PLOS ONE

RESEARCH ARTICLE

Using cascade CNN-LSTM-FCNs to identify AI-altered video based on eye state sequence

Muhammad Salihin Saealal[1,2], Mohd Zamri Ibrahim[1]*, David. J. Mulvaney[3], Mohd Ibrahim Shapiai[4], Norasyikin Fadilah[1]

1 Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, Pekan Campus, Pekan, Pahang, Malaysia, **2** Electrical Engineering Technology Department, Faculty of Electric and Electronic Engineering Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia, **3** School of Electronic, Electrical and Systems Engineering, Loughborough University, Loughborough, United Kingdom, **4** Centre for Artificial Intelligence and Robotics, Malaysia-Japan International Institue of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

* zamri@ump.edu.my

Citation: Saealal MS, Ibrahim MZ, Mulvaney D.J, Shapiai MI, Fadilah N (2022) Using cascade CNN-LSTM-FCNs to identify AI-altered video based on eye state sequence. PLoS ONE 17(12): e0278989. https://doi.org/10.1371/journal.pone.0278989

Editor: Sathishkumar V E, Hanyang University, KOREA, REPUBLIC OF

Received: June 29, 2022

Accepted: November 28, 2022

Published: December 15, 2022

Data Availability Statement: FaceForensics++ public dataset can be downloaded from Github (https://github.com/ondyari/FaceForensics/blob/master/dataset/README.md). All relevant data sample and programming codes related to submitted manuscript can be accessed from the Zenodo repository (https://doi.org/10.5281/zenodo.7137271).

Funding: This research is financially supported by the Fundamental Research Grant Scheme (FRGS/1/2021/ICT07/UMP/02/1) with the RDU number RDU210136 which is awarded by the Ministry of

Abstract

Deep learning is notably successful in data analysis, computer vision, and human control. Nevertheless, this approach has inevitably allowed the development of DeepFake video sequences and images that could be altered so that the changes are not easily or explicitly detectable. Such alterations have been recently used to spread false news or disinformation. This study aims to identify *Deepfaked* videos and images and alert viewers to the possible falsity of the information. The current work presented a novel means of revealing fake face videos by cascading the convolution network with recurrent neural networks and fully connected network (FCN) models. The system detection approach utilizes the eye-blinking state in temporal video frames. Notwithstanding, it is deemed challenging to precisely depict (i) artificiality in fake videos and (ii) spatial information within the individual frame through this physiological signal. Spatial features were extracted using the VGG16 network and trained with the ImageNet dataset. The temporal features were then extracted in every 20 sequences through the LSTM network. On another note, the pre-processed eye-blinking state served as a probability to generate a novel BPD dataset. This newly-acquired dataset was fed to three models for training purposes with each entailing four, three, and six hidden layers, respectively. Every model constitutes a unique architecture and specific dropout value. Resultantly, the model optimally and accurately identified tampered videos within the dataset. The study model was assessed using the current BPD dataset based on one of the most complex datasets (FaceForensic++) with 90.8% accuracy. Such precision was successfully maintained in datasets that were not used in the training process. The training process was also accelerated by lowering the computation prerequisites.

Introduction

Advancements in camera technology and the prevalence of social networks (Facebook, WhatsApp, and Instagram) and video-sharing sites (YouTube and Vimeo) have rendered digital

Higher Education (MOHE) via the Research and Innovation Department, Universiti Malaysia Pahang (UMP) Malaysia. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

media production, editing, and distribution easier and more popular. Processing requires each video frame to be altered individually, thus prolonging the fake video production process and time without sophisticated editing equipment and software. Consequently, realistic fake videos were rare as they are typically identified with the presence of explicit visual aberrations.

This situation has radically changed with the recent emergence of generative deep neural networks. The generative adversary networks (GANs) application [1, 2] has catalyzed the creation of software that limited manual intervention and produced videos from substantial photograph collections. The resulting fake media proved far more realistic when evaluated by a human viewer. As the first widely-available software implementing GANs, DeepFake was digitally published in early 2018. DeepFake, which employs GANs to replace an individual's face in a video sequence with synthesized faces of another counterpart, witnessed a significant rise in the number of fake online videos involving a breach of privacy and identification and legal repercussions [1–7]. The necessity to detect such false videos has led forensic science community members to develop novel technology.

Conventional media forensic techniques have utilized signal level cues (double JPEG compression), physical level information, or semantic level consistencies (meta-data consistency). Nevertheless, such approaches were not sufficiently reliable or efficient in identifying more generic DeepFake videos [8–12]. Traditional contrast enhancement (CE) anti-forensic methods have depicted their practical forging ability in erasing the forensic fingerprints of enhanced images within histograms and the grey-level co-occurrence matrix (GLCM) with color filter array (CFA) interpolation using signal-level cues [13]. Regardless, the pixel value changes resulting from this approach are frequently exposed in the pixel domain. Latent GANs could be alternatively applied to mitigate this issue. The method outperformed deep-learning-based CE detection techniques in the pixel, histogram, and GLCM domains under anti-forensic attacks. Summarily, fake video detection with CFA is no longer deemed reliable [14, 15].

Adaptive PRNU denoising (APD) counter-forensic attacks on digital images did not previously affect image textural properties [16, 17]. Hence, Venkata et al. proposed an image-texture layer-based forensic solution for the source identification of APD counter-forensic images, which reported successful counter-forensic image source attribution with over 96% accuracy [18]. A calibration loss function could be applied to alleviate the variance gap in the high-frequency sub-bands between generated images and their calibrated versions to evade forensic detection [19, 20]. Following Jianyuan Wu et al., this method outperformed current advanced JPEG anti-forensic counterparts. In other words, fake image detection using double JPEG compression no longer proves suitable [21, 22].

Much emphasis has been placed on deep learning-based approaches and DeepFake countermeasures in addition to reviewing traditional media forensics methods. The attacker assumably modifies the metadata to render it useless as it would provide otherwise valuable information to verify image and video authenticity [14, 23]. Meanwhile, metadata are frequently omitted when media assets are uploaded to a social network. Thus, it is no longer deemed appropriate to rely on metadata consistency for authentication purposes [8, 24]. The current state of forensic approaches in terms of DeepFake video identification implies the urgent need for novel detection techniques. This research introduced a novel means of revealing DeepFake videos through the eye-blinking pattern of the synthesized face.

Blinking involves rapidly closing and opening one's eyelid. The pre-motor brain stem controls these spontaneous blinks, which occur without conscious effort and serves as an essential biological function to (i) moisturize the cornea and conjunctiva with tears and (ii) remove irritants from the surface [25]. Generally, the interval between each blink is approximately two to 10 seconds for a healthy adult human with the actual rate varying based on the individual [26–28]. The duration of a typical blink cycle is between 0.1 and 0.4 seconds [11]. As such,

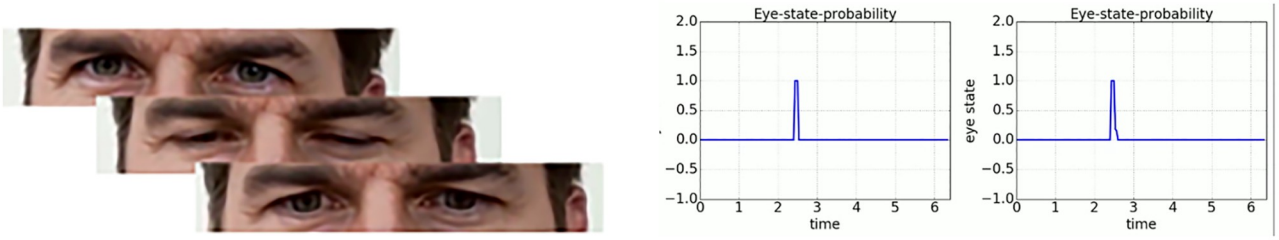

Fig 1. Explicit eye-blinking in real video sequences.

https://doi.org/10.1371/journal.pone.0278989.g001

spontaneous eye-blinking occurs within this specified frequency range and duration in videos of human. Contrarily, the core GAN model of DeepFake is trained with a large number of isolated human face pictures in many DeepFake videos. With an exposure time of 1/60 seconds, the probability of capturing a still image with the subject blinking is approximately 15% as illustrated in Fig 1. Based on the graphs, each action performed by the right and left eye proves useful to document any anomaly occurring throughout the target video. In reality, most of an individual's online pictures would not depict closed eyes as such images were probably not selected for publication. The absence of eye-blinking is a useful characteristic to ascertain DeepFake media.

The current study proposed a method that potentially leverages the benefits of using a deep learning model to train blinking pattern features. A fully connected network (FCN) was utilized in this approach with processed datasets running into a trained long short-term memory (LSTM) network. Appropriate video pre-processing techniques were incorporated to reduce the FCN computation time.

The three study contributions are presented as follows:

1. The current work demonstrates how spatial and temporal information could be derived from the input eye sequence and the eye blinking probability could be computed by cascading convolutional neural networks (CNNs) with LSTM. The data, which are converted into meaningful knowledge by pre-processing the information into a labeled sequence, could be more easily employed by other researchers.
2. The eye-blinking pattern is utilized as a feature to detect anomalies in fabricated videos. The novel dataset facilitates the training process for classification purposes by minimizing the computation prerequisites and the memory required by GPU.
3. An FCN with three distinct models is applied to the classification stage. Each model is subjected to three separate tests entailing a range of epoch counts, patience values, and batch sizes. The optimal model and associated attributes are duly obtained.

Related works

DeepFake videos generators

Artificial images and videos are conventionally generated with detailed 3D computer graphics models. Goodfellow et al. [1] first proposed the use of GANs, which encompassed a network generator and discriminator. The generator employed a set of training images from which output candidate images were extracted for subsequent analysis by the discriminator. Both networks underwent training with the creator striving to generate images that could deceive the discriminating unit in its attempts to distinguish synthetic images from actual training ones.

Several articles that described picture or face synthesis methods through GANs have been published. Denton et al. [29] recommended a Laplacian pyramid GAN [30] for coarse-to-fine picture generation, whereas Radford et al. [31] who suggested deep convolutional GANs (DC-GAN) demonstrated the potential of such an approach for unsupervised learning. Meanwhile, Arjovsky et al. [32] utilized the gaps in Wasserstein distances to stabilize training. Isola et al. [5] employed conditional adverse networks to learn mappings from image to image and train the loss function while Shrivastava et al. [33] utilized an integration of adversarial loss and self-regularization loss to close a distance measure between artificial and real picture distributions. Liu et al. [2] proposed a coupled GAN-based unsupervised image-to-image conversion process to examine mutual image representations in several domains. Notably, this algorithm is the basis for that of DeepFake. The original face would be located while the landmarks for the whole face region would be defined to facilitate subsequent operations, such as accurate face-cropping to precisely warp the target face into the original. The face swap was subsequently applied to the original frame, which led to the creation of DeepFake.

