# Enhanced Preprocessing Stage For Feature Extraction of Deepfake Detection Based on Deep Learning Methods

2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS) | 979-8-3503-8306-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISAS60782.2023.10391672

Mohamed Abdulrahman Abdulhamid.[1,2]
*Department of Computer Science.*[1]
*University of Basra*
Basra, Iraq
mohammed@uobasrah.edu.iq

Asaad Noori Hashim.[2]
*Department of Computer Science.*[2]
*University of Kufa*
Najaf, Iraq
asaad.alshareefi@uokufa.edu.iq

***Abstract*—: Biometric identities are at risk of deepfake facial manipulation. Over the past few years, deepfake detection has been the subject of extensive research, most which centered on the application of machine learning strategies. However, the recognition of deepfake is still one of the most challenging problems in computer vision today. Deep learning has recently gained popularity due to its potential to solve a wide range of real-world problems, including detecting deepfakes. Our research focused on improving feature extraction methods by using deep learning and comparing preprocessing methods to show how they affect detection performance. A preprocessing approach was proposed to improve the regions of interest (ROIs) used to feed Moodle feature extraction. The proposed algorithmic approach involves finding one key frame extracted from a video by using the oriented fast and rotated brief algorithm, detecting the face oval region using the DLIB library as an ROI and performing quality improvement via contrast-limited adaptive histogram equalization (CLAHE)/adaptive histogram equalization (AHE) and comparing them. On the FaceForensics ++ dataset, CLAHE demonstrated superior performance compared with AHE with InceptionV3 as a feature extractor. The final classification result had an accuracy ratio of 89%, and the area under the curve was measured to be 77%.**

***Keywords—Preprocessing, Deepfake, Histogram equalization, Machine learning***

## I. INTRODUCTION

In recent years, fake videos and image content created by digital manipulation tools with advanced artificial intelligence (AI) have increased. Deepfakes are a mix of deep learning and fake content. People can be fooled into believing that a targeted individual said words that were actually said by another individual by swapping a human's face with that of another in a photo or video. Deepfake is the process of changing the face on a picture or video or swapping the faces of two people [1], [2]. With the rapid advancement of technology, extremely realistic images and videos can be easily created by replacing features, and manipulation detection has become difficult [3]. Cyberattacks using these technologies pose a threat not only to the privacy of individuals, but also to the security of nations. Moreover, deepfakes can be used to create celebrity porn movies, spread fake news, impersonate politicians and commit financial fraud [4], [5]. Given these reasons, deep forgery has become an increasingly prevalent threat, and highly effective methods must be established to distinguish fake content from real content, especially because social networking programs facilitate the quick spread of fake information. According to Reference [6], deep learning, machine learning and statistical models can be used to detect deepfake manipulation, as shown in Figure (1-a). Systematic steps, which are illustrated in Figure (1-b), must be followed to create a model for deep fake detection [5]. The current study focuses on one of the most important steps in developing a deepfake detection model, namely, the preprocessing step, and its effect on the final classification results [7]. Machine learning preprocessing is crucial to model performance and correctness. It cleans, organises and standardises data for analysis and modelling. Preprocessing addresses data issues, such as missing values, outliers and noise, which can hinder learning. Preprocessing guarantees that the machine learning algorithm obtains high-quality inputs and therefore lowers the chance of biased or misleading outcomes. Preprocessing scales, normalises and reduces dimensionality, thereby improving model convergence and efficiency [8]. Preprocessing increases the model's predictive capability and saves computational resources, resulting in improved generalisation and reliable machine learning insights [9].

The presence of bias in datasets utilised for AI and machine learning has long been a subject of concern. Research indicates that AI models trained on imbalanced datasets are likely to exhibit bias against specific demographic groups, resulting in suboptimal performance. The efficacy of deepfake detection methods that utilise AI models may be compromised at times because of the limited diversity in the training dataset. Therefore, efficient pretreatment processes that can overcome such problems must be considered [10].

