark deepfake datasets such as DFDC, FF++, etc.

[4] https://github.com/wuhuikai/FaceSwap.

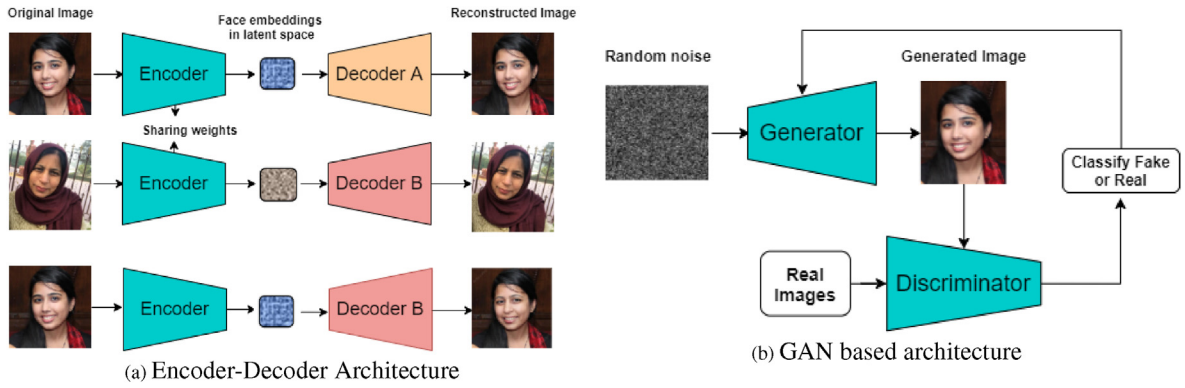


Fig. 1. Deepfake generation mechanisms.

Keeping in view of these challenges, this paper explored texture pattern inconsistencies of facial regions to differentiate deepfaked faces from the real ones. Owing to its computational simplicity and adequate performance in various computer vision (Xiao et al., 2018; Huang et al., 2011a) and face recognition (Zhang et al., 2005; Li et al., 2007) tasks, LBP based texture descriptor is utilized to analyze inconsistencies in facial texture. The proposed technique couples the benefits of extracted LBP based feature with a novel CNN based model. Since LBP coded image exhibits more information (location and intensity information) than LBP-coded histogram (only intensity information) as utilized in (Khalil et al., 2021), the proposed model is trained using LBP coded images. Fig. 2 demonstrates the difference between texture pattern of deepfaked faces and real faces for some benchmark datasets (Rössler et al., 1901; Dolhansky et al., 2006) through their respective LBP's. Closer look at texture pattern reveals that different regions of deepfaked faces are either blended with each other or the entire face is bulged out. Even on smooth faces, deepfaking can cause irregular texture pattern. However, texture pattern of pristine face exhibit clear boundaries along different face regions such as nose, eyebrow, eyes, lips, and fine lines. Performance of the proposed LBPNet technique is analyzed on advanced deepfake datasets such as DFDC, DeeperForensics, Celeb-DF, and FaceForensics with significant accuracy.

The major contributions of the proposed technique are summarized as follows:

1. This paper presents a novel deepfake detection solution that analyzes facial texture irregularities. For the purpose, a well known texture analysis method, LBP, is utilized and a deep learning based deepfake detection model, LBPNet, is proposed. The novelty of the proposed technique is the combination of LBP based texture features with LBPNet for classifying deepfaked faces that increased the learning ability of the proposed model.
2. The model is trained and tested on different datasets namely Deeper-Forensics, DFDC, Celeb-DF, and FF++ datasets, separately, for 25 epochs, and an accuracy of 83.9%, 75.7%, 87.56% and 97% is observed respectively. Performance of proposed model is further improved by considering textural inconsistency of multiple frames in a video that resulted in an accuracy of 86.4%, 80%, 92% and 99.1% on Deeper-Forensics, DFDC, Celeb-DF, and FF++ datasets respectively. Moreover, performance on multi-face videos of DFDC dataset is also reported.
3. The proposed model is also tested on the dataset developed using a user-friendly mobile application and deepfake animation technique (Siarohin et al., 2019) with a detection accuracy of 92.9% and 91.2%. Moreover, LBPNet is found good for both high quality and low quality videos.
4. To gain better insights of the proposed LBPNet, class activation maps are also presented. These activation maps make the model more explainable as to why it predicts a particular outcome.

Followed by an introduction of deepfakes, their impact and how they are created, major contributions of the paper are stated in Section 1. Afterwards, some state-of-art approaches are reviewed in Section 2. Section 3, then, provides a thorough discussion on proposed methodology which includes feature extractor and classification model for deepfaked face detection. The technique is evaluated on various benchmark datasets along with the self-created one, results of which are provided in Section 4. This section also reports an analysis of cross-dataset performance of the proposed technique followed by visualization of activation maps. Section 5, at last, concludes the paper along with a discussion on limitations of proposed approach and future scope.

2. Related work

Deepfake detection began in early 2018 by analyzing visual inconsistencies caused by deepfake generator in deepfake videos. From human-specific artefacts (Li et al., 1806) to generator-led artefacts (Guarnera et al., 2020), state-of-art approaches (Kingra et al., 2022; Nguyen et al., 2022) utilized different features for deepfake detection. Early-generation datasets like UADFV (Yang et al., 2019a),

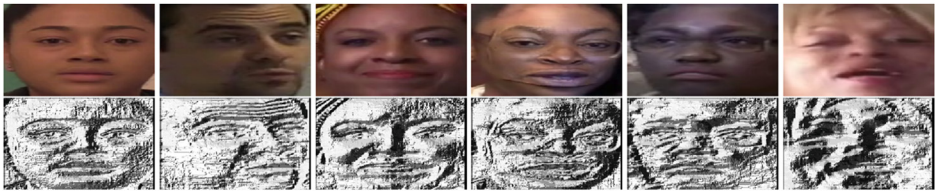


Fig. 2. Visual analysis of texture consistency of real and deepfaked faces. (First 3 columns contains real faces from DFDC (Dolhansky et al., 2006) dataset (1–2 rows) while the last 3 contain deepfaked faces.).

Deepfake-TIMIT (Korshunov and Marcel, 1812), and FF++ (Rössler et al., 1901) contain low quality deepfake videos which exhibit visual inconsistencies. Thereby, most of the techniques reported good results on these datasets (Afchar et al., 2018; Durall et al., 2020). Recently, Facebook, along with other reputed organizations, initiated a challenge named DFDC (Dolhansky et al., 2006) (DeepFake Detection Challenge) to contribute in this field and also released a dataset for the same. This dataset contains most realistic deepfaked videos but very few state-of-art approaches were tested on the same (Tan and LeEfficientnet, 1905; Mehra, 2020). Among the various deepfake detection solutions, some of the approaches analyzed specific visual cues while others performed automatic analysis of deepfake features through deep networks. This section categorizes possible solutions for deepfake detection based on the approach utilized.

2.1. Visual cues based deepfake detection approaches

The foremost technique (Li et al., 1806) analyzed eye-blinking frequency in suspected videos, a lack of which was observed in early deepfakes. The technique analyzed sequence of eyes using a combination of CNN (Convolutional Neural Network) and LSTM (Long Short Term Memory) architecture which resulted in an AUC of 99% on self-created dataset. Afterwards, a deep network, MesoNet (Afchar et al., 2018) was proposed using InceptionNet architecture, which analyzed image at mesoscopic and microscopic levels. The technique was tested on FaceForensics dataset and provided a detection rate of 98% on deepfake videos. As GAN based deepfake generators utilized an upsampling layer to improve image resolution, some researchers analyzed traces left by this up-convolution operation using EM clustering (Guarnera et al., 2020) and frequency artefacts (Durall et al., 2020). Accuracy of 90.22% on self-created dataset and 90% on FF++ were obtained by these techniques. However, such visual disturbances are not supposed to be seen in the foreseeable future when manipulators become more mature.

Meanwhile, researchers explored frequency-domain features of facial region (Kohli and Gupta, 2021) which generated an accuracy of 85.24% and 66.50% on FF++ (Rössler et al., 1901) and Celeb-DF (Li et al., 1909) dataset. Another technique (Matern et al., 2019) utilized three different sets of facial features i.e., color difference in left and right eye, a combination of inconsistent eye and teeth details, and a combination of irregular face and nose border. AUC of 86.6% was obtained from the said technique. One of the early approaches (Yang et al., 2019b) also utilized facial keypoints to extract 3D-head pose of a person for deepfake detection and provided an AUC of 86.65%. Also, an approach based on optical flow analysis (Amerini et al., 2019) of suspected videos through a convolutional network provided an accuracy of 81.61% on FF++ dataset. Recently, a technique (Khalil et al., 2021) utilized texture artefacts of facial region and provided AUC of 76.8% on DFDC-P (Dolhansky et al., 1910) dataset and 77.1% on Celeb-DF dataset. Another technique (Akhtar and Dasgupta, 2019) utilized various handcrafted features and performed SVM classification of same.