Detection of blinking eyes

The identification of eye blinks was previously examined in machine vision under fatigue detection applications [34, 35]. Pan et al. [25] developed an undirected conditional random field system to detect eye blinking by inferring eye closeness. The model simplifies the complex inference and optimizes the performance by omitting dependent eye state variables in a linear chain. Sukno et al. [36] employed active shape models with invariant optimal features (IOF-ASM) to delineate the eye contour and compute the vertical eye gap for eye condition assessment. A statistical analysis of the resulting shape sequence enabled the estimation of several blinking parameters. The outcome validation against manual annotations yielded a high level of accuracy in blink frequency estimation.

Yang et al. [27] incorporated a pair of parameterized parabolic curves to model the human eye shape and subsequently fitted a model to each frame in order to track eyelid movement. The face tracker, which was based on active shape models (ASMs) [37], employed a Kanade-Lucas-Tomasi tracker to continually track face landmarks. Determined by the face tracker, the eye region was refined by a deformable contour template for eyelid-fitting. A scalar quantity was proposed by Soukupova et al. [38], in which a rectangular bounding box was placed around the eye with an aspect ratio paralleling the degree of eye openness. The elicited pattern relied on the speed of eye-closing and opening, degree of eye-squeezing, and blink duration. In this vein, an 'eye aspect ratio support vector machine' was developed to identify the final eye condition with these features as inputs.

Kim et al. [6], who studied CNN-based classifiers to assess if one's eyes are open or closed, employed a ResNet-50 revised model [39] with a specified number of nodes in its fully connected layer. Li et al. subsequently extended the CNN-based classifier to consider the temporal relationship between consecutive frames as eye blinking occurs over several frames. Long-term recurrent convolutional networks proved suitable to analyze changes over several frames while mitigating the significance of changes introduced between consecutive images [40]. The authors then extended the method with LSTM-RNN to better control when and how to forget previous and update the currently hidden states.

Methodology

Fig 2 below depicts the DeepFake detection approach, whose process can be divided into the following stages:

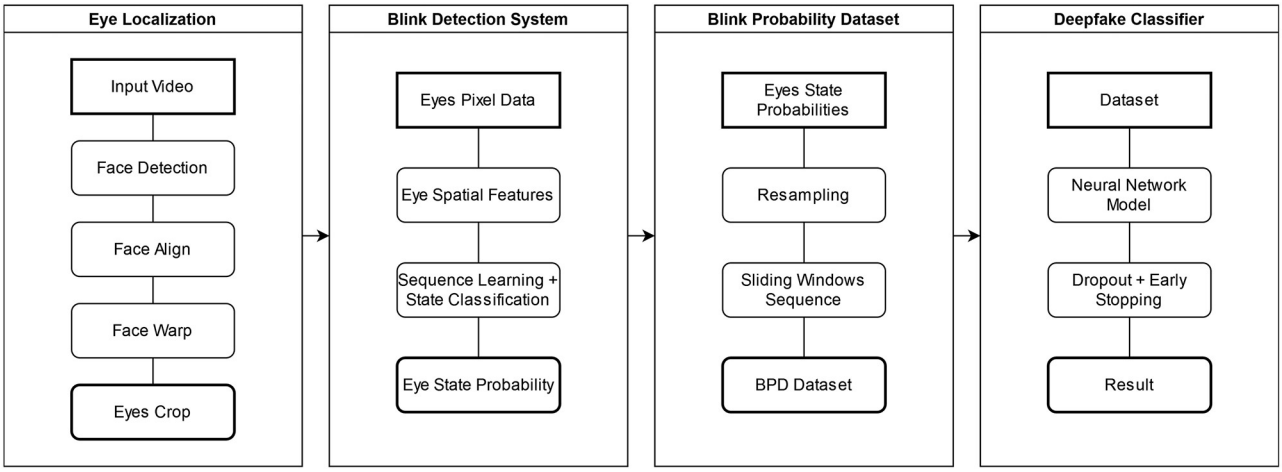

Fig 2. DeepFake detection approach based on eye state data.

https://doi.org/10.1371/journal.pone.0278989.g002

- Stage 1: Eye Localization. The video is extracted into frames during the pre-processing period with each face in each frame being characterized. The recognized faces are subsequently aligned and warped to ensure the consistency of their eye orientation and direction. The eyes are then cropped and saved as pixel values.
- Stage 2: Blink Detection System. The characteristics-extracting method allows the status of both eyes to be recognized. Every frame series is fed into the cascade model, which entails convolution, LSTM, and an FCN. Essentially, the output represents the probability of each eye condition.
- Stage 3: Blink Probability Dataset (BPD): The probability data collected in Stage 2 are resampled and processed within 4.5-second sliding windows to develop a novel BPD dataset.
- Stage 4: DeepFake Classifier. This classifier functions to train real and fake videos with data from the BPD dataset on eye state probability. The FCN architecture model is selected for training together with its dropout, early-stopping, batch size, and network configuration.

The visual studio code-`Python` integration served to run the algorithm. The `OpenCV` library was incorporated for frame extraction from the videos and image manipulation. Meanwhile, `dlib` library was applied to recognize the face mark, which allows bending, warping, and cropping of the eye. `Tensorflow` functioned as a framework to run the algorithm during the eye-blink state extraction based on a pre-trained model. This extraction was utilized to develop the blink pattern from the BPD dataset. A further model was structured following this new dataset to predict DeepFake videos with `PyTorch` and `PyTorchLightning`. Both `Tensorboard` and `matplotlib` were employed for outcome virtualization.

Eye localization

First, the datasets provided by Zhou T, Wang W, Liang Z, et al., the `FaceForensics++` dataset [41] and YouTube videos were extracted into individual frames. The face is detected through the `dlib` library while the landmark of the face is duly determined. The function of the library and FL detectors depend on the approach represented by [42], which outputs an array of 68 points in the (x, y) coordinate format. Essentially, the faces are aligned and warped

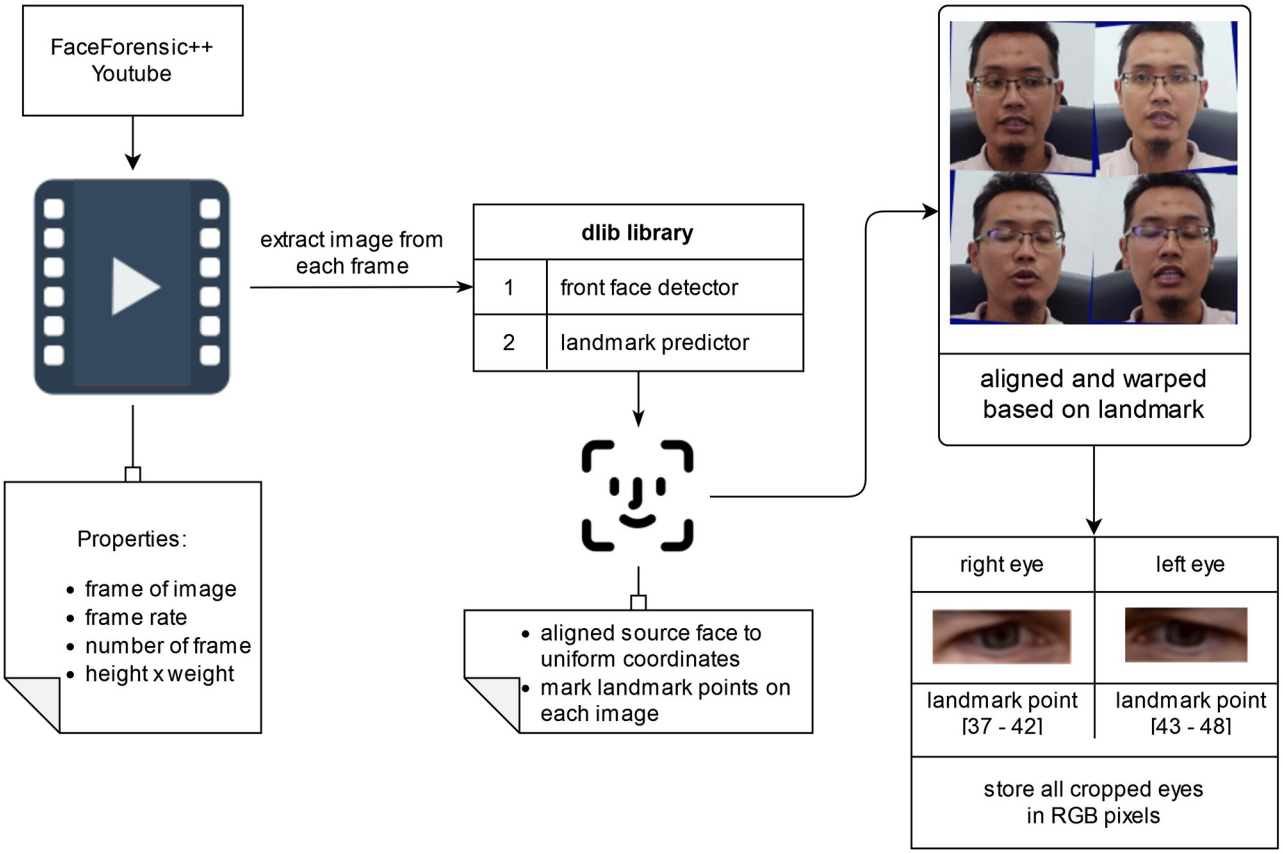

Fig 3. Obtaining the eyes from the input videos.

https://doi.org/10.1371/journal.pone.0278989.g003

before the eyes are cropped into a single input to ensure that the line joining the eye centers is horizontal and scaled to a uniform size. The eye positions are determined by points 37 to 48. Furthermore, the rectangular region is established by omitting the bounding boxes of each eye landmark points, thus scaling each bounding box by a factor of 1.25 in the horizontal direction and a factor of 1.75 in the vertical direction and ensuring the inclusion of the eye region in the cropped zone. Fig 3 illustrates the relevant processes. The depicted individuals provided informed consent in written form based on the PLOS consent form to publish their image alongside the manuscript.

Blink detection system

Fig 4 illustrates the blink detection system performance procedures. The sequences encompassing the cropped eye regions, which were derived from the eye localization in Stage 1, were saved as RGB images. These data sequences were fed into a pre-trained CNN to extract the spatial information for each eye. Additionally, the extracted features were fed into the LSTM network for further feature extraction.