This research seeks to detect deepfakes on the basis of deep learning and show the effects of data preprocessing on the proposed model's performance. The data preprocessing methods include key frame extraction, face detection, face ellipse extraction, quality enhancement and normalisation. A preprocessing algorithm is proposed to assist in the deepfake detection process of models based on advanced AI methods. To determine the accuracy difference, we compare the preprocessing methods by using the adaptive histogram equalisation (AHE) or contrast-limited adaptive histogram equalisation (CLAHE) algorithm.

Therefore, the contributions of this study can generally be summarised as follows: This study pursued two primary approaches. First, a method was proposed to extract a singular main frame from each video in the dataset, using ORB descriptors. Subsequently, the extracted frame was enhanced through the application of two widely employed techniques for quality improvement, namely AHE and CLAH. Therefore, a representative data set was curated for FF++, which is considered one of the pivotal video datasets in the realm of study of deep fake technology.

**979-8-3503-8306-5/23/$31.00 ©2023 IEEE**

The remainder of this paper is organised as follows. Section 2 presents a review of a few relevant studies. Section 3 provides the proposed preprocessing algorithm for feeding the feature extraction model. Section 4 presents a comparison and discussion of the extracted results. The findings are summarised in Section 5.

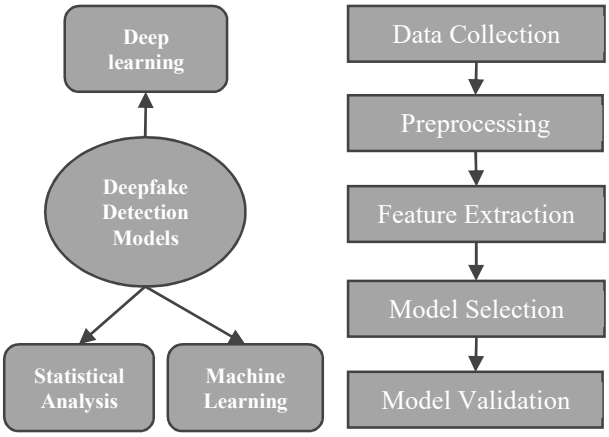



**Figure (1):** For (left) Types of Models of Deepfake Detection and (right) Deepfake Detection Steps.

## II. Related Work

Several approaches have been proposed for deepfake detection. According to Reference [11], fake images can be detected by looking for details, such as the colour of the eyes and missing details on the teeth, eyes and reflections. The study used logistic regression (LR) and multilayer perceptron (MLP) models to distinguish the fake faces. LR and MLP methods were tested on a FaceForensics database, where LR obtained 86.6% accuracy and MLP obtained 82.3% accuracy.

In Reference [12], three hypotheses were proposed to explain how deepfake detection models learn artifact features from binary labels alone. The first one was that through the use of deepfake detection models, fake images can be detected based on visual concepts that are not relevant to their sources or targets (i.e. relevant to their artifacts). The second hypothesis was that through FST-Matching, deepfake detection models can learn artifact-relevant visual concepts implicitly from the training set. The experiments revealed that FST-Matching implicitly teaches artifact visual concepts susceptible to video compression in the raw training set. The third hypothesis was that forgery detection performance on compressed videos can be enhanced by the FST-Matching deepfake detection model. Test results obtained from the FaceForensics++ (FF++) dataset showed 81.33% accuracy and 77.01% the area under the curve (AUC).

An anticompression facial forgery detection framework was proposed in Reference [13] to overcome performance degradation in the identification of compressed images and videos. From the original and compressed forgeries, the method extracts compression-insensitive features by using an adversarial learning strategy, robust partitioning and an attention-transfer module. Compressed and uncompressed images can be handled efficiently by this method according to the experimental results. According to the test results, the model is 80.03% accurate, and its AUC is close to 77.71%.

The framework proposed in Reference [14] employs temporal modelling of precise geometric features, a module for calibration and a two-stream recurrent neural network. It is lightweight, simple to train and robust in detecting videos that have been highly compressed or corrupted by noise. The model has an AUC of 0.999 on the FF++ dataset, but its performance in compressed videos decreases. However, if the trained weights are directly used for evaluation, accuracy and AUC will be 55.22% and 67.82%, respectively.