2.2. Deep network based deepfake detection approaches

Researchers in deepfake detection also explored an area of deep networks and proposed different deep architectures (Wang et al., 2020; Agarwal et al., 2004; Mehra, 2020; Guo et al., 2005; Mittal et al., 2003; Khalid and Woo, 2020; Masi et al., 2008). On one side, there is a FakeSpotter (Wang et al., 2020) that keeps track of activation behavior of neurons of a Face Recognition system and reported an average accuracy of 90.6%. On the other side, ARENnet (Adaptive Residual Extraction Network) (Guo et al., 2005), which mainly focused on image residuals provided an average accuracy of 93.99%. One of the state-of-art approaches is based on human-specific behavior and appearance features (Agarwal et al., 2004). This approach identifies a video as deepfaked if identities extracted using both features are found to be different and obtained an accuracy of 94.14% on self-created dataset. One of the proposed approaches trained ResNet-18 models on spatial face regions which provided an accuracy of 96.75% on FF dataset. Afterwards, some researchers analyzed similarities between audio and visual features using siamese based network architecture with an AUC of 84.4%.

Along with supervised binary classification approaches, an unsupervised way of deepfake detection was considered in (Khalid and Woo, 2020), wherein deepfakes were classified as anomaly by a VAE (Variational AutoEncoder) based architecture. By utilizing only real videos for training a VAE, an accuracy of 88.28% was obtained. The technique proposed in (Arini et al., 2022) utilized trained Xception and ResNet50 based on Gaussian filtered and LBP-image, and reported an accuracy of 79% on Celeb-DF dataset while (Wang et al., 2021) proposed a new model by inserting LBP-layers in between the layers of ResNet and GramNet models.

Early works also utilized temporal nature of the video into consideration for deepfake detection. Training 3D-CNN based model (Nguyen et al., 2021) using 16 frames per sequence provided an accuracy of 99.33% and 99.7% on deepfake videos of FF++ (Rössler et al., 1901) and DF-TIMIT (Korshunov and Marcel, 1812) dataset respectively. Also, ResNet-50 (Caldelli et al., Del Bimbo) trained on optical flows of concerned video sequences provided an accuracy of 97.35% on deepfake videos of FF++ dataset. Another approach (Montserrat et al., 2004) utilized EfficientNet-b5 (Tan and LeEfficientnet, 1905) (initialized with ImageNet weights) and bi-directional GRU[5] for spatial and temporal analysis. An AFW (Automatic Face Weighting) layer is employed in between to emphasize the relevant features. End-to-end training of proposed model provided an accuracy of 92.61% on DFDC dataset. Other techniques that utilized a combination of convolution and recurrent architectures are Capsule Net with LSTM (Mehra, 2020) and color domain/LoG features with Bi-LSTM (Masi et al., 2008). Accuracies reported by these techniques are 83.42% and 93.18% on DFDC and FF++ datasets respectively.

2.3. Biological artefacts based deepfake detection

Other than visual artefacts and deep learning variants for deepfake detection, some researchers analyzed that computer-generated videos lack some attributes pertaining to human biological system. Considering the fact, some state-of-art approaches analyzed heart rate (Ciftci and DemirFakecatcher, 1901; Fernandes et al., 2019) of a person in suspected video which reported an accuracy of 77.33% for deepfake detection. Due to the lack of efficient techniques for analyzing biological artefacts from face sequence only, biological signals were not found much accurate for the task of deepfake detection. Also, the advancement of deepfake generation models is continuously reducing the gap between deepfaked videos and pristine ones by gaining knowledge of footprints utilized by deepfake detectors. Some researchers also worked in the direction of deepfake prevention (Hasan and Salah, 2019) and proposed PoA (Proof of Authenticity) system that is supposed to track transactions performed on digital data through blockchaining. These transactions would then be recorded on smart contracts.

Most of the state-of-art techniques proposed till now were evaluated on first and second generation datasets. Deepfake videos

[5] Gated Recurrent Unit.

contained in current generation datasets look more realistic and are hard to detect. Moreover, most of the approaches utilized a blind approach for deepfake detection without any focus on specific attributes. A deep learning based complex architecture is supposed to extract deepfake led artefacts automatically. Relatively simple models take large time to learn features while complex models may not be memory-efficient. Thereby, in this paper, a simple model is designed that is supposed to focus on some peculiar features only. Face swap performed through GAN based architecture usually cause blurriness or irregular texture on deepfaked face. Moreover, if GAN is trained for less time, certain features from source face and target face doesn't segregate effectively resulting in irregularity in texture pattern, and sometimes induce ghosting artefacts. Further, some GAN's are not much efficient to deepfake faces from different angles. Thereby, face movement at different angles can not be generated efficiently among subsequent frames in deepfake video that tends to generate blurriness and ghosting artefacts.

In this paper, a simple yet efficient model is trained for very few epochs for deepfake classification. Rather than training a model using raw faces, the proposed LBPNet model focuses on texture features that results in fast learning of deepfake features. These texture features are extracted through LBP which is discussed in Section 3.1. The proposed approach provides a fair trade-off between computational complexity and performance.

3. Methodology

3.1. Local Binary Pattern as texture descriptor

Local Binary Pattern, originated in 1996 (Ojala et al., 1996), became popular very early due to its relevance in various computer vision applications (Ojala et al., 2002). Along with simple computation, the most important property of LBP approach is its tolerance against illumination variations (Huang et al., 2011b). The basic nature of LBP is to analyze local texture pattern of 2D image. This local texture representation is computed by comparing each pixel in an image with its surrounding ones. This comparison between a pair of pixels results in a binary value 0 if respective pixel is of lower intensity than its neighboring one or binary value 1 otherwise. This sequence of binary digits constitutes a binary number. Then, the pixel is labelled with decimal equivalent of the resultant binary number. Procedure of computing LBP is demonstrated in Fig. 3. In the figure, an image block of size 3*3 is shown with an anchor pixel (center pixel) denoted by A. After thresholding each pixel by the anchor one, a binary sequence is computed. In LBP coded image, anchor pixel value is replaced by decimal equivalent of thresholded binary sequence. Note that each pixel would be considered anchor pixel repeatedly to obtain the resultant LBP coded image.

3.1.1. Methodology adopted to compute LBP of 2D image

1. Divide a 2D image into overlapping blocks of size 3*3.
2. Consider the center pixel of each block as Anchor pixel (A). Each anchor pixel would have 8 neighbors in 3*3 neighborhood.
3. Threshold all pixels of block by the corresponding anchor pixel as per following conditions:
   - 3.1 If intensity of any pixel is greater than anchor pixel Label the corresponding pixel as 1.
   - 3.2 If intensity of any pixel is smaller than anchor pixel Label the corresponding pixel as 0.
4. In thresholded block, starting from the top left pixel, concatenate all the binary digits by moving in clockwise direction. Constitute 8-bit binary number from these.
5. Convert the binary number to decimal one. To perform binary to decimal conversion, each bit in binary number is multiplied by power of two (with respect to bit position) and then summed.
6. Replace the intensity value of corresponding anchor pixel with the resultant decimal number.

For generalized representation, consider an image divided into overlapping blocks of size b*b with p_i as i^{th} pixel of a particular block. LBP of each block can be represented in mathematical form as follows.

$$LBP = \sum_{i=0}^{b-1} t(p_i - a)*2^i \tag{1}$$

Here, t is the threshold value of each neighbouring pixel computed by its comparison with center pixel, which can be written mathematically as:

$$t(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x >= 0 \end{cases} \tag{2}$$

3.2. Classification of deepfakes

The proposed automatic deepfake detection approach is split into three phases. First phase comprises preprocessing that is vital for any computer vision classification problem. Since the proposed technique is focused on facial manipulation detection, the foremost step is to detect and extract facial region from an image. Afterwards, feature extraction is performed in second phase through LBP procedure as discussed in Section 3.1. Third phase consists of CNN based architecture to perform automatic classification of deepfaked data from pristine one. The classifier is trained and tested with LBP coded face regions. An overview of proposed methodology along with description of phase-wise steps for deepfake classification is demonstrated in Fig. 4.