The LSTM nodes illustrated in Fig 5 denote memory units that regulate when and how (i) previous hidden states are forgotten and (ii) hidden states are updated [4]. The first sigmoid function from the left of the block diagram, which is known as the forget gate, f_t, pushes the input x_t into the [0, 1] range.

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

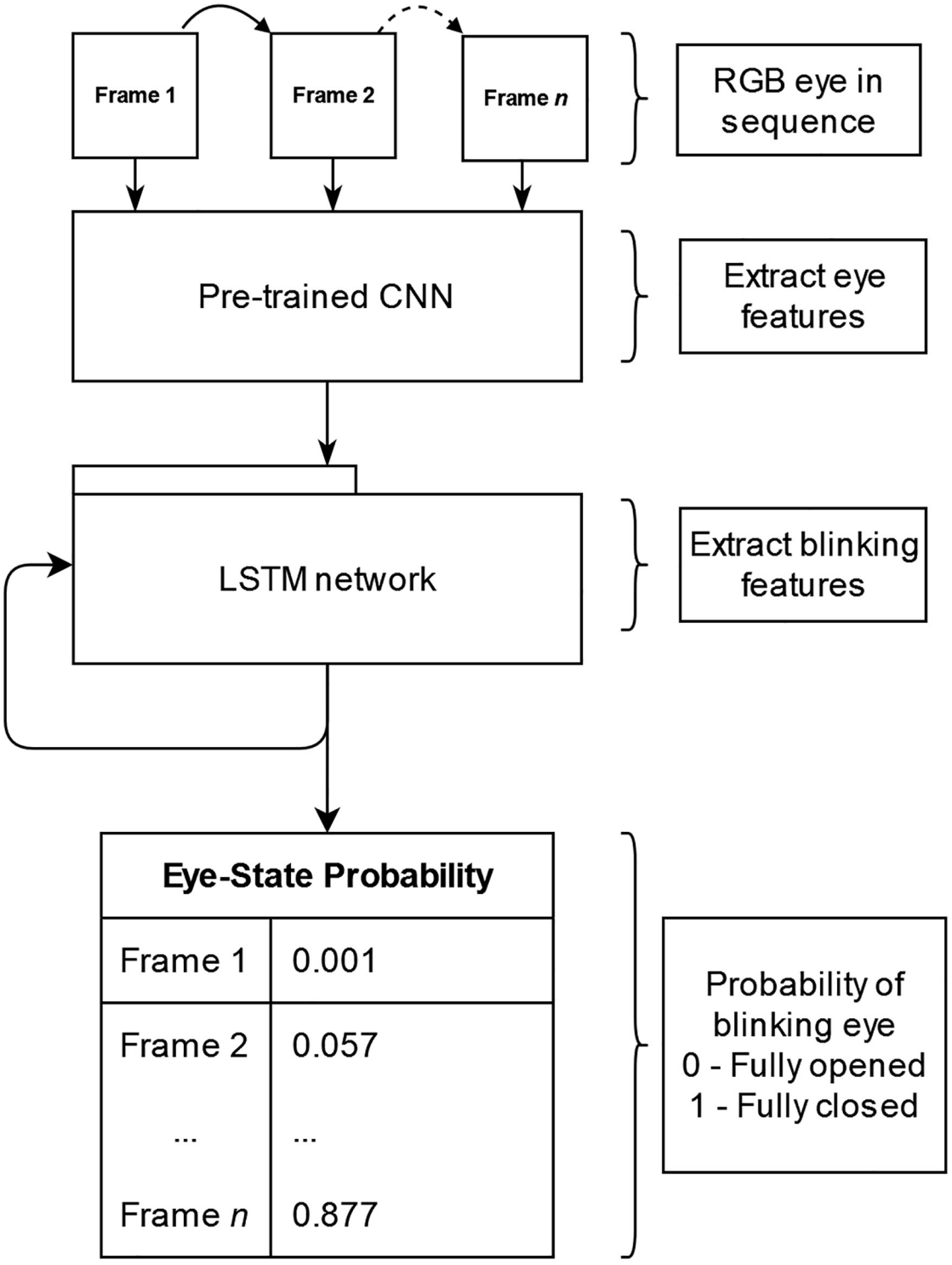

Fig 4. Blink detection system.

https://doi.org/10.1371/journal.pone.0278989.g004

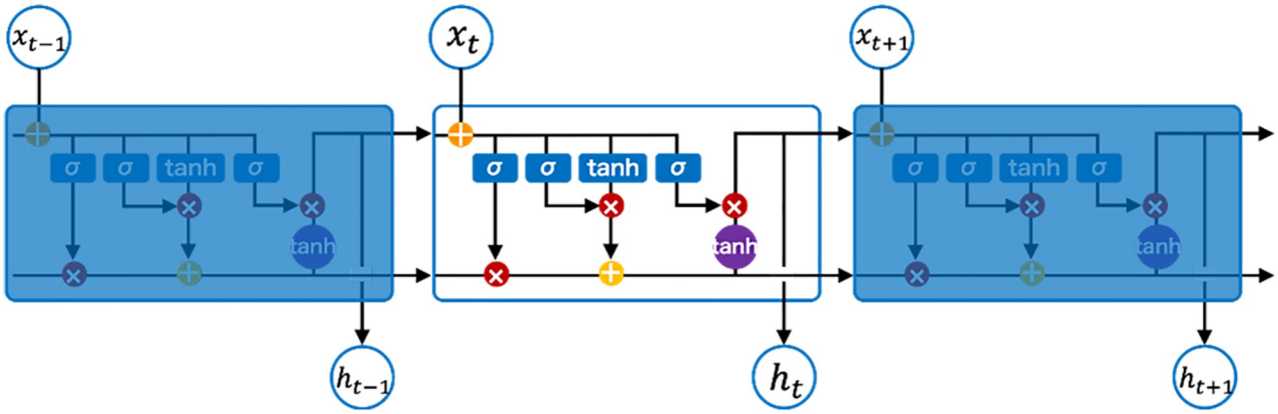

Fig 5. The LSTM.

https://doi.org/10.1371/journal.pone.0278989.g005

where σ denotes sigmoid function, w_x implies the weight for respective gate(x) neurons, h_t reflects the LSTM block output, and b_x indicates the biases for the respective gates(x).

The vector output of the forget gate was simply formed from a dot product of the input weights and previous cell states. The result would ascertain whether to amplify or attenuate the original value. As a cross product with the hyperbolic tangent function of previous cell states, the second sigmoid function constrains the input to the [−1, 1] range.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

where $\tilde{C}_t$ implies the candidate for cell state at timestamp(t), C_t denotes the cell state at timestamp(t), and other mirror Eq 1.

The sigmoid output (amplifier or attenuator) was subsequently utilized to scale the encoded data based on its appearance pre-application to the cell state. Plausibly, the inclusion of such features to render the present state essential to recall implies their reference as input gates in (Eq 1), which would later be integrated with the forget gate from (Eq 3) to form the new cell state (C_t) as expressed in (Eq 4). A hyperbolic tangent was applied to the new cell state in compressing the values into the [-1, 1] range as presented in (Eq 5). Lastly, the cross-product with the input sigmoid function was expressed in (Eq 6).

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

where o_t represents the output gate and h_t depicts the LSTM block output at timestamp(t).

This outcome formed the current recurrent network output, which also became the hidden state for the subsequent network. At this point, the LSTM model output provided the temporal features derived from the sequence of cropped eyes. Regarding the final prediction state, each LSTM neuron output was sent to a neural network constituting a fully connected layer, which incorporated the LSTM output and generated the probabilities of both open and closed eye states.

Blink Probability Dataset (BPD)

The eye state probability values gathered in the Blink Detection System section were stored in a temporal sequence that paralleled the input source frame rate. Each video sequence encompassed a modified frame rate and duration with the probability sequence for each video also undergoing changes. As such, it proves necessary to standardize the frame rate for the blinking pattern to be used as a legitimate feature in system training and the incoming data to have a specific length of sequence before proceeding. An input sequence that is too long would require trimming within a particular sliding window formation. The frame rate for each probability sequence was converted to 50 frames per second. Any data in between were resampled using the value from the prior data. As a result, the sliding windows are set at 4.5 seconds, resulting in 225 eye-state probabilities in each sequence, so ensuring that there is the opportunity for the eyes to blink at least once within the trimmed sequence.

The final (BPD) dataset had the input marked with a 0 for an authentic video and 1 for a tampered one (see Fig 6). Each input constituted 225 features ranging from the second to the 226^{th} column. These properties stored the probability of blinking with values ranging between 0 (eye fully open) and 1 (eye fully closed). The 225 probability values reflected the eye condition for 4.5 seconds, during which at least one eye blink would normally occur. As stated in the final stage, the whole dataset was fed into the study models for preparation, validation, and testing purposes.

DeepFake classifier

An FCN is typically defined to provide appropriate discrimination of features. Specifically, the FCN output denotes a predicted classification label. All inputs pass through the fully connected layer with a separate weight applied for each connected neuron. Selecting the most optimal combination of layers would offer an optimum network with minimal calculation cost.

The operations performed by the FCN are presented below:

$$n_0^{out} = I \tag{7}$$

$$n_0^{in} = n_0^{out} * W_i + B_i \tag{8}$$

$$n_i^{out} = F_i(n_i^{in}) \tag{9}$$

where F_i denotes an activation function for layer i. With this formula, a forward pass could be executed and the network output produced for each network layer.

Backpropagation was then applied to update all the weights and biases. The weight and bias update process in this study involved Diederik and Jimmy et al.'s adaptive moment estimation [7], whereas stochastic gradient descent served to update the neural network parameters. Meanwhile, adaptive gradients (AdaGrad) offered a direct means of gradually varying the learning rate to accommodate dataset changes, as small- or large-scale shifts are possible based on how the learning rate is selected. The equation is expressed as follows:

$$\theta_{t+1,i} = \theta_{t,1} - \frac{\eta}{\sqrt{\epsilon + \sum_{\tau=1}^{t} (\nabla J(\theta_{\tau,i}))^2}} \nabla J(\theta_{t,i}) \tag{10}$$

where θ implies a parameter consisting of the weight, biases and activation,η reflects the learning 258 rate, ∇ denotes the gradient, and J is the objective function with its features and labels.