## III. Proposed Method

This section presents the algorithm proposed for the preprocessing step in the processing an FF++ video dataset that is used for training and testing purposes. Figure (2) presents a flowchart of the proposed algorithm and an explanation of each individual step.

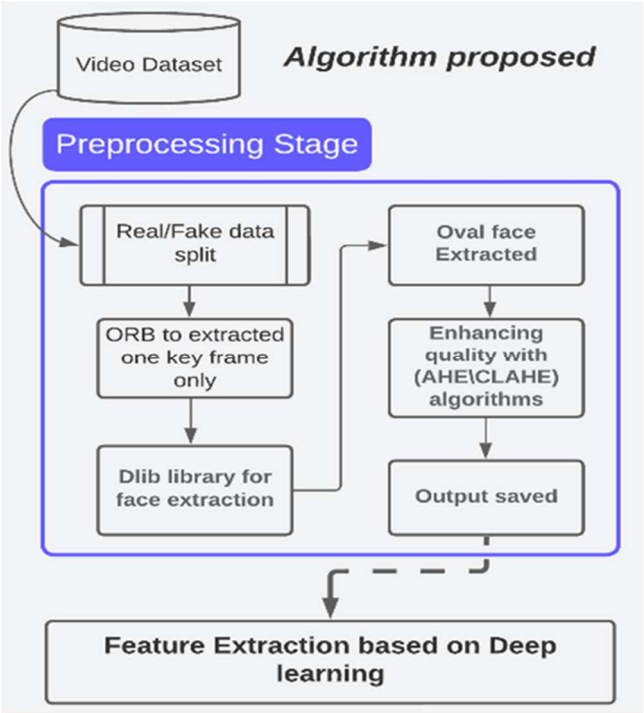



**Figure (2):** Proposed Preprocessing Stage.

### *3.1. FF++ Dataset [15]*

The FF++ dataset is a crucial tool for scholars and practitioners in the domain of computer vision and image forensics. The dataset consists of a large number of manipulated facial videos created through the use of advanced machine learning techniques, such as generative adversarial networks (GANs) [16]. The FF++ tool offers a wide array of facial manipulations, such as deepfakes, making it a valuable resource for examining the growing risk posed by synthetic media and devising effective methods for detecting and verifying their authenticity. This dataset was used to train and validate the proposed model. Approximately 800 training-related videos and more than 300 examination-related videos were utilised.

### *3.2. Key Frame Extracted by the Oriented Fast and Rotated Brief Algorithm [17]*

The oriented fast and rotated brief (ORB) binary descriptor is a useful mathematical technique used in computer vision to identify and match objects in images. It works by breaking down an image into small parts called keypoints and

extracting binary values for each keypoint. These binary values help in understanding the unique features and characteristics of the object being recognised. With the help of the ORB binary descriptor, computers can analyse images and determine if they contain the same objects or not. Therefore, it can be used to identify the most important frame amongst all the frames extracted from the video clip after comparing them and selecting the best one. From a mathematical perspective, the approach incorporates two fundamental concepts: the FAST detector and the BRIEF descriptor. The FAST detector determines the orientation of keypoints on the basis of the intensity centroid [18]. BRIEF describes an image patch by using binary intensity tests [6], resulting in concise bit string representation. Consider a smoothed image patch denoted as p. The binary test theta is mathematically defined as

$$\tau(\boldsymbol{p};\boldsymbol{x},\boldsymbol{y}) := \begin{cases} 1 & : \boldsymbol{p}(x) < \boldsymbol{p}(\boldsymbol{y}) \\ 0 & : \boldsymbol{p}(x) \geq \boldsymbol{p}(\boldsymbol{y}) \end{cases}, \qquad (1)$$

where p(x) refers to $p$'s intensity at a particular point $x$. A vector of n binary tests can be described as the feature.

$$f_n(\boldsymbol{p}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\boldsymbol{p};\boldsymbol{x}_i,\boldsymbol{y}_i) \qquad (2)$$

Overall, the selection of the key frame to represent a video can be determined by utilising the analysis of variance and correlation of orientated BRIEF features based on ORB. By considering these crucial characteristics, the frame with the highest number of matching pixels among all the extracted frames can be identified as the winner and chosen as the representative key frame.