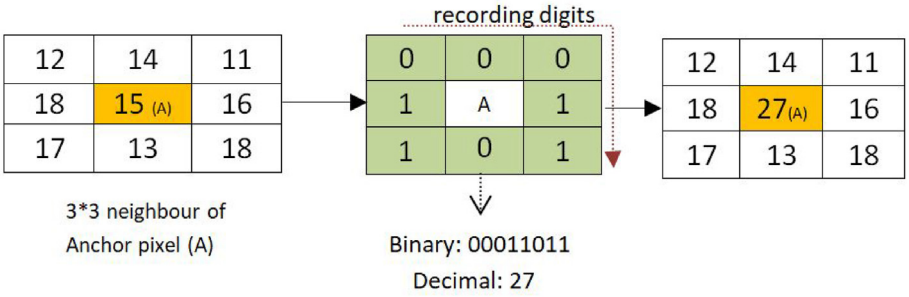


Fig. 3. LBP computation from 3*3 neighbourhood.

3.2.1. Preprocessing

Training data is first preprocessed before it is fed to deep learning algorithm to extract meaningful insights from it. Steps adopted for preprocessing are shown in Fig. 4. From each input video, a frame is first extracted. To focus on texture details of face region, only face area is extracted from all the frames. Face detection is performed using MTCNN (Xiang and Zhu, 2017) classifier. MTCNN detector outputs three keys: coordinates of facial boundary, confidence probability by which face is detected, and 5 facial keypoints as demonstrated in Fig. 5. Through bounding box coordinates, facial area is cropped from the whole frame, and resized to (256,256,3) for training. Due to deepfaking procedure, some of the images exhibit compression artefacts and blurriness around face boundaries that reduces the confidence value of face being detected. Thereby, threshold of 95% is chosen for the confidence value. However, the utilized MTCNN classifier is not able to detect face from the low contrast images. Also, it performs false face detections in case input image contains another face like structure in the background along with the subject (person). Owing to such false face detections, following preprocessing steps are employed in the proposed work:

1. **Gamma Correction:** Gamma correction (Rafael and Gonzalez, 2007) is utilized for face detection in low contrast images by performing contrast enhancement. However, it is applicable if input image is very dark, and its average intensity is less than threshold, which is set to 50 during the evaluation phase.
2. **Inter-frame face similarity:** To eliminate the chances of false face detections among the subsequent frames, 128-D facial embeddings of each face are extracted and compared. Facial embeddings are extracted using Facenet (Schroff et al., 2015) classifier pretrained for face recognition. After selecting the first face with high confidence value, faces from all other frames are chosen based on their similarity with previously selected face. Similarity matching is also used to keep track of same person in multi-person video so as to avoid any false face detection in between.

3.2.2. Convolutional network design

Increasing demand and impactful performance of CNN in computer vision encouraged to utilize the same for tampering detection too. For deepfake detection, different variants of CNN performed well (Kohli and Gupta, 2021; Shang et al., 2021; Khalil et al., 2021). Thereby, instead of subjective analysis of LBP images, a CNN based model is proposed that can automatically detect deepfaking through texture features. The model proposed in this paper is shown in Fig. 6. The proposed model exhibits 14 sets of Convolutional, Batch Normalization, and ReLU layer triplets followed by 2 fully connected layers. Input provided to each fully connected layer is optimized by a dropout of 0.4 to prevent overfitting of training data. At the end, sigmoid activation is applied for binary classification. The four prominent layers of any CNN model are described here:

Fig. 5. Output of MTCNN Face Detector (Bounding box in red color and 5 keypoints colored green). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

1. **Convolutional layer:** Considered as the core building block of convolutional network, this layer performs convolution of different filters (kernels) with an input to learn meaningful features. These filters are itself learnable, and are applied through the whole depth of input image to output a location-wise feature map. An image is first padded to make corner pixels of block as significant as other pixels.
2. **Batch Normalization:** Acting as a regularizer, BatchNorm layer (Ioffe and Szegedy, 2015) mitigates the effect of unstability in input distribution and initialization on the actual output. It also speeds up the training procedure by normalizing entire batch at a time.
3. **ReLU:** Rectified Linear Unit is a combination of non-linearity and rectification layers. Without effecting the receptive fields of convolutional layers, this layer maps all negative values in the feature map to 0.
4. **Pooling layer:** Employed after some convolution layers, the pooling layer is used to spatially reduce the convolved feature map. It eventually makes the model learn dominant features, and reduces the need of more computational power. The proposed model utilizes Max Pooling and Average Pooling. The former one outputs a maximum value of pixels from each local feature map of input while the later returns an average of the same.

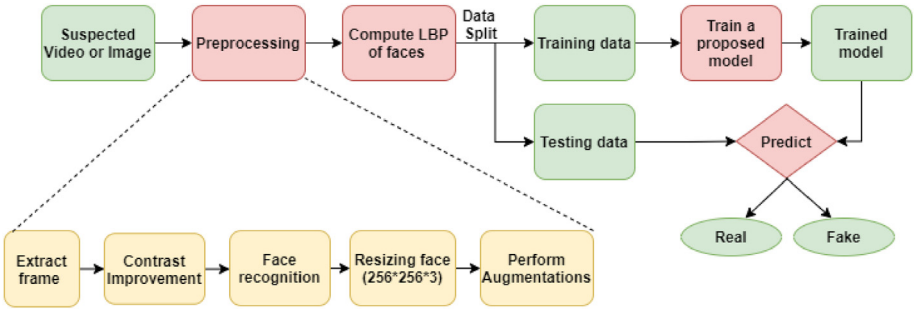


Fig. 4. Methodology of proposed LBPNet model for deepfake classification.

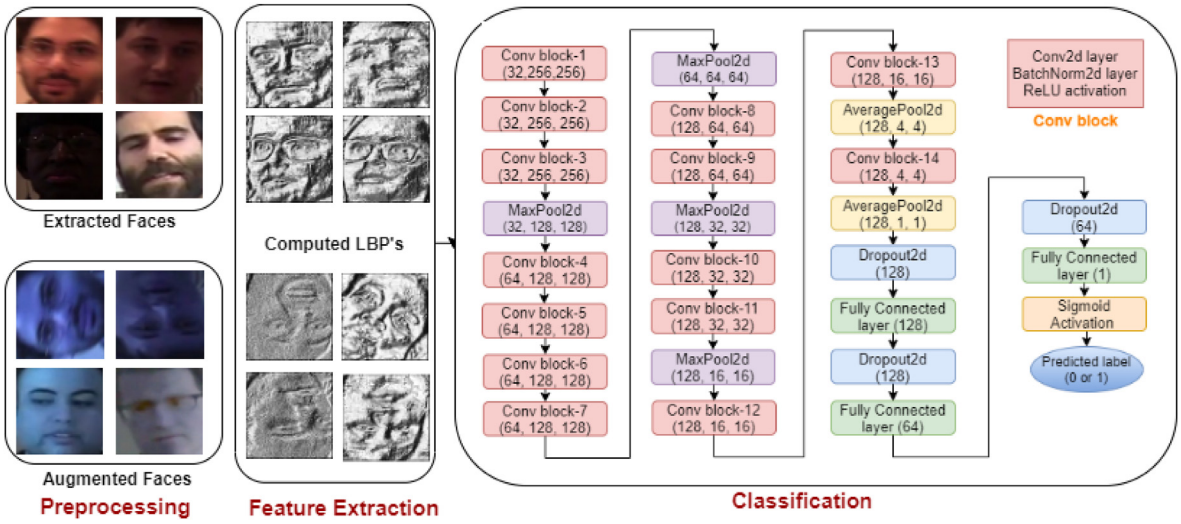


Fig. 6. Proposed Deep Learning based Architecture.

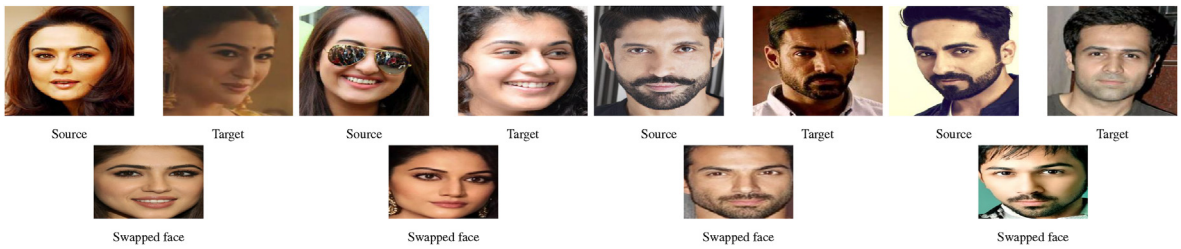


Fig. 7. Face manipulation of Indian celebrities done through FaceApp

5. **Fully Connected layer:** Based on feature extracted through all layers, a set of fully connected layers are utilized for classification of an input image. Thereby, output size of last fully connected layer must be equal to the number of classes predicted from the model.
6. **Sigmoid activation:** Sigmoid activation is utilized for mapping the probability outputted by last fully connected layer to a binary value for binary classification. If model is proposed for multi-class classification, softmax activation function is employed.