To initialize the calculation of the error gradients, it is necessary to provide an error calculation for determining the losses. This work uses the cross-entropy loss as it is widely adopted by

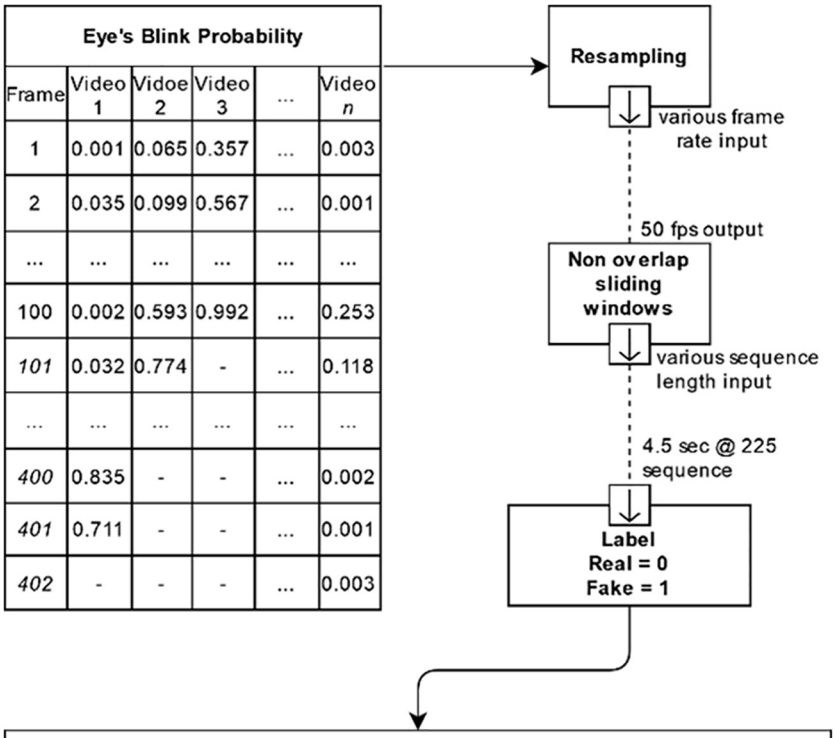

Fig 6. Processing the raw data into a complete BPD.

https://doi.org/10.1371/journal.pone.0278989.g006

researchers in this field. The cross-entropy loss L is given by

$$L = -\sum_{j}^{C} y_j \ln \hat{y}_j \tag{11}$$

where C represents the number of classes, y denote the labels, and $\hat{y}$ imply the predicted labels.

The loss, which dictated the gradient of the backpropagation steps, was updated based on the selected learning rate. A batch of inputs were fed into the network during each epoch with

Table 1. Layers of FCN models used.

Layer	input	h_1	h_2	h_3	h_4	h_5	h_6	output
*Model*1	225	512	1024	256	64			2
*Model*2	225	512	1024	128				2
*Model*3	225	512	1024	2048	1024	512	256	2

https://doi.org/10.1371/journal.pone.0278989.t001

the weights and biases updated at the end of the epoch. The final accuracy and loss value were computed as FCN performance measures.

Based on the newly-established BPDs, each input constitutes 225 features reflecting the eye blinking state every 0.02 seconds, thus providing a total duration of 4.5 seconds for the eye sequence. It is designed so that at least one eye blinking condition would be present in each sequence. The sequences were fed to FCN networks with a range of different layer designs, each with a 0.1 dropout. The rectified linear unit (ReLU) selected as the activation function was applied at the output of each layer. The probability used in the prediction of the video ingenuity status was provided by the output layer values.

The process for distinguishing between genuine and tampered videos in this research was derived by feeding the BPD into three different FCN architectures. In line with Table 1, each model has its own number of hidden layers and individual number of nodes in each layer.

Other than the models used for testing purposes, each model was executed with a distinct set of settings and parameters. The number of epochs in Experiment 1 was set to 100, 300, and 500. Experiment 2 incorporated an early stopping function into each model with patience values of 20, 40, 60, 80, and 100, whereas experiment 3 demonstrated the result of adjusting the batch value from 5, 10, 15, 20, and 25 to achieve the highest precision for the epoch setting derived from experiments 1 and 2. All the experiments assessed the model accuracy with the outcomes and the analysis elaborated in the following section.

Results and discussion

The current study datasets were established through the `FaceForensics++` dataset and converted into probability values for the eye-blinking state, which were subsequently trimmed to fit the required 225 values of the input data sequence. This new data collection trained the model and validated the classification of real and fake videos. This experimental study encountered several limitations. For example, videos with more than one individual were excluded as the method only analyzed one face at a time. The selected face was then pre-processed with only the left and right eye areas included in the training model. The dataset source was elicited from the FaceForensics++ dataset. Furthermore, five different tools involving DeepFake, DeepfakeDetection, Face2Face, FaceSwap, and Neural Textures were employed to generate the Deepfake media although the media source was restricted to a single dataset. In this vein, the high levels of diversity and complexity were regarded `In the Wild` dataset.

Dataset

A total of 451 videos consisting of 200 real videos and 251 tampered ones were generated. Each video represented one individual with at least one blink occurring to form the BPDs. The datasets were prepared with Yuezun Li et al.'s [11] annotation tools where each video provided two CSV files with timestamps determined based on video length and frame per second (fps). These files were subsequently processed to form part of the final BPD dataset.

Eye blinking extraction result using CNN-LSTM pre-trained model

Each video was analyzed before being fed to the proposed model. The pre-trained CNN-LSTM model omitted the eye blinking state for each frame. Specifically, the model predicted a probability value for every (open/closed) state for each eye and frame in each clip. Fig 7 illustrates the model operation. The depicted individuals in Figs 7 and 10 had provided written informed consent in compliance with the PLOS consent form to publish their image alongside the manuscript.

The CNN used in this work is based on VGGNet-16 model (see Fig 8) and comprised 16 convolutional layers. Pre-training was performed on the model with ImageNet datasets. This model was selected given its outstanding precision in eliminating features from still images and ability to distinguish between apparent changes in the eye size when opening and closing.

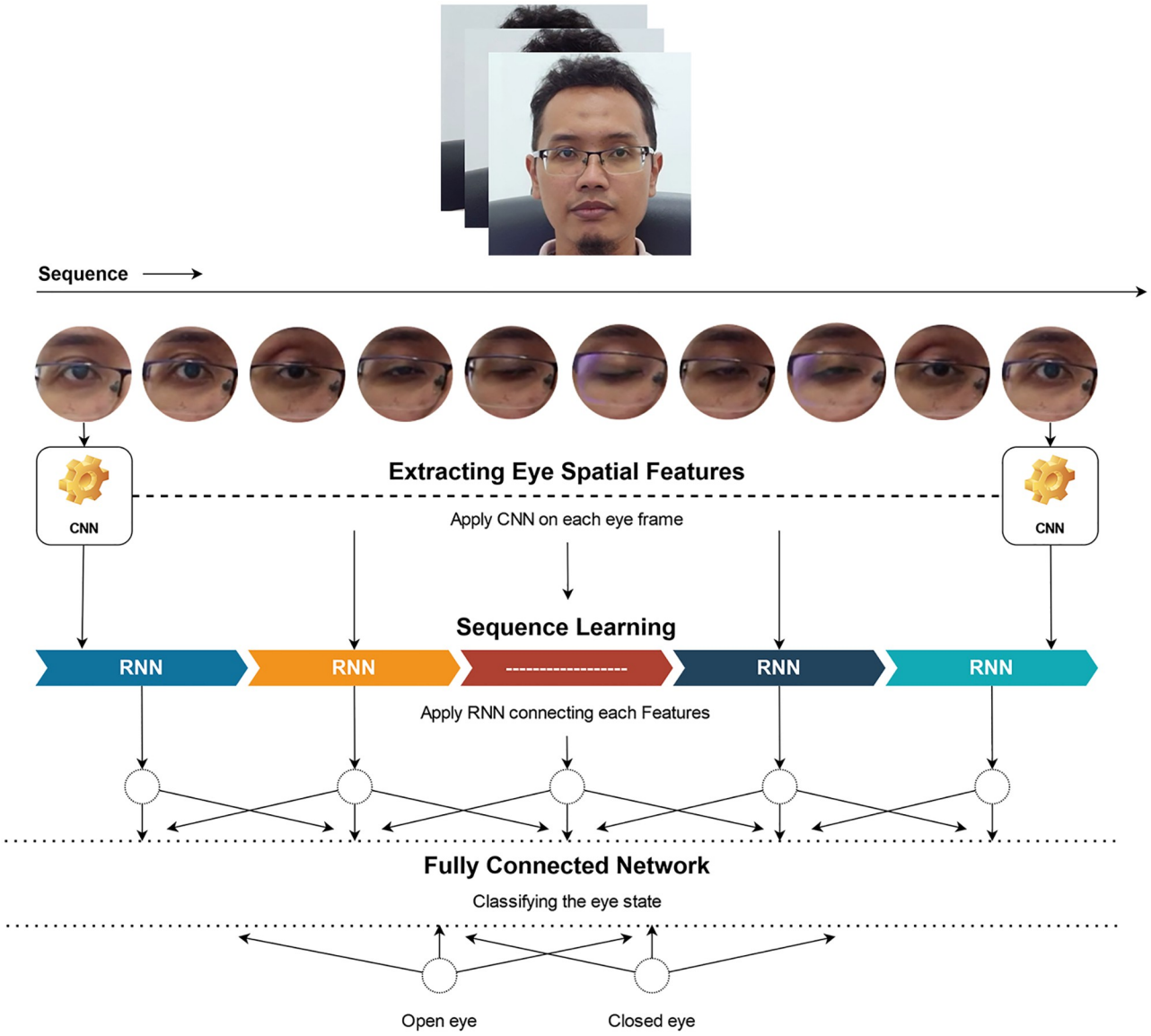

Fig 7. Pre-trained CNN-LSTM model for eye state prediction.