### *3.3. Face Extraction Based on the DLIB Library [19]*

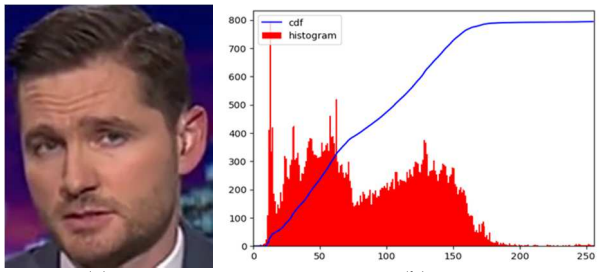
The DLIB library is an exceptional open-source machine learning library that occupies a prominent position in the forefront of deep learning research. This tool, which was developed by Davis King, offers a diverse set of algorithms and functions that allow individuals in academic and professional settings to effectively utilise deep learning techniques for various applications. The techniques are employed to extract the facial region, which is commonly referred to as the region of interest (ROI). Subsequently, the necessary configurations are established to exclusively extract the facial disc.

### *3.4. Enhance the Quality of ROIs [20]*

Histogram equalisation improves image quality by redistributing pixel intensities. It increases contrast and dynamic range by mapping an image to a desired histogram, transforming pixels for a uniform distribution and enhancing visual clarity and details. This study employed two variants of histogram equalisation, namely, AHE and CLAHE, to enhance image quality. A comparative analysis of the two methods was conducted, which will be elaborated in Section 4. Each method is explained in the following subsections.

#### *3.4.1. Adaptive Histogram Equalisation (AHE) Algorithm [21]*

AHE is a popular image enhancement technique that reshapes pixel intensities to increase contrast. It is mathematically represented by dividing an image into regions and computing local histograms. A transformation function that extends the intensity values inside each zone adjusts these histograms. The AHE transformation function is customised for the local region to increase contrast enhancement. However, AHE exhibits a proclivity to excessively increase noise in portions of a picture that are somewhat uniform in nature. Contrast-restricted adaptive histogram equalisation is a modified version of adaptive histogram equalisation that effectively addresses the issue of amplification by imposing limitations. In Figure 3, we will see the OpenCV function to implement AHE applied to a face sample extracted from the FF++ dataset.




(a) (b)

**Figure (3):** Example of Adaptive Histogram Equalization.

#### *3.4.2. Contrast-Limited Adaptive Histogram Equalisation (CLAHE) Algorithm [21]*

CLAHE is a common method to improve image quality. It computes the cumulative distribution histogram for each tile and restricts contrast within a range by dividing the image into tiles. It prevents over-enhancement and produces a natural-looking image. CLAHE's adaptive enhancement parameters react to the image's local features for accurate results. Figure 4 illustrates the facial processing that has been extracted based on use CLAHE method.

(a) (b)

**Figure (4):** CLAHE Example (a) Original Face (b) face after CLAHE enhancement.

The image is divided into 8x8 "tiles," which OpenCV defaults to. Each block then gets histogram equalisation as usual. In a confined spatial area, the histogram would have a narrow distribution without noise. Noise is amplified if detected. To fix this, contrast limiting is used. When a histogram bin exceeds the contrast limit (default: 40), the pixels are clipped and evenly dispersed over the remaining bins before histogram equalisation in the OpenCV library. After equalisation, bilinear interpolation removes tile border artefacts. Two crucial parameters—tile count and clip limit—impact Contrast Limited Adaptive Histogram Equalisation

(CLAHE) performance. The number of tiles indicates the image's tile count. This parameter has two values, m and n. Thus, the image is divided into m x n local portions. The second option, clip limit, controls image noise amplification. Clip limit restricts tile histogram intensity. These parameters can affect image quality and noise if set wrong. [22] .