4. Data collection

Due to the continuous emergence of deepfake detection, numerous datasets have been proposed for the same. To curb the spread of deepfake menace, it is necessary to develop a technique to efficiently detect advanced deepfakes. In this paper, proposed LBPNet is evaluated on a wide variety of deepfake datasets with special emphasis on advanced deepfake datasets such as DFDC and Deeper-Forensics. Table 1 provides a thorough information about training and testing data utilized to evaluate the proposed model. The details of different datasets are described here:

1. **FaceForensics++ (FF++)** (Rössler et al., 1901)**:** This dataset is amongst the first generation deepfake datasets, wherein deepfaked videos exhibit visual artifacts and high texture irregularities. This dataset is comprised of 4 types of tampered contents namely, Deepfakes (DF), Faceswap (FS), Face2Face (F2F), and NeuralTexture (NT). Faceswap and Deepfake content exhibit face swapped videos of source person with target one, which is performed using computer graphic and deep learning respectively. Contrary to this, Face2Face and NeuralTextured set exhibit graphics-based and deep-learning based reenacted videos of target person with respect to the source person. Each set comprises of 1000 videos. In this paper, 700 videos from each set are used for training while remaining 300 videos are used for evaluation.
2. **Celeb-DF** (Li et al., 1909)**:** Gathered from real youtube videos of 59 different subjects, Celeb-DF comprises 590 real videos having diverse distribution of gender, age, lightning condition, and background. Since videos in Celeb-DF dataset exhibit less visual artefacts, it was claimed more realistic than other deepfake datasets. To balance two classes, 590 videos (10 videos of each subject) are chosen from 5639 deepfake videos.
3. **DeepFake Detection Challenge (DFDC)** (Dolhansky et al., 2006)**:** One of the challenging deepfake detection dataset, DFDC, was released through a competition organized by Facebook partnered with AWS and Microsoft. Deepfakes contained in this dataset were synthesized using 8 different mechanisms causing face/audio swapping. On videos recorded from 960 different subjects, 19 types of perturbation were employed. This dataset exhibits a total of 119245 labeled videos, arranged in 50 different parts. From 50 parts of DFDC dataset, first 40 are utilized for training while last 10 for evaluation. As each part contains videos of different subjects, train-test split between folders ensure generalizability of technique amongst different individuals. However, multi-person videos from this dataset are evaluated separately by tracking face of each individual separately. As it is not known which person's face is deepfaked, such videos are not considered during training.

Table 1
Training and testing data to evaluate proposed LBPNet (excluding augmentation data).

Dataset	Total Videos		Balancing required	Frames per video	Training Videos (#Frames)		Testing Videos	
	Real	Fake			Real	Fake	Real	Fake
DF	1000	1000	No	30	700 (21k)	700 (21k)	300	300
FS	1000	1000	No	30	700 (21k)	700 (21k)	300	300
F2F	1000	1000	No	30	700 (21k)	700 (21k)	300	300
NT	1000	1000	No	30	700 (21k)	700 (21k)	300	300
Celeb-DF	590	5639	Yes	30	485 (15k)	485 (15k)	105	105
DFDC	19154	100k	Yes	3	13594 (41k)	13594 (41k)	4005	4005
Deeper Forensics	1000	10000	Yes	20r/2f	700 (14k)	7000 (14k)	300 (6k)	3000 (6k)
FaceApp (Images)	1050	1115	No	–	726	789	324	326
FOM (Images)	7175	7175	Yes	1	5022	5023	2153	2152

4. **Deeper-Forensics** (Jiang et al., 2020)**:** Deeper-Forensics is the most recent publicly available dataset contributing in the field of deepfake detection. Here, 100 actors were paid to record videos with varied expressions, poses, illuminations and emotions. This dataset contains FaceForensics++ videos deepfaked using advanced techniques, and with new target identities. To maintain balance between real and fake faces, 2 faces are extracted from 10000 fake videos while keeping 20 faces from 1000 real ones.
5. **Self-generated dataset (FaceApp):** With increase in user-friendly deepfaking applications, it is necessary to design some deepfake detectors that can work on images generated by such applications. To evaluate the robustness of proposed technique on faces manipulated through real-life deepfaking applications, a new dataset is generated. Since early deepfake datasets doesn't consider Indian faces which itself exhibits high range of diversity, manipulation is done on Indian celebrity faces. This bollywood dataset is collected from kaggle.[6] For performing face swap among different celebrities, a freely available mobile application, FaceApp,[7] is utilized. A small dataset of 1000 real and 1000 fake images is generated for this paper. Samples of this dataset are provided in Fig. 7.
6. **Self-generated dataset (FOM):** Recently, first order model (FOM) (Siarohin et al., 2019) was proposed and become popular as a user-friendly tool for providing motion to images. To evaluate proposed model, we also generated another dataset using first order animation model. This model performs animations on image by mapping motions from some driving video given to it. To perform animations on images of bollywood dataset, some driving videos are recorded and applied randomly on each image. However, along with real set of celebrity images, one frame from each generated video is included in fake part of this dataset.

5. Experiments and evaluation

To prove the relevancy of texture features of facial region for deepfake detection, various experiments are performed on different datasets. Various image processing operations are performed using OpenCV (Howse, 2013) library of python. Albumentations (Buslaev et al., 2020) library is utilized to augment the training data. Also, the proposed model is designed, trained, and tested in Pytorch (Paszke et al., 1912). For training, NVIDIA P5000 GPU is utilized having 16 GB memory.

[6] https://www.kaggle.com/havingfun/100-bollywood-celebrity-faces.
[7] https://www.faceapp.com/.

5.1. Augmentations performed

To make the proposed model more robust, training data was made a bit noisy by adding different variations. Videos contained in the dataset have faces in erect position. In real-life cases, face could be found in an inverted position due to mirror view or rotated by some angle due to the camera position. Also, resaving and sharing of videos cause compression that can conceal the tampering artifacts. Thereby, JpegCompression, Vertical/Horizontal flip, and Random Rotate augmentations are performed randomly on each image of the training data. In some cases, real-life videos may also contain blur and noise due to camera focal length or motion. To prevent concealing of tampering artifacts from such images, Gaussian blur and Gaussian noise are employed on the training images. These augmentations are demonstrated in Fig. 8. To make an augmented dataset, one frame of each video of the train set is augmented. LBP's extracted from augmented images are described in Section 3.1.

5.2. Choice of hyperparameters

To train a proposed network, Adam optimizer is employed with default moment values of $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 10^{-8}$. The network is trained from scratch with batch size of 32, for 25 epochs. After 15 epochs, network starts overfitting on the training data. Since multiple frames of same subject and same video are utilized for all except DFDC dataset, network may overfit on certain subjects. Thereby, learning rate is initialized to 0.01 for DFDC, while 0.001 for the rest for gradual learning. However, the learning rate is decayed by a factor of 0.2 if validation loss does not decrease for five epochs.

5.3. Evaluation parameters

Once a model is built and trained, the most important task is to determine how valuable it is. Depending upon the nature of the model, different parameters are used for evaluation. This paper proposed a binary classification model which is supposed to predict whether a video is deepfaked or not. Prediction value should be 1 (positive) for deepfake video, while 0 (negative) for the pristine one. Deepfaked videos and pristine videos which are identified correctly by the model are counted in True Positives (TP) and True Negatives (TN) respectively. Sometimes, model may classify certain data samples and predict original video as deepfake while deepfaked video as original. This results in False Positives (FP) and False Negatives (FN). These parameters help to evaluate the proposed model based on different performance metrics provided in Table 2.

Fig. 8. Demonstrating different augmentations performed.

Table 2
Performance metrics used for evaluation (FPR: False Positive Rate, FNR: False Negative Rate).