https://doi.org/10.1371/journal.pone.0278989.g007

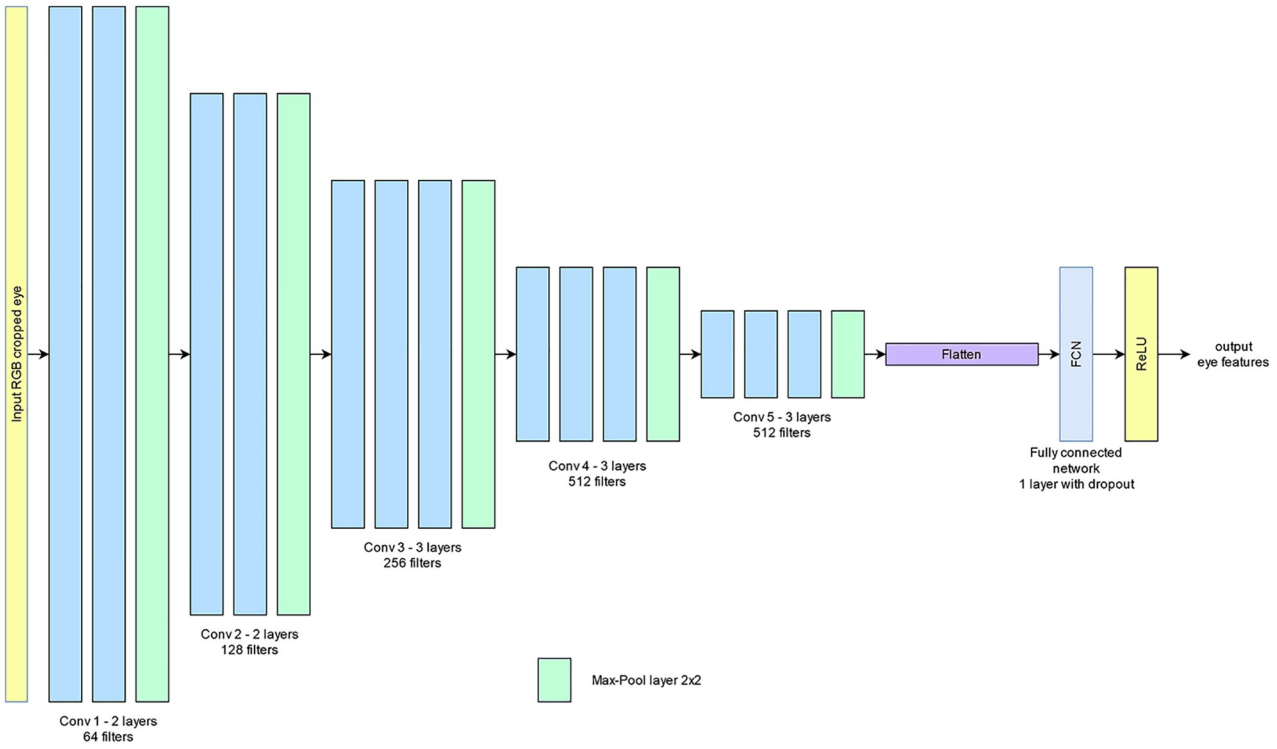

Fig 8. The VGGNet-16 model for extracting eyes spatial features.

https://doi.org/10.1371/journal.pone.0278989.g008

A technique with sole reliance on CNN to train the pixel pattern of the eyes and the system is generally incapable of ascertaining whether the eyes were in the closing or opening state. Predictions on changing eye state could be improved through an RNN that incorporates temporal features. The CNN output was reshaped into 20 sequences and fed to a RNN with the LSTM variant. Notably, LSTM could extract the relationship between the temporal features of the sequences for each set of the 20 spatial features elicited from CNN. The performance measures derived for the many-to-many LSTM network was routed through FCN for classification. Fig 9 illustrates the complete architecture.

A technique that relies solely on CNN to train the pixel pattern of the eyes and the system is often unable to distinguish whether the eyes were in a state of closing or opening. The prediction of changing eye state can be improved by using a RNN that incorporates, temporal features.

The video signal, which was fed into the network, yielded a probability value for the eye state ranging between 0 (eye fully open) and 1 (eye fully closed). In line with Fig 10, the CNN-LSTM network precisely predicted the eye probability that represented the eye-blinking status on both sides.

Fig 11 depicts an individual's blinking sequence from the original video versus the DeepFake-generated counterpart. Observably, the generated eye sequence between opening and closing proved intermittent: an unnatural blinking pattern compared to the original sequence. The anomaly in the fake video was amplified by the temporal features of the blinking sequence and fed into the train model.

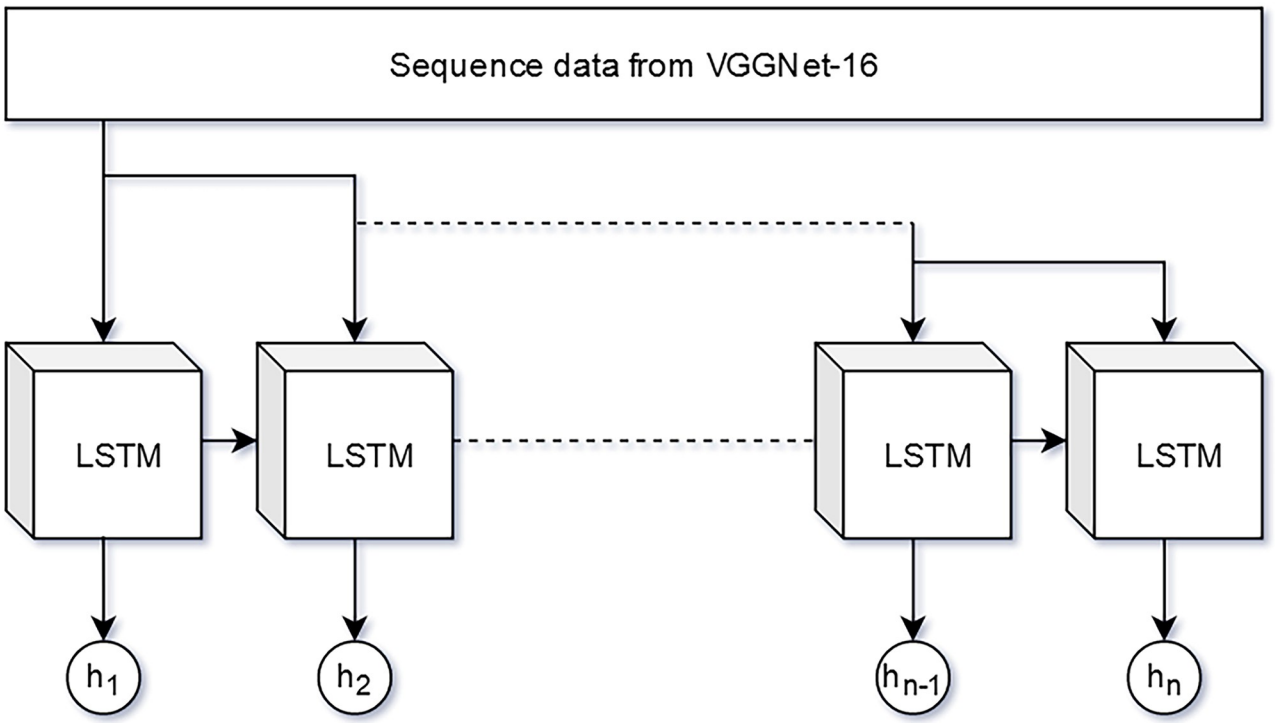

Fig 9. The LSTM network for extracting temporal features of eye-blinking state.

https://doi.org/10.1371/journal.pone.0278989.g009

Classification

The BPD dataset elicited in the Dataset section was fed into the model for training purposes. Particularly, the dataset was divided into (i) training-evaluating and (ii) testing sets. The training-evaluation set encompasses 3484 data values for both real and fake videos with 70% of the set utilized for training, 15% for validation and 15% for testing. The *train*, *validate*, and *test* `Dataloaders` were then distributed using a random split. Dropout with a value of 0.1 was used for each neural network layer to avoid overfitting cases. The learning rate of 0.0004 was selected with ADAM incorporated as the optimizer. Meanwhile, the random seed was set to 42 in each epoch in standardizing the initial weight and demonstrating a valid comparison.

The input data length was fixed at 225 data points during training. Each hidden layer passes was fully connected, albeit with a different number of nodes and a specific rectified linear activation function. The FCN generated a SOFTMAX output that implied the media authenticity

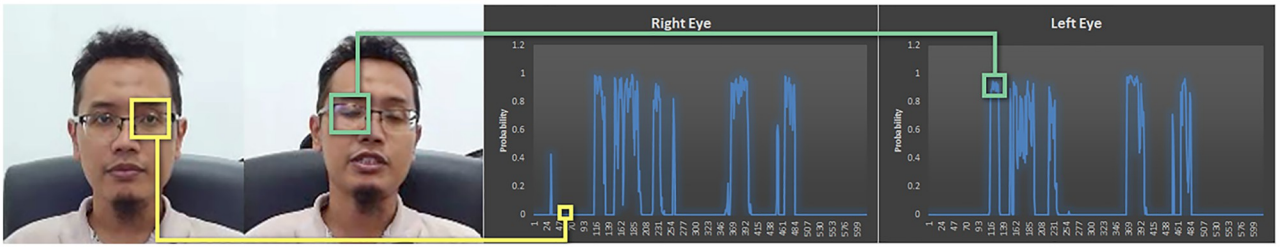

Fig 10. Result of CNN-LSTM network for every frame in the video input.