### *3.5. Feature Extraction by Inception v3 [23]*

Inception v3 is widely recognised in the field of deep learning because of its strong performance in tasks related to image recognition. This sophisticated model developed by Google showcases the remarkable capacity of AI in terms of comprehension and adaptability. Inception v3 demonstrates the remarkable advancements and future prospects of machine learning technologies because it has the capability to accurately capture intricate patterns and effectively classify images across a diverse array of categories. To achieve the intended objective, this study used the feature extraction capability of the system, and the upper layers of the convolutional neural network (CNN) were utilised in the subsequent evaluation phase. The hyperparameters used to fit the model are shown in Table (2). Figure (5) shows in a simplified manner the structure of the networks by Inception v3.

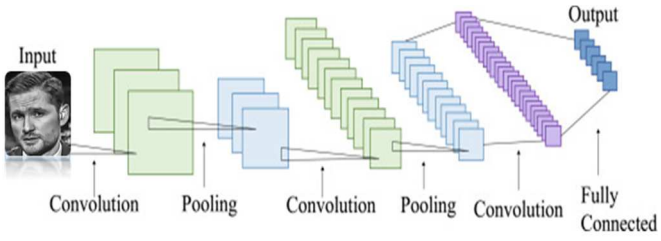



**Figure (5):** The Architecture of Inception v3 [24].

### *3.6. Feature Extraction by Custom CNN*

In this study, we suggest deep neural networks with CNN features for feature extraction and fully conected layers for decision-making. We estimate the training error using binary cross-entropy as the loss function. In this section, we examine the design of a CNN that was proposed for feature extraction. The structure was created on the basis of a straightforward design to assess the compatibility between the proposed preprocessing technique and other models. The specific architecture and components of the custom CNN are given in Table 1. Table 2 shows the hyperparameters employed throughout the training phase.

TABLE I. PROPOSED CNN FOR FEATURE EXTRACTION.

| Layer name | Output | Activation function |
|---|---|---|
| Input Layer | (150,150,3) | - |
| Conv_0 (2D) | (148,148,32) | Relu |
| Pooling_0 (Max pooling) | (74,74,32) | |
| Conv_1 (2D) | (72,72,32) | |
| Pooling_1 (Max pooling) | (36,36,32) | |
| Conv_2 (2D) | (34, 34, 32) | |
| Pooling_2 (Max pooling) | (17, 17, 32) | |
| Conv_3 (2D) | (15, 15, 64) | |
| Pooling_3 (Max pooling) | (7, 7, 64) | |
| Conv_4 (2D) | (5, 5, 64) | |
| Pooling_4 (Max pooling) | (2, 2, 64) | |
| Flatten | 256 | - |
| Dense_1 | 64 | Relu |
| Dropout | 64 | - |
| Dense_2 | 1 | Sigmoid |

TABLE II. HYPERPARAMETER VALUES USED FOR THE TRAINING STAGE.

| Hyperparameter Type | Value |
|---|---|
| Optimiser | Root Mean-squared Propagation (Learning Rate = 0.0001) |
| Loss | Binary Cross Entropy |
| Metrics | Accuracy |
| Batch size | 10 |
| Verbose | 1 |
| Epochs | 300 |

## IV. EXPERIMENTAL RESULTS

Using the previously described FF++ dataset, we performed experimental comparisons to assess the classification and computing performance of the proposed approach. The computational tasks were executed on the Google Colab platform, using the T4 type of graphics processing unit. The programming language used was Python 3, with the Keras module of the TensorFlow library serving as the backend. Two evaluation criteria, namely AUC and accuracy (Acc), were used to assess the proposed model; both are widely regarded as optimal measures for evaluating classification models [5]. One metric is used to provide an overall measure of success across all possible classification thresholds and to assess the level of generality exhibited by the suggested model, and the other metric is used to evaluate the performance of the model. Equation (3) shows mathematical representations that demonstrate the Acc metric.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \qquad (3)$$

Where $TP$: $True\ positive$, $TN$: True negative, $FP$: False positiveand $FN$: False Negative.

Figure (5) shows some of the results of the face extraction and the optimisation process that was performed for each ROI extracted. In the figure, each item is represented by two samples, each of which has been applied one of the quality improvement techniques (AHE or CLAHE).


**Figure (6):** Region of Interest Based on the Proposed Algorithm on the FF++ Dataset: (A) CLAHE and (B) AHE algorithms.