Metric	Description	Equation
Accuracy	Measure of correctly predicted data instances with respect to total data instances.	(TP + TN)/(TP + TN + FP + FN)
Precision	Measure of correctly predicted positive data with respect to all positive predicted data.	TP/(TP + FP)
Recall	Measure of correctly predicted positive data with respect to all positive data.	TP/(TP + FN)
FPR	Measure of positive predicted negative data with respect to all negative data.	FP/(FP + TN)
FNR	Measure of negative predicted positive data with respect to all positive data.	FN/(FN + TP)
ROC curve	Degree of separability between two classes of particular model	–

5.4. Implementation and results

As already discussed, most of the deepfake generators tend to produce irregular texture around fake regions due to blending or mapping steps. Intra-frame texture analysis is performed to anatomize irregularities within a single image/frame. To train a proposed model, deepfake specified datasets are split into training and testing sets. As demonstrated in Fig. 9, proposed model has learnt efficiently from LBP coded faces of different datasets, and improved the performance on each epoch. Fig. 9 demonstrates the changes in accuracy and loss with each epoch on validation part of different datasets. Since DFDC and Celeb-DF are more diverse datasets than others, they exhibit efficient learning curve. On the other hand, FF++ dataset is easy to learn and results in immediate overfitting. During splitting, a subject level uniqueness is maintained to prevent overfitting of the trained model to certain subjects. Table 1 shows the train and test distribution of different datasets. Evaluation performance of different datasets, demonstrated in Table 3, revealed that post-prediction aggregation of multiple frames of a video greatly improves the overall performance. The procedure adopted for frame level and video level prediction is described here:

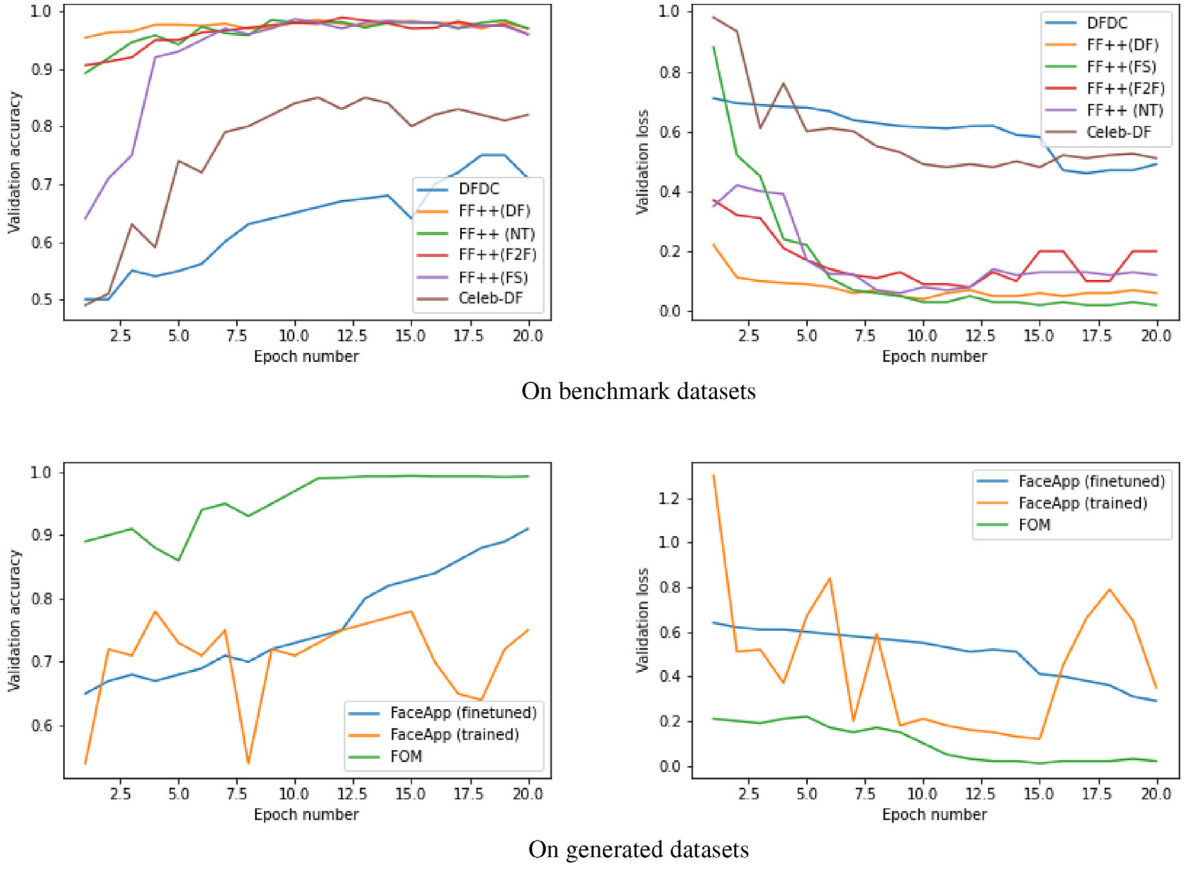


Fig. 9. Training performance of proposed LBPNet on each dataset. Validation performance per epoch is demonstrated using frame level prediction.

Table 3
Frame level and video level performance on different datasets.

Dataset	Testing level	Accuracy	Precision	Recall	FPR	FNR
DFDC	Frame-level	75.70	80.59	67.7	32.28	16.30
	Video-level	80.04	85.22	72.70	27.2	12.60
DFDC (multi-face)	Frame-level	71.3	64.6	94.6	52.2	5.38
	Video-level	74.6	67.6	95	46	4.9
Celeb-DF	Frame-level	87.56	83.4	91	16.36	9
	Video-level	92.38	91.2	93	8.18	7
Deeper- Forensics	Frame-level	83.9	79.5	100	42.6	0.0
	Video-level	86.4	82.1	100	36	0.0

1. **Frame-level prediction:** First experiment was performed by considering two frames from each video along with one augmented one for training. However, augmentation increased the performance of proposed model. Since DFDC dataset is completely imbalanced, fake videos are under-sampled with respect to the number of real videos. Prediction is performed by considering one random frame from each video.
2. **Video-level prediction:** While analyzing videos, it was observed that some frames exhibit more texture inconsistency and are much easy to be detected compared to others. Thereby, to improve the prediction result, post prediction aggregation is performed. In this, trained model predict results for ten frames of the video. Final result is obtained by averaging these ten predictions.

5.4.1. Evaluation on multi-face video sequences

Existing deepfake detectors did not pay attention to video sequences containing multiple persons. In this paper, multi-face videos are also evaluated separately with model pre-trained on DFDC dataset as this dataset exhibits such video sequences. Although these video sequences are labeled but no information is provided about which face is deepfaked in a particular video. To extract individual faces and maintaining consistency along a sequence, 128-D facial embeddings are extracted and compared as discussed in section 3.2.1. Evaluation is performed on each face of the video sequence separately. Video is considered deepfaked if both faces are predicted fake otherwise real. Performance evaluation on multi-face videos is reported in Table 3.

5.4.2. Evaluation on self-generated datasets

After evaluating the proposed LBPNet on different benchmark datasets, faces manipulated through user-friendly mobile applications are also tested. First experiment was performed by training network with training part of FaceApp dataset. Since the generated data is not sufficient for training, network did not perform very well. Thereby.LBPNet trained on benchmark datasets are fine-tuned for another 25 epochs with train-split of FaceApp data. Performance was found best with model trained on Celeb-DF dataset which provided an accuracy of more than 90%. Detailed results are reported in Table 4.

5.4.3. Robustness against different compression levels

In this social networking era, where any digital media can be easily shared among whole population, deepfake media can greatly harm someone's dignity. Transfer of digital media through networking indulge compression which can conceal tampering artefacts to a large extent. Thereby, it is necessary that the deepfake detector works efficiently on compressed media too. Considering the requirement, the proposed technique is evaluated on both high-quality and low-quality deepfaked videos. One of the deepfake specified datasets, FF++, contains deepfaked and pristine videos at different compression levels, i.e., raw (no compression), easy (compression factor: 23), and hard (compression factor: 40). Table 5 demonstrates that LBPNet provides good performance even on highly compressed deepfake videos with a detection accuracy of 93.8%. On raw videos, proposed model is able to efficiently detect any kind of tampering. NT deepfake is supposed to deepfake lip motion only and may not induce texture inconsistency. Still, LBPNet performs better for NT deepfake videos compared to state-of-art.

5.4.4. Robustness against different tampering type

Performance results provided in Tables 3 and 5 revealed that LBPNet is able to detect a wide variety of deepfake videos of different datasets. Good performance of LBPNet proves the presence of texture inconsistencies in deepfaked faces. Along with deepfake videos, FF++ dataset contain different types of tampered videos. Fig. 10 presents the AUC-ROC curve of the proposed model tested on different tampering types. Although best performance is achieved on DF videos of FF++ dataset, LBPNet is also able to detect tampering in FS, F2F, and NT videos of high and moderate quality with a detection accuracy of more than 95%. FS and F2F videos were generated using computer graphics technique where texture variability can not be analyzed to much extent. This texture variability further reduces in highly compressed videos that causes a slight reduction in performance for highly compressed FS and F2F videos. On the other hand, NT is a deep learning procedure employed to fake lip movements, analysis of which becomes difficult in highly compressed videos. In spite of this, LBPNet provided good performance even for low quality videos as compared to state-of-art.

5.4.5. Generalization ability of proposed model

To evaluate the generalization ability of LBPNet, extensive cross-database evaluations are conducted in two different settings. First experiment is performed to evaluate generalization from one compression factor to another while another experiment evaluates generalization from one tampering type to another.