https://doi.org/10.1371/journal.pone.0278989.g010

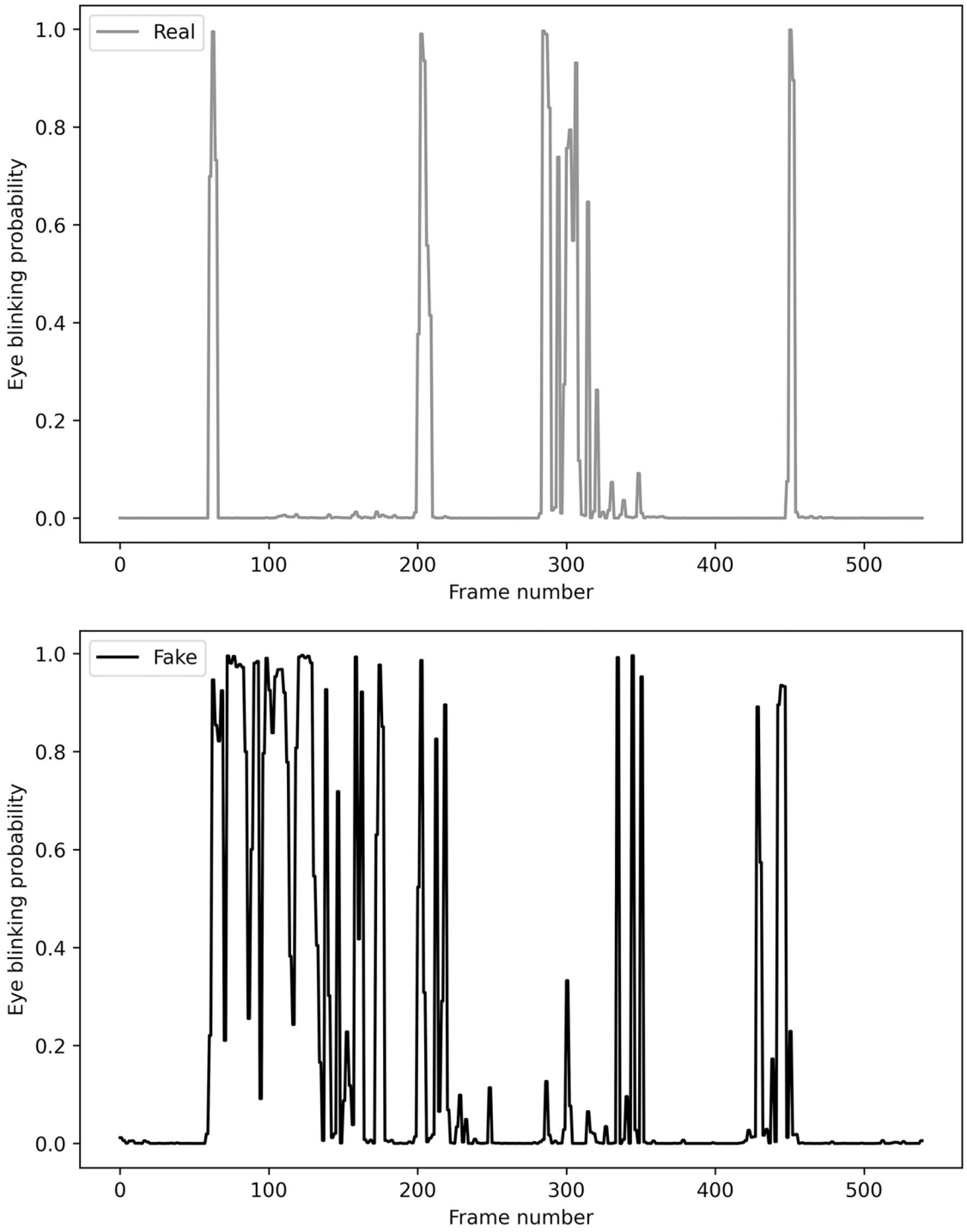

Fig 11. Sample of blink sequence pattern for real and fake video.

https://doi.org/10.1371/journal.pone.0278989.g011

Table 2. Test accuracy for a range of epoch values in three FCN models.

Number of Epoch	100	200	300	400	500
*Model*1	82.18%	84.29%	83.52%	**88.31**%	86.97%
*Model*2	82.76%	84.10%	84.11%	85.82%	82.57%
*Model*3	80.65%	84.29%	84.10%	86.20%	83.14%

https://doi.org/10.1371/journal.pone.0278989.t002

or fakeness. Several training cycles were attempted with various batch sizes, epoch counts, and early stopping mechanisms. The training model was evaluated for each set of experiments by considering the *test_acc* and *test_loss* derived from the trained model. Notably, this model employed the same dropout value and activation mechanism, albeit with differing layers between the study models (see Table 1).

Experiment 1: Epoch. The first experiment was run with a batch size of 15 and no early stopping. Nevertheless, each model was trained for 100, 200, 300, 400, and 500 epochs. Table 2 presents the elicited outcomes.

Based on the test dataset accuracy, all the models could deliver over 80% of accuracy with Model 1 demonstrating the highest at 88.31% when predicting real or fake films through 400 epochs. Both Models 2 and 3 also performed optimally with 400 epochs. Although Model 3 depicted the most hidden layers with prolonged training time, it failed to provide a significantly improved performance.

Experiment 2: Early stopping. The second experiment was conducted by fixing the batch size at 15. Regardless, the early stopping incorporated into each model stopped at a specific number of epochs upon meeting the final condition. The *patience* values used were set at 20, 40, 60, 80, and 100. Table 3 presents the elicited outcomes.

Although the early stopping function rapidly completed the training, the degree of rapidity depended on the validation accuracy value requirement. This approach could generate a better model with minimal training time. For example, Model 1 demonstrated an outstanding result of 89.66% in test accuracy when performed with a patience value of 100, whereas Models 2 and 3 failed to attain better accuracy in any patience setting. It is deemed possible to cease training early when a large neural network could generalize in a manner comparable to a smaller counterpart. This ability could efficiently minimize calculation time and generate optimal performance.

Experiment 3: Batch size. The final experiment fixed the epoch number that provided the most optimal outcomes in Experiments 1 and 2. The employed batch sizes were 5, 10, 15, 20, and 25 (see Table 4).

With a batch size of 20 and 105 epochs, Model 1 offered the most optimal accuracy value of 90.80%. All the models denoted little variation in terms of test accuracy. As the number of connected layers was reduced, the model could rapidly achieve convergence to better minima. Larger batch sizes expanded the training to include additional compute nodes, which would save energy in reduced computation efforts.

Table 3. Test accuracy for early stopping with patience setting of 20, 40, 60, 80, and 100 on three FCNs models.

Patience	20	40	60	80	100
*Model*1	81.61%	83.72%	84.29%	85.05%	**89.66**%
*Model*2	81.42%	79.89%	74.33%	81.99%	81.23%
*Model*3	79.69%	83.33%	83.14%	83.33%	83.33%

https://doi.org/10.1371/journal.pone.0278989.t003

Table 4. Test accuracy for a batch size of 5, 10, 15, 20, and 25 at best epoch number for each FCNs models.

	Batch Size Epoch	5	10	15	20	25
*Model*1	105	90.04%	90.61%	89.66%	**90.80**%	89.85%
*Model*2	400	86.59%	85.25%	85.82%	84.67%	84.67%
*Model*3	400	85.44%	86.78%	86.20%	86.78%	86.59%

https://doi.org/10.1371/journal.pone.0278989.t004

Fig 12 presents a summary of the experimental outcomes. Despite having fewer hidden layers than other models, Model 1 reflected the most optimal performance in detecting fake video sequences. Establishing early stopping as an extra callback enables specific models to end the learning process earlier, alleviate training costs, and minimize time consumption. Notwithstanding, the end model in this study relied on an FCN that rendered early stopping ineffective as observed in Experiment 2. Selecting a specific batch size could significantly improve accuracy while saving time during the training session. Following the results derived from all three experiments, Model 1 with a batch size of 15 offered the highest accuracy without depending on an early stopping mechanism.

Table 5 presents several approaches to the same problem. Using only spatial information, Guo et al.'s incorporation of AMTEN and CNN provided 87.05% accuracy post-testing on their own datasets [43]. This accuracy significantly improved when RGB information was utilized as input along with the noise features extracted by a spatial rich model, hence resulting in 90.36% accuracy assessed on a still image in the FaceForensics++ dataset [44]. Regardless, insufficient temporal information deterred the model from performing optimally on sequence data. Integrating temporal information as additional features enables one to examine the inconsistency between frames as the change occurs at the single frame level. Although the performance of [45] and [47] on the FaceForensics++ dataset proved slightly lower than the spatial-based technique, merely utilizing the basic model to extract both spatial and temporal features could still provide a substantial outcome with accuracy up to 85.80% [46].

In terms of overall performance, the model probability of failing to forecast the test video remained significant. Based on the model evaluation of false predictions, most cases occurred when the generated video generated an eye sequence that closely resembled the source. Fig 13 illustrates the relative similarity of the generated eye sequence with that of the original eye. Perceivably, the generated eye sequence between opening and closing states was more instantaneous and delayed than the original sequence despite a minute difference between them.

Conclusion

The current study introduced a novel means of exposing fake videos created with deep neural networks, which depended on identifying eye blinking in videos: a physiological signal not typically included in fake videos. This approach, which was tested on datasets containing sequences that include eye-blinking, demonstrated optimal outcomes in detecting the fake videos created with the DeepFake-based programme. The method could distinguish between real and fake image sequences with up to 90.8% accuracy with Model 1 and a batch size of 20 at 105 epochs. DeepFakeDetection-generated media could be identified with up to 95.57% accuracy (the highest percentage) in the FaceForensics++ dataset, followed by DeepFakes, FaceSwap, and Face2Face at 94.65%, 91.54%, and 90.37%, respectively. Neural Textures-generated media denoted the lowest model accuracy performance at 86.76%.

The researchers intend to take this study in several distinctive directions for improved performance. First, alternative deep neural network architectures require further examination to determine the presence of more effective training methods to identify eye-blinking patterns.

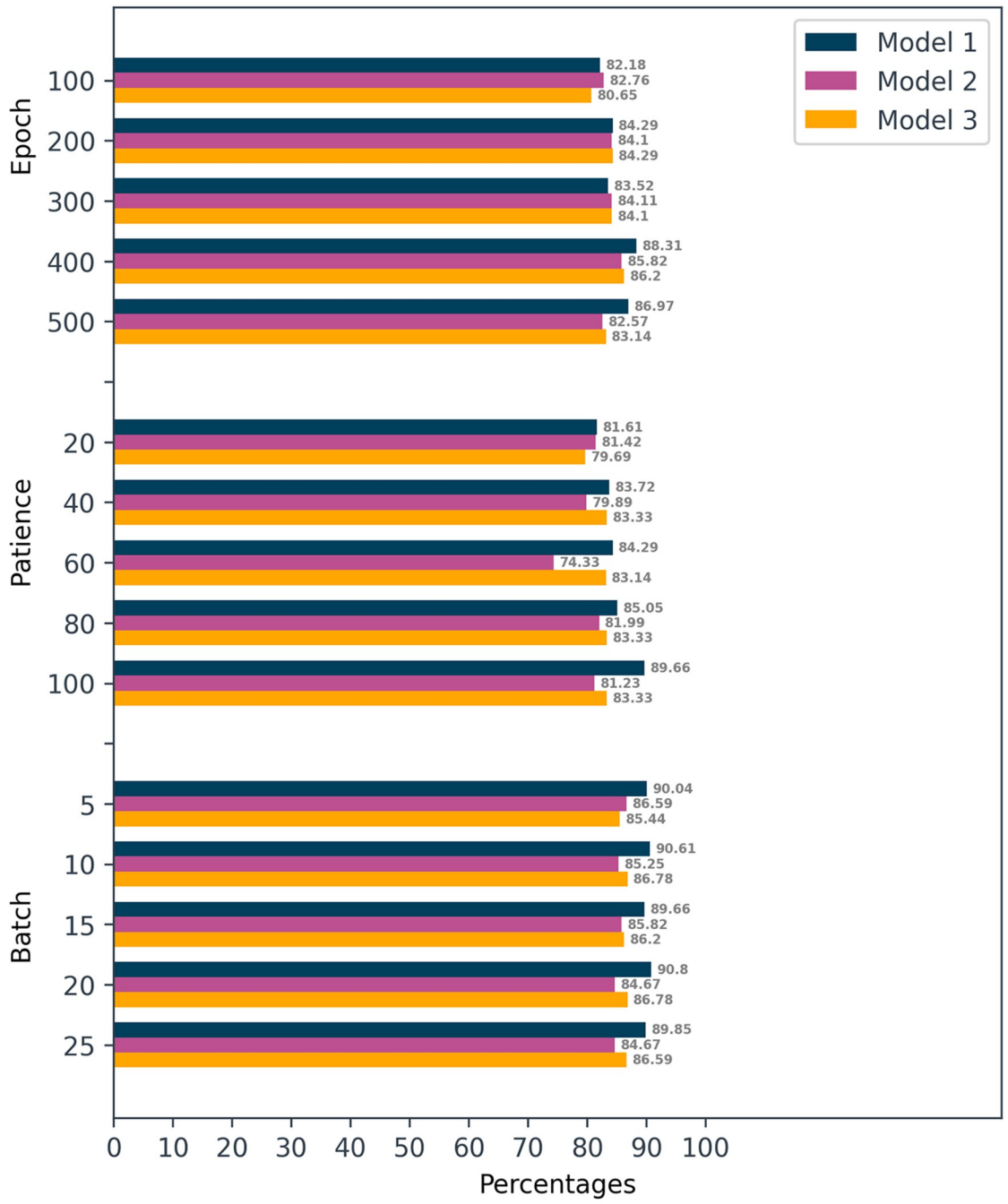

Fig 12. Comparative performance of the three models.