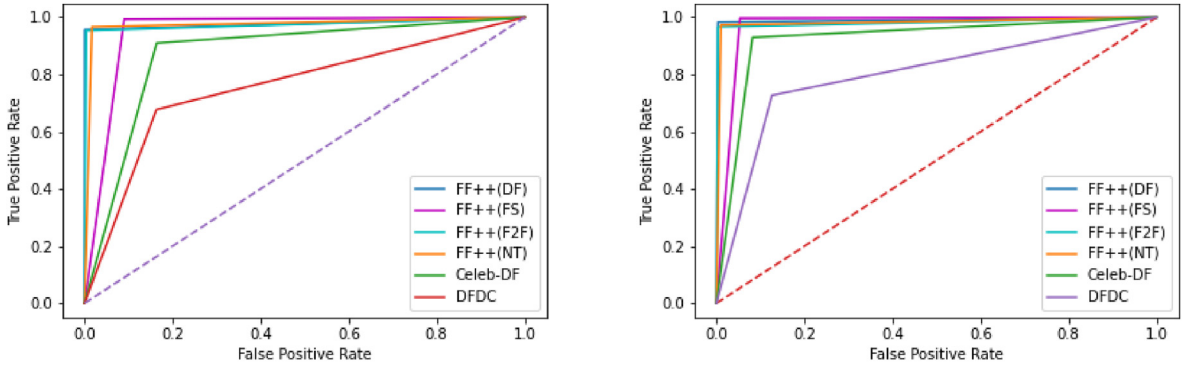
In Table 6, generalization ability of the proposed technique across varied quality videos is demonstrated. To perform this, training is done on similar quality videos of respective tampering type, and trained model is tested on videos compressed at different compression levels. It can be clearly seen that models trained on raw videos of all manipulation types doesn't generalize well on compressed videos of the same. Possible reason can be the inability of model to learn deeper artefacts from raw videos. Since real and deepfaked videos can be easily distinguished if compression is not there, model starts overfitting after few epochs and thereby, cannot learn minute details. Contrary to this, models trained on mildly-compressed and highly compressed videos learn sufficient details of tampering artefacts, and hence generalize on raw videos also.

Table 4
Performance on self generated datasets.

Dataset	Testing level	Accuracy	Precision	Recall	FPR	FNR
FaceApp (trained)	Frame-level	78.8	99.4	56.5	0.26	43.4
FaceApp (fine-tuned)	Frame-level	92.9	93.75	88.96	5.8	12.03
FOM-dataset	Frame-level	91.2	99.94	83.3	0.46	16.6

Table 5
Performance of proposed model on FF++ dataset at different resolutions (in %). Models are trained for different manipulation methods and compression levels separately.

Dataset	Prediction	Compression	Accuracy	Precision	Recall	FPR	FNR
FF++ (DF)	Frame-level	No	97.83	99.6	96	0.3	4
		Easy	95	95	95	5	5
		Hard	90.67	90.13	91.33	10	8.67
	Video-level	No	99.17	99.67	98.34	0.3	1.67
		Easy	97.1	97.3	97	2.67	3
		Hard	93.8	98.17	89.33	1.67	10.67
FF++ (FS)	Frame-level	No	95.17	91.41	99.33	9.33	0.67
		Easy	94.17	94.9	93.33	5	6.67
		Hard	85	88.88	80	10	20
	Video-level	No	97.17	94.92	99.67	5.33	0.33
		Easy	96.5	96.6	96.33	3.33	3.67
		Hard	89.83	95.09	84	4.33	16
FF++ (F2F)	Frame-level	No	97.5	99.65	95.33	0.33	4.67
		Easy	94	92.5	95.67	7.67	4.33
		Hard	85	83.9	86.67	16.67	13.33
	Video-level	No	98.17	99.66	96.67	0.33	3.33
		Easy	96.1	94.8	97.67	5.33	2.33
		Hard	88.5	89.15	87.67	10.67	12.33
FF++ (NT)	Frame-level	No	97.5	98.3	96.67	1.67	3.33
		Easy	86	86.5	85.33	13.33	14.67
		Hard	69.16	66.57	77	38.67	23
	Video-level	No	98.17	98.98	97.33	1.00	2.67
		Easy	87.67	94.14	80.33	5	19.67
		Hard	75.83	71.8	85	33.33	15



(a) ROC curve for frame level prediction

(b) ROC curve for video level prediction

Fig. 10. ROC curve for different types of tampering.

Table 6
Cross-compression generalization ability on FF++ dataset (Rows represent training dataset while columns represent testing data) (DF:DeepFake, FS:FaceSwap, F2F:Face2Face, NT:NeuralTexture

DF	C0	C23	C40
C0	99	53	54
C23	98	95	94
C40	87	85	93
FS	C0	C23	C40
C0	97	81	58
C23	95	96	80
C40	84	79	89
F2F	C0	C23	C40
C0	98	61	52
C23	96	96	72
C40	89	88	88
NT	C0	C23	C40
C0	98	56	57
C23	95	87	67
C40	72	71	75

Table 7
Cross-manipulation generalization ability on FF++ dataset.

C0	DF	FS	F2F	NT
DF	97.83	49.4	51.2	53.3
FS	49	95.17	56.2	45.2
F2F	77.2	49.2	97.5	57.3
NT	68.5	51.2	52.3	97.5

Eventually, proposed model achieved convincing results on different compression factors. Moreover, Table 7 demonstrates the performance of LBPNet on unseen manipulation types. The testing is performed by training the model on one manipulation type and testing the same on other three types. Since DF and FS videos contain visual artefacts that are easy to learn, model trained on such videos does not perform well on other datasets. Contrastingly, model trained on F2F and NT videos are more generalizable on DF manipulation.

Fig. 11. Explaining activations of LBPNet on real and deepfaked faces of DFDC dataset. (First and second row contains examples of faces and their LBP's (true negatives (1–3 columns) and true positives (4–6 columns)) while (third and fourth row contains examples of faces and their LBP's (false negatives (1–3 columns) and false positives (4–6 columns)).

5.4.6. Visualizing class activation maps of LBPNet

To understand and visualize the class activation performance of the proposed LBPNet, further investigation is performed in the context of activation maps. Here, Grad-CAM (Selvaraju et al., 2017) is used to highlight face content that is responsible for prediction. Random examples for true and false predictions are selected from DFDC and FF++ dataset. Fig. 11 demonstrates activation of DFDC dataset. From this figure, it can be seen that the network focuses on some peculiar features of face area for deepfake detection. It has made true predictions on diverse range of faces as shown in first row of Fig. 11. However, network is not able to accurately detect side faces. Sometimes, network may falsely predict a real face as fake in case of blurred images. In DFDC dataset, some fake videos exhibit voice swaps rather than face swaps, so fake face features learnt by network may not perform well on that. It results in false negative predictions in case face does not exhibit any deepfake artefacts.

Class Activation Maps for FF++ videos are presented in Fig. 12. In some videos of FF++, deepfake artefacts are itself visible. However, activation maps highlights the features utilized by the last layer of trained network to perform classification. Similar activations are found on faces from Celeb-DF dataset as demonstrated in Fig. 12. Contrary to this, Face2Face and NeuralTextured videos in FF++ dataset exhibit lip-sync manipulation only, and thereby, the network does not learn features from the whole face. Some examples are demonstrated in Fig. 13.

. For comparison, accuracies reported by state-of-art models are utilized.

5.4.7. Comparison with state-of-art

Although numerous techniques for deepfake detection have been proposed in the state-of-art, research in deepfake detection can never put to stop owing to generation of more realistic deepfakes. Keeping in mind the continuous evolvement of deepfake detectors, development of more efficient deepfake detectors is still the demand of research fraternity. Moreover, most of the existing deepfake detectors are not evaluated on current generation deepfake dataset that are the challenging ones. Comparison of proposed deepfake detector LBPNet against fCNN (Kohli and Gupta, 2021), PPPNet (Shang et al., 2021), and Optical flow (Amerini et al., 2019) based approaches is shown in Fig. 14. The results demonstrate the efficiency of proposed LBPNet approach over the others. Fig. 14(a) and (b) shows the comparison of proposed technique with others