https://doi.org/10.1371/journal.pone.0278989.g012

Second, more complex and complete rhythms of blinking, such as physiologically-impossible and excessive blinking that may indicate tampering could be included as the blinking state is the only input in the current method. Lastly, eye-blinking is merely a basic cue to detect fake face images as forgers would produce convincing blinking effects with post-processing and advanced models trained using blinking image sequence once the detection techniques gains

Table 5. Performance comparison with other detection technique.

Dataset	Architecture	Classifier	Strategy	Accuracy
Face RGB and Own Dataset	AMTEN with CNN [43]	FCN	Spatial	87.05%
Face RGB FF++	2 stream XceptionNet (pixel & filtered) [44]	FCN	Spatial	90.36%
Face RGB, FF++	Optical flow feed into ResNet50 [45]	FCN	Temporal	80.56%
Face RGB, FF++, Deepfake-TIMIT dataset	Use spatial angle and temporal rotation as classifier input [46]	SVM	Spatial Temporal	85.80%
Face RGB seq, CelebDF, FF++ dataset	2 stream: MesoNet + ResNet [47]	FCN	Spatial Temporal	80.00%

https://doi.org/10.1371/journal.pone.0278989.t005

popularity. Thus, it proves necessary to consider alternative physiological signals that could distinguish a real image sequence from those generated by AI synthesis methods.

The study is concluded as follows:

1. Using a novel dataset potentially accelerated the training process by lowering the calculation requirements and reducing the memory needed by the GPU in the training process.
2. By providing the eye state probability as the FCN model input, the trained model could provide optimal results with up to 90.80% accuracy.
3. Early stopping can provide good models faster and automatically but prevent the establishment of a better model, which could be derived by training over more epochs.
4. Based on the most complex model (Model 3), large batch size did not provide a significant improvement. Nevertheless, controlling the batch size proved pivotal for Models 1 and 2

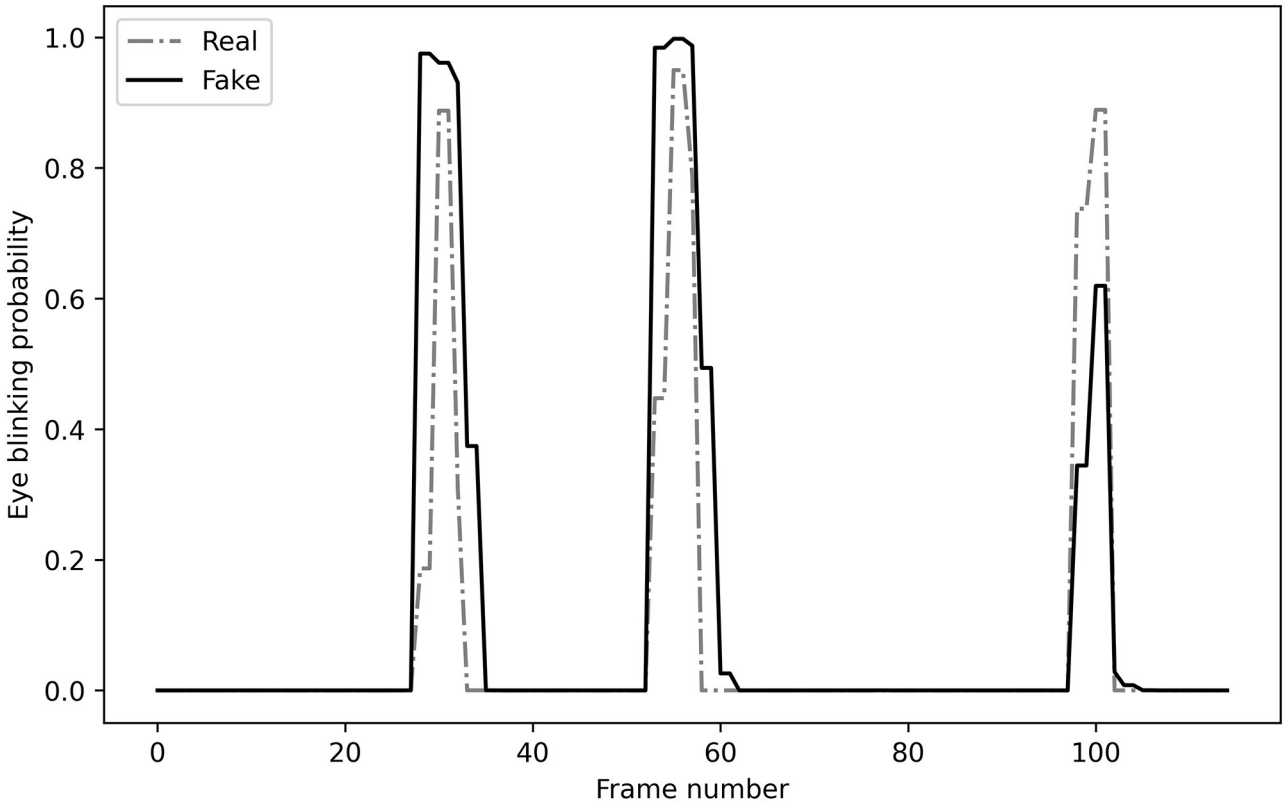

Fig 13. Comparison between real and fake eye blinking sequence for false prediction output.

https://doi.org/10.1371/journal.pone.0278989.g013

with fewer hidden layers to ensure that they were not overfitting to the training and validation datasets.

Future works

Significant advancements have been identified in multimedia forensics over the last 16 years. The establishment of new detection methods is a continuous process given the perpetuity of various unresolved issues and challenges. In this vein, deep learning catalysed the development of both media manipulation techniques and forensic technologies. An FCN proved suitable for classification with no specific assumptions on the inputs. Notwithstanding, an input with more dimensions or features would lead to an increase in the number of weights slow training time, and high GPU memory usage. Alternative classification techniques, such as CNNs could accelerate the process and minimize the possibility of overfitting during training. Overall, this study generated empirical results with time domain input features. By pre-processing the datasets to produce frequency domain features, additional temporal information could be provided to the training process for enhanced performance.

Author Contributions

Conceptualization: Mohd Zamri Ibrahim.

Funding acquisition: Mohd Zamri Ibrahim.

Investigation: Muhammad Salihin Saealal.

Methodology: Muhammad Salihin Saealal.

Software: Muhammad Salihin Saealal.

Supervision: Mohd Zamri Ibrahim.

Validation: David. J. Mulvaney, Mohd Ibrahim Shapiai, Norasyikin Fadilah.

Visualization: Norasyikin Fadilah.

Writing – original draft: Muhammad Salihin Saealal.

Writing – review & editing: Mohd Zamri Ibrahim, David. J. Mulvaney, Mohd Ibrahim Shapiai.

References

1. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. Communications of the ACM. 2020; 63(11):139–144. https://doi.org/10.1145/3422622
2. Liu MY, Breuel T, Kautz J. Unsupervised Image-to-Image Translation Networks. In: Advances in Neural Information Processing Systems. vol. 2017-December. Neural information processing systems foundation; 2017. p. 701–709. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041131132&partnerID=40&md5=8d363aab109ab94adfb2ddc778d61090.
3. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2016-December. IEEE Computer Society; 2016. p. 770–778. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84986274465&doi=10.1109%2fCVPR.2016.90&partnerID=40&md5=f67e8d2a623bac88aad535d2c0a6d374.
4. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997; 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276
5. Isola P, Zhu JY, Zhou T, Efros AA. Image-to-Image Translation with Conditional Adversarial Networks. In: Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. vol. 2017-January. Institute of Electrical and Electronics Engineers Inc.; 2017. p. 5967–5976. Available

from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85030759098&doi=10.1109%2fCVPR.2017.632&partnerID=40&md5=fb2a5fe5a1479af939bffe544bd49dcd.

6. Kim KW, Hong HG, Nam GP, Park KR. A Study of Deep CNN-Based Classification of Open and Closed Eyes Using a Visible Light Camera Sensor. Sensors (Switzerland). 2017; 17(7). https://doi.org/10.3390/s17071534 PMID: 28665361
7. Kingma DP, Lei Ba J. Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings. International Conference on Learning Representations, ICLR; 2015. Available from: https://arxiv.org/abs/1412.6980.
8. Verdoliva L. Media Forensics and DeepFakes: An Overview. IEEE Journal on Selected Topics in Signal Processing. 2020; 14(5):910–932. https://doi.org/10.1109/JSTSP.2020.3002101
9. Wang SY, Wang O, Zhang R, Owens A, Efros AA. CNN-Generated Images Are Surprisingly Easy to Spot. . . for Now. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. p. 8692–8701.
10. Li Y, Lyu S. Exposing DeepFake Videos By Detecting Face Warping Artifacts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops; 2019.
11. Li Y, Chang MC, Lyu S. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In: 10th IEEE International Workshop on Information Forensics and Security, WIFS 2018. Institute of Electrical and Electronics Engineers Inc.; 2019. p. 1–7. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062882418&doi=10.1109%2fWIFS.2018.8630787&partnerID=40&md5=3dec5b984609ccca745fca39a2bfe4fb.
12. Chen YL, Hsu CT. Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection. IEEE Transactions on Information Forensics and Security. 2011; 6 (2):396–406. https://doi.org/10.1109/TIFS.2011.2106121
13. Ferrara P, Bianchi T, De Rosa A, Piva A. Image Forgery Localization via Fine-Grained Analysis of CFA Artifacts. IEEE Transactions on Information Forensics and Security. 2012; 7(5):1566–1577. https://doi.org/10.1109/TIFS.2012.2202227
14. Zhang W, Zhao C, Li Y. A Novel Counterfeit Feature Extraction Technique for Exposing Face-Swap Images Based on Deep Learning and Error Level Analysis. Entropy. 2020; 22(2). https://doi.org/10.3390/e22020249 PMID: 33286023
15. Zou H, Yang P, Ni R, Zhao Y. Anti-Forensics of Image Contrast Enhancement Based on Generative Adversarial Network. Security and Communication Networks. 2021; 2021. https://doi.org/10.1155/2021/6663486
16. Bestagini P, Milani S, Tagliasacchi M, Tubaro S. Local Tampering Detection in Video Sequences. In: 2013 IEEE International Workshop on Multimedia Signal Processing, MMSP 2013; 2013. p. 488–493. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84892505453&doi=10.1109%2fMMSP.2013.6659337&partnerID=40&md5=00f115de87a906874b2d1ff4228aa4cb.
17. Dirik AE, Karaküçük A. Forensic Use of Photo Response Non-uniformity of Imaging Sensors and a Counter Method. Optics Express. 2014; 22(1):470–482. https://doi.org/10.1364/OE.22.000470 PMID: 24515007
18. Sameer VU, Naskar R, Modalavalasa S. Mitigating Adaptive PRNU Denoising in Camera Model Identification: An Anti–Counter Forensic Approach. In: IEEE Region 10 Annual International Conference, Proceedings/TENCON. vol. 2019-October. Institute of Electrical and Electronics Engineers Inc.; 2019. p. 903–907. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85077713715&doi=10.1109%2fTENCON.2019.8929355&partnerID=40&md5=e8868763f6d83a84ba75073be88ef061.
19. Xie C, Tan M, Gong B, Wang J, Yuille AL, Le QV. Adversarial Examples Improve Image Recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2020. p. 816–825. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85093074996&doi=10.1109%2fCVPR42600.2020.00090&partnerID=40&md5=f0d08e1eacc4fb6a452becba5868efca.
20. Bappy JH, Simons C, Nataraj L, Manjunath BS, Roy-Chowdhury AK. Hybrid LSTM and Encoder–Decoder Architecture for Detection of Image Forgeries. IEEE Transactions on Image Processing. 2019; 28(7):3286–3300. https://doi.org/10.1109/TIP.2019.2895466 PMID: 30703026
21. Barni M, Bondi L, Bonettini N, Bestagini P, Costanzo A, Maggini M, et al. Aligned and Non-aligned Double JPEG Detection Using Convolutional Neural Networks. Journal of Visual Communication and Image Representation. 2017; 49:153–163. https://doi.org/10.1016/j.jvcir.2017.09.003
22. Wu J, Liu L, Kang X, Sun W. A Generative Adversarial Network Framework for JPEG Anti-Forensics. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2020—Proceedings. Institute of Electrical and Electronics Engineers Inc.; 2020. p. 1442–1447. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100919989&partnerID=40&md5=dd7789183499aa6683b62c0c5c977229.

23. Tariq S, Lee S, Kim H, Shin Y, Woo SS. GAN is a Friend or Foe? A Framework to Detect Various Fake Face Images. In: Proceedings of the ACM Symposium on Applied Computing. vol. Part F147772. Association for Computing Machinery; 2019. p. 1296–1303. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85065671310&doi=10.1145%2f3297280.3297410&partnerID=40&md5=edbd72fc9f298c514102ecb02187645a.

24. Maksutov AA, Morozov VO, Lavrenov AA, Smirnov AS. Methods of Deepfake Detection Based on Machine Learning. In: Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2020. Institute of Electrical and Electronics Engineers Inc.; 2020. p. 408–411. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85082991408&doi=10.1109%2fEIConRus49466.2020.9039057&partnerID=40&md5=8b87feae61afa027b45dca27840204b3.

25. Pan G, Sun L, Wu Z, Lao S. Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera. In: Proceedings of the IEEE International Conference on Computer Vision; 2007. p. 1–8. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-50949086465&doi=10.1109%2fICCV.2007.4409068&partnerID=40&md5=39692c7cae0151182976a40fa0077e91.

26. Jung T, Kim S, Kim K. DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern. IEEE Access. 2020; 8:83144–83154. https://doi.org/10.1109/ACCESS.2020.2988660

27. Yang F, Yu X, Huang J, Yang P, Metaxas D. Robust Eyelid Tracking for Fatigue Detection. In: Proceedings—International Conference on Image Processing, ICIP; 2012. p. 1829–1832. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84875865452&doi=10.1109%2fICIP.2012.6467238&partnerID=40&md5=a1f96e4e7685478f02fb36cb5ce48720.

28. Nguyen TT, Nguyen QVH, Nguyen DT, Nguyen DT, Huynh-The T, Nahavandi S, et al. Deep Learning for Deepfakes Creation and Detection: A Survey. Computer Vision and Image Understanding. 2022; 223. https://doi.org/10.1016/j.cviu.2022.103525

29. Denton E, Chintala S, Szlam A, Fergus R. Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks. In: Advances in Neural Information Processing Systems. vol. 2015-January. Neural information processing systems foundation; 2015. p. 1486–1494. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84965143571&partnerID=40&md5=ccb301abee9ef185cb0df79a0ed7c0d7.

30. Burt PJ, Adelson EH. The Laplacian Pyramid as a Compact Image Code. In: Fischler MA, Firschein O, editors. Readings in Computer Vision. San Francisco (CA): Morgan Kaufmann; 1987. p. 671–679. Available from: https://www.sciencedirect.com/science/article/pii/B9780080515816500659.

31. Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: 4th International Conference on Learning Representations, ICLR 2016—Conference Track Proceedings. International Conference on Learning Representations, ICLR; 2016. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083950271&partnerID=40&md5=d56c1a588cadb82fd3fc0f705a144a73.

32. Arjovsky M, Chintala S, Bottou L. Wasserstein Generative Adversarial Networks. In: 34th International Conference on Machine Learning, ICML 2017. vol. 1. International Machine Learning Society (IMLS); 2017. p. 298–321. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85047016172&partnerID=40&md5=7fde6f9714ef8880afa04f3ed13c7234.

33. Shrivastava A, Pfister T, Tuzel O, Susskind J, Wang W, Webb R. Learning from Simulated and Unsupervised Images through Adversarial Training. In: Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. vol. 2017-January. Institute of Electrical and Electronics Engineers Inc.; 2017. p. 2242–2251. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041912723&doi=10.1109%2fCVPR.2017.241&partnerID=40&md5=67b60a2b6177dba9328c2026ea55a252.

34. Azim T, Jaffar MA, Mirza AM. Fully Automated Real Time Fatigue Detection of Drivers Through Fuzzy Expert Systems. Applied Soft Computing Journal. 2014; 18:25–38. https://doi.org/10.1016/j.asoc.2014.01.020

35. Mandal B, Li L, Wang GS, Lin J. Towards Detection of Bus Driver Fatigue Based on Robust Visual Analysis of Eye State. IEEE Transactions on Intelligent Transportation Systems. 2017; 18(3):545–557. https://doi.org/10.1109/TITS.2016.2582900

36. Sukno FM, Pavani SK, Butakoff C, Frangi AF. Automatic Assessment of Eye Blinking Patterns through Statistical Shape Models. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2009; 5815 Lncs:33–42.

37. Cootes TF, Taylor CJ, Cooper DH, Graham J. Active Shape Models—Their Training and Application. Computer Vision and Image Understanding. 1995; 61(1):38–59. https://doi.org/10.1006/cviu.1995.1004

38. Cech J, Soukupova T. Real-time Eye Blink Detection using Facial Landmarks. Cent Mach Perception, Dep Cybern Fac Electr Eng Czech Tech Univ Prague. 2016; p. 1–8.

39. Li B, Lima D. Facial Expression Recognition via ResNet-50. International Journal of Cognitive Computing in Engineering. 2021; 2:57–64. https://doi.org/10.1016/j.ijcce.2021.02.002

40. Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017; 39(4):677–691. https://doi.org/10.1109/TPAMI.2016.2599174 PMID: 27608449

41. Zhou T, Wang W, Liang Z, Shen J. Face Forensics in the Wild. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2021. p. 5774–5784. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85115136497&doi=10.1109%2fCVPR46437.2021.00572&partnerID=40&md5=2481dada26c9801025e7a337f7891eb8.

42. Kazemi V, Sullivan J. One Millisecond Face Alignment with an Ensemble of Regression Trees. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Cvpr'14. Usa: IEEE Computer Society; 2014. p. 1867–1874. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84911391543&doi=10.1109%2fCVPR.2014.241&partnerID=40&md5=a3223af23757fd145907bf1b229c16bb.

43. Guo Z, Yang G, Chen J, Sun X. Fake Face Detection via Adaptive Manipulation Traces Extraction Network. Computer Vision and Image Understanding. 2021; 204:103170. https://doi.org/10.1016/j.cviu.2021.103170

44. Han B, Han X, Zhang H, Li J, Cao X. Fighting Fake News: Two Stream Network for Deepfake Detection via Learnable SRM. IEEE Transactions on Biometrics, Behavior, and Identity Science. 2021; 3(3):320–331. https://doi.org/10.1109/TBIOM.2021.3065735

45. Caldelli R, Galteri L, Amerini I, Del Bimbo A. Optical Flow based CNN for Detection of Unlearnt Deepfake Manipulations. Pattern Recognition Letters. 2021; 146:31–37. https://doi.org/10.1016/j.patrec.2021.03.005

46. Li M, Liu B, Hu Y, Zhang L, Wang S. Deepfake Detection Using Robust Spatial and Temporal Features from Facial Landmarks. In: Proceedings—9th International Workshop on Biometrics and Forensics, IWBF 2021. Institute of Electrical and Electronics Engineers Inc.; 2021. p. 1–6.

47. Hu J, Liao X, Wang W, Qin Z. Detecting Compressed Deepfake Videos in Social Networks Using Frame-Temporality Two-Stream Convolutional Network. IEEE Transactions on Circuits and Systems for Video Technology. 2022; 32(3):1089–1102. https://doi.org/10.1109/TCSVT.2021.3074259