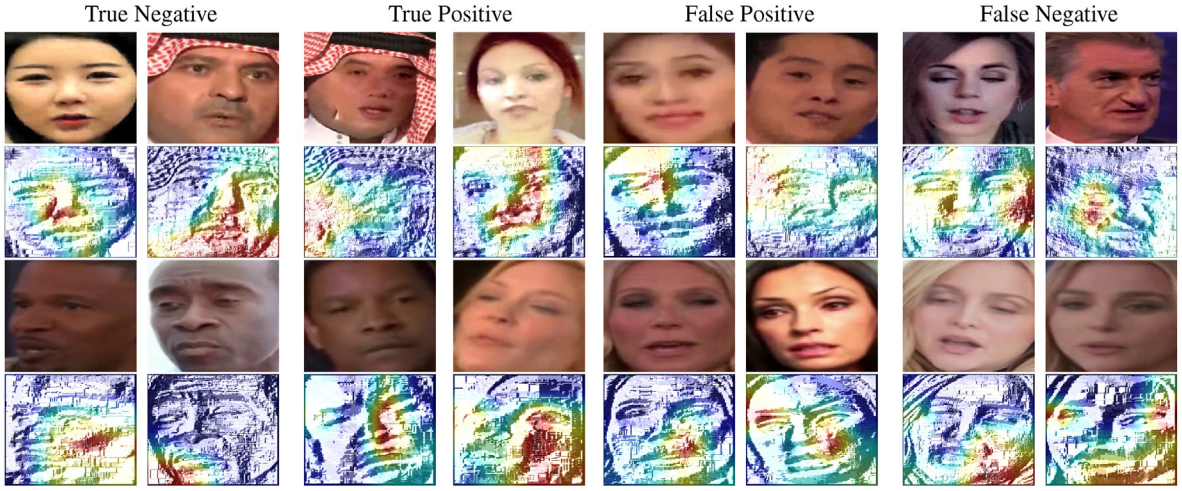


Fig. 12. Explaining activations of LBPNet on real and deepfaked faces and their respective LBP's from FF++ (1–2 rows) and Celeb-DF (3–4 rows) dataset.

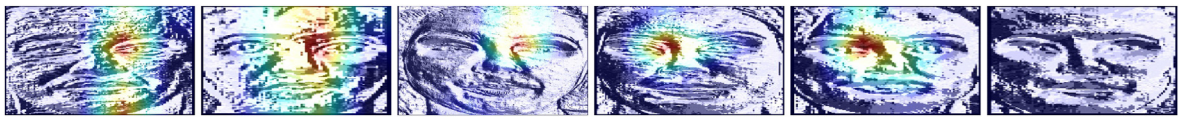
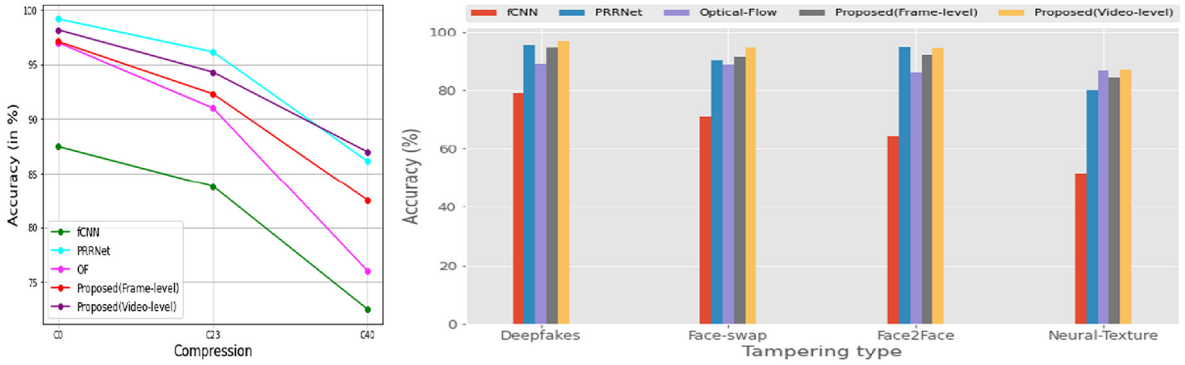


Fig. 13. Correctly predicted examples of Face2Face and NeuralTextured Videos from FF++ dataset.



(a) Different compression levels (b) Different tampering types

Fig. 14. Comparison of proposed model with state-of-art for FF++ dataset.

Table 8
Performance comparison of proposed model with iCap-Dfake (Khalil et al., 2021).

Approach	Dataset	Accuracy	Precision	Recall	FPR	FNR	Parameters
iCaps-Dfake (Khalil et al., 2021)	Celeb-DF	91.70	93.29	94.12	12.92	5.88	11,715,121
	DFDC-P	79.41	90.12	76.45	15.22	23.55	
Proposed	Celeb-DF	92.38	91.2	93	8.18	7	1,118,846
	DFDC-P	82.1	83.3	80.07	15.9	19.9	

with respect to different compression levels and varying tampering types respectively. Results reveal that LBPNet exhibits better performance for different tampering types and different compression levels. Although technique proposed in (Shang et al., 2021) reported slight better performance, but the proposed model is found better in terms of complexity and performs better for detecting NeuralTextured videos as well. Additionally, technique (Shang et al., 2021) is not evaluated on advanced deepfakes.

Further, comparison of LBPNet with texture based deepfake detection technique (Khalil et al., 2021) is provided in Table 8. As (Khalil et al., 2021) was evaluated on DFDC-P dataset, proposed LBPNet is also evaluated on the same dataset for fair comparison. Results prove the efficiency of proposed technique as compared to the existing deepfake detectors on advanced deepfakes such as DFDC and Deeper-Forensics. Moreover, using very few trainable parameters, LBPNet increased the performance of deepfake detection by 3% and 2% on diverse datasets DFDC and Celeb-DF in comparison to (Khalil et al., 2021).

6. Conclusion and future scope

Seeing the continuous advancement in deepfake generation technologies and their misuse in spread of fake news, development of efficient deepfake detectors has become the necessity of multimedia forensic system. Deepfaking procedures generally cause inconsistency in the texture pattern of mapped area that has been analyzed in this paper. Existing techniques have not considered texture based analysis for deepfake detection so far. To analyze texture pattern, proposed CNN-based model called LBPNet, is trained with LBP pattern of faces. Developed LBPNet model is evaluated on different benchmark datasets and provided a substantial accuracy of 99% on deepfake videos of FF++ dataset. One of the striking feature of the proposed technique is the accuracy of deepfake detection on advanced deepfake datasets such as Celeb-DF (92%) and DFDC (80%). Additionally, the paper reports the performance of deepafakes generated through user-friendly applications such as FaceApp and FOM. Results reveal the robustness and generalizability of LBPNet for compressed videos tampered through varied mechanisms, and from one compression level to another. In future, texture inconsistency analyzed from a single frame can be extended further by analyzing texture inconsistencies among sequence of frames.

Data availability

Data will be made available on request.

References

Afchar, D., Nozick, V., Yamagishi, J., Echizen, I., 2018. Mesonet: a compact facial video forgery detection network. In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, pp. 1–7.

S. Agarwal, T. El-Gaaly, H. Farid, S.-N. Lim, DEtecting Deep-Fake Videos from Appearance and Behavior, arXiv preprint arXiv:2004.14491.

Akhtar, Z., Dasgupta, D., 2019. A comparative evaluation of local feature descriptors for deepfakes detection. In: 2019 IEEE International Symposium on Technologies for Homeland Security (HST). IEEE, pp. 1–5.

Amerini, I., Galteri, L., Caldelli, R., Del Bimbo, A., 2019. Deepfake video detection through optical flow based cnn. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, 0–0.

Arini, A., Bahaweres, R.B., Al Haq, J., 2022. Quick classification of xception and resnet-50 models on deepfake video using local binary pattern. In: 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE). IEEE, pp. 254–259.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A., 2020. Albumentations: fast and flexible image augmentations. Information 11 (2), 125.
Caldelli, R., Galteri, L., Amerini, I., Del Bimbo, A., 2021. Optical flow based cnn for detection of unlearnt deepfake manipulations. Pattern Recogn. Lett. 146, 31–37.
Chan, C., Ginosar, S., Zhou, T., Efros, A.A., 2019. Everybody dance now. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5933–5942.
U. A. Ciftci, I. Demir, Fakecatcher: DEtection of Synthetic Portrait Videos Using Biological Signals, arXiv preprint arXiv:1901.02212.
Cole, S., 2017. Ai-assisted Fake Porn Is Here and We're All Fucked december. https://www.vice.com/en_us/article/gydydm/gal-gadot-fake-ai-porn.
B. Dolhansky, R. Howes, B. Pflaum, N. Baram, C. C. Ferrer, THe Deepfake Detection Challenge (Dfdc) Preview Dataset, arXiv preprint arXiv:1910.08854.
B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, C. C. Ferrer, THe Deepfake Detection Challenge Dataset, arXiv preprint arXiv:2006.07397.
Durall, R., Keuper, M., Keuper, J., 2020. Watch your up-convolution: cnn based generative deep neural networks are failing to reproduce spectral distributions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7890–7899.
Fernandes, S., Raj, S., Ortiz, E., Vintila, I., Salter, M., Urosevic, G., Jha, S., 2019. Predicting heart rate variations of deepfake videos using neural ode. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, 0–0.
Guarnera, L., Giudice, O., Battiato, S., 2020. Deepfake detection by analyzing convolutional traces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 666–667.
Z. Guo, G. Yang, J. Chen, X. Sun, FAke Face Detection via Adaptive Residuals Extraction Network, arXiv preprint arXiv:2005.04945.
Hasan, H.R., Salah, K., 2019. Combating deepfake videos using blockchain and smart contracts. IEEE Access 7, 41596–41606.
E. Hofesmann, THe state of deepfakes in 2020, Skynet Today.
Howse, J., 2013. OpenCV Computer Vision with python. Packt Publishing Ltd.
Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L., 2011a. Local binary patterns and its application to facial image analysis: a survey. IEEE Trans. Syst. Man. Cyber. Part C. (Applications and Reviews) 41 (6), 765–781.
Huang, D., Shan, C., Ardabilian, M., Wang, Y., Chen, L., 2011b. Local binary patterns and its application to facial image analysis: a survey. IEEE Trans. Syst. Man. Cyber. Part C. (Applications and Reviews) 41 (6), 765–781.
Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift arXiv:1502.03167.
Jiang, L., Li, R., Wu, W., Qian, C., Loy, C.C., 2020. Deeperforensics-1.0: a large-scale dataset for real-world face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2889–2898.
Jr, E.O., 2019. Thieves Used Audio Deepfake of a Ceo to Steal $243,000. https://www.vice.com/en_in/article/d3a7qa/thieves-used-audio-deep-fake-of-a-ceo-to-steal-dollar243000.
Khalid, H., Woo, S.S., 2020. Oc-fakedect: classifying deepfakes using one-class variational autoencoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 656–657.
Khalil, S.S., Youssef, S.M., Saleh, S.N., 2021. icaps-dfake: an integrated capsule-based model for deepfake image and video detection. Future Internet 13 (4), 93.
Kingra, S., Aggarwal, N., Kaur, N., 2022. Emergence of deepfakes and video tampering detection approaches: a survey. Multimed. Tool. Appl. 1–45.
Kohli, A., Gupta, A., 2021. Detecting Deepfake, Faceswap and Face2face Facial Forgeries Using Frequency Cnn. Multimedia Tools and Applications, pp. 1–18.
P. Korshunov, S. Marcel, DEepfakes: a New Threat to Face Recognition? Assessment and Detection, arXiv preprint arXiv:1812.08685.
Y. Li, M.-C. Chang, S. Lyu, In Ictu Oculi: Exposing Ai Generated Fake Face Videos by Detecting Eye Blinking, arXiv preprint arXiv:1806.02877.
Y. Li, X. Yang, P. Sun, H. Qi, S. Lyu, CEleb-Df: A New Dataset for Deepfake Forensics, arXiv preprint arXiv:1909.12962.
L. Li, J. Bao, H. Yang, D. Chen, F. Wen, FAceshifter: towards High Fidelity and Occlusion Aware Face Swapping, arXiv preprint arXiv:1912.13457.
Li, S.Z., Chu, R., Liao, S., Zhang, L., 2007. Illumination invariant face recognition using near-infrared images. IEEE Trans. Pattern Anal. Mach. Intell. 29 (4), 627–639.
I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, W. AbdAlmageed, TWo-Branch Recurrent Network for Isolating Deepfakes in Videos, arXiv preprint arXiv: 2008.03412.
Matern, F., Riess, C., Stamminger, M., 2019. Exploiting visual artifacts to expose deepfakes and face manipulations. In: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW). IEEE, pp. 83–92.
Mehra, A., 2020. Deepfake Detection Using Capsule Networks with Long Short-Term Memory Networks, Master's Thesis. University of Twente.
T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, D. Manocha, EMotions Don't Lie: A Deepfake Detection Method Using Audio-Visual Affective Cues, arXiv preprint arXiv:2003.06711.
D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horváth, E. Bartusiak, J. Yang, D. Güera, F. Zhu, et al., DEepfakes Detection with Automatic Face Weighting, arXiv preprint arXiv:2004.12027.
Nguyen, X.H., Tran, T.S., Nguyen, K.D., Truong, D.-T., et al., 2021. Learning spatio-temporal features to detect manipulated facial videos created by the deepfake techniques. Forensic Sci. Int.: Digit. Invest. 36, 301108.
Nguyen, T.T., Nguyen, Q.V.H., Nguyen, D.T., Nguyen, D.T., Huynh-The, T., Nahavandi, S., Nguyen, T.T., Pham, Q.-V., Nguyen, C.M., 2022. Deep Learning for Deepfakes Creation and Detection: A Survey, Computer Vision and Image Understanding, 103525.
Nirkin, Y., Keller, Y., Hassner, T., 2019a. Fsgan: subject agnostic face swapping and reenactment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7184–7193.
Nirkin, Y., Keller, Y., Hassner, T., 2019b. FSGAN: subject agnostic face swapping and reenactment. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7184–7193.
Ojala, T., Pietikäinen, M., Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. Pattern Recogn. 29 (1), 51–59.
Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987.
A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an Imperative Style, High-Performance Deep Learning Library, arXiv preprint arXiv:1912.01703.
Posters, B., 2018. Bill Posters on Instagram. Artificially Generated Video of Mark Zuckerberg. https://twitter.com/PressSec/status/1060374680991883265.
Rafael, R.E.W., Gonzalez, C., 2007. Digital Image Processing, third ed. Pearson.
Rahim, Z., 2020. 'deepfake' queen delivers alternative christmas speech. In: Warning about Misinformation december. https://edition.cnn.com/2020/12/25/uk/deepfake-queen-speech-christmas-intl-gbr/index.html.
A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Nießner, FAceforensics: A Large-Scale Video Dataset for Forgery Detection in Human Faces, arXiv preprint arXiv:1803.09179.
A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, M. Nießner, Faceforensics++: Learning to Detect Manipulated Facial Images, arXiv preprint arXiv:1901.08971.
Schroff, F., Kalenichenko, D., Philbin, J., Facenet, 2015. A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823.
Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626.
Shang, Z., Xie, H., Zha, Z., Yu, L., Li, Y., Zhang, Y., 2021. Prrnet: pixel-region relation network for face forgery detection. Pattern Recogn., 107950
Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N., 2019. First order motion model for image animation. Adv. Neural Inf. Process. Syst. 32, 7137–7147.
K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, J. Wang, HIgh-Resolution Representations for Labeling Pixels and Regions, arXiv preprint arXiv:1904.04514.
Suwajanakorn, S., Seitz, S.M., Kemelmacher-Shlizerman, I., 2017. Synthesizing obama: learning lip sync from audio. ACM Trans. Graph. 36 (4), 1–13.
M. Tan, Q. V. Le, Efficientnet: REthinking Model Scaling for Convolutional Neural Networks, arXiv preprint arXiv:1905.11946.
Tulyakov, S., Liu, M.-Y., Yang, X., Kautz, J., 2018. Mocogan: decomposing motion and content for video generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1526–1535.
Vincent, J., 2018. Jordan Peele Use Ai to Make Barack Obama Deliver a Psa about Fake News. https://www.theverge.com/tldr/2018/4/17/17247334/aifakenewsvideobarackobamajordanpeelebuzzfeed.
Wang, R., Juefei-Xu, F., Ma, L., Xie, X., Huang, Y., Wang, J., Liu, Y., 2020. Fakespotter: a simple yet robust baseline for spotting ai-synthesized fake faces. In: International Joint Conference on Artificial Intelligence. IJCAI).
Wang, Y., Zarghami, V., Cui, S., 2021. Fake face detection using local binary pattern and ensemble modeling. In: 2021 IEEE International Conference on Image Processing (ICIP). IEEE, pp. 3917–3921.
Xiang, J., Zhu, G., 2017. Joint face detection and facial expression recognition with mtcnn. In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE). IEEE, pp. 424–427.
Xiao, B., Wang, K., Bi, X., Li, W., 2018. J. Han, 2d-lbp: an enhanced local binary feature for texture image classification. IEEE Trans. Circ. Syst. Video Technol. 29 (9), 2796–2808.
Yan, S., He, S., Lei, X., Ye, G., Xie, Z., 2018. Video face swap based on autoencoder generation network. In: 2018 International Conference on Audio, Language and Image Processing (ICALIP). IEEE, pp. 103–108.
Yang, X., Li, Y., Lyu, S., 2019a. Exposing deep fakes using inconsistent head poses. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 8261–8265.
Yang, X., Li, Y., Lyu, S., 2019b. Exposing deep fakes using inconsistent head poses. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 8261–8265.
Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H., 2005. Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 1. IEEE, pp. 786–791.