Table 3 shows a summary of the results when the proposed models were tested in the processing algorithm using the AHE and CLAHE methods to improve image quality. Based on the data presented in the table, it can be inferred that there is a discernible impact on the ultimate accuracy result and (), but with a

relatively minor variation. It is observed that the area under the curve (AUC) derived from the contrast limited adaptive histogram equalisation (CLAHE) method yielded a value of 77%, with the clip limit parameter set at 2.0.

TABLE III. SUMMARY OF FEATURE EXTRACTION RESULTS.

| Model Test | Type of Quality Enhancement | Acc | AUC |
|---|---|---|---|
| Custom CNN | AHE | 68% | 71% |
| Custom CNN | CLAHE | 71% | 76% |
| Inception v3 | AHE | 84% | 74% |
| Inception v3 | CLAHE | 89% | 77% |

The receiver operating characteristic (ROC) for each proposed model is depicted in Figure 7.

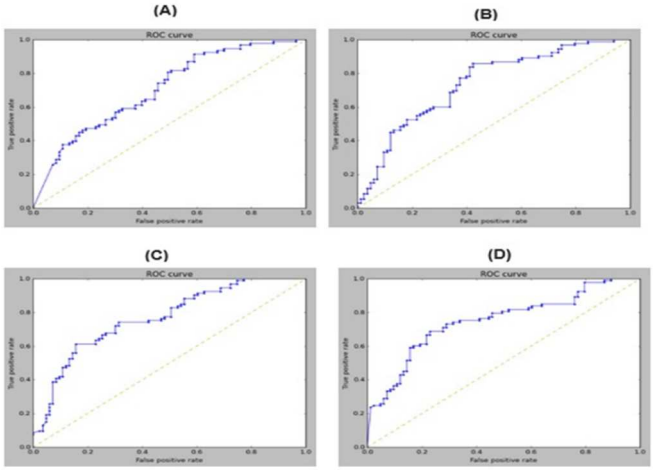



**Figure (7):** ROC of the Proposed Model. (A) Custom CNN by AHE, (B) Custom CNN by CLAHE, (C) Inception v3 by AHE and (D) Inception v3 by CLAHE.

The comparison aforementioned revealed that the CLAHE approach employed to enhance the quality of the key frame retrieved by ORB exhibited superiority over the AHE method in both of the models presented. For the reasons mentioned above, the model with the highest percentages was compared to that in another relevant study. A comparative analysis was conducted on relevant literature, as depicted in Table 4. The FF++ dataset served as the basis for all of the comparison methods used.

TABLE IV. COMPARISON WITH A RELATED STUDY

| Model | Acc | AUC |
|---|---|---|
| [1] | 86.6% and 82.3% | - |
| [2] | 81.33% | 77.01% |
| [3] | 80.03% | 77.71% |
| [4] | 55.22% | 67.82% |
| InceptionV3 – CLAHE (Proposed) | 89% | 77% |

The comparison revealed the superiority of the proposed preprocessing algorithm in terms of accuracy and convergence ratio (AUC), despite the focus being only on extracting and optimising an ROI (the face disc). These results suggest that improving the construction of a highly efficient feature extractor in the future can improve the accuracy results of the entire model.

The **limitations** of the suggested study were evident in its restricted scope, since it focused solely on evaluating the proposed algorithmic approach using a single dataset. This study, on the other hand, mostly uses a comparison method to find out how different pre-processing methods affect the final classification's accuracy, especially in tricky situations like deep bleeding detection. In the realm of visual media.

## V. CONCLUSION

Pre-processing samples before feeding them to deep learning networks can improve their performance. In this study, we compared the effects of two histogram equalisation methods on the quality of the key frame retrieved by the ORB algorithm. Two types of CNNs were used; one was proposed in this research (custom CNN), and the other was based on a pre-trained neural network (Inception v3). The CLAHE method was superior to the AHE method in terms of enhancing the quality of the extracted ROI. For the Inception v3 network that was improved by the CLAHE algorithm by using data from the FF++ dataset, the test result had an accuracy of 89% and its AUC was 77%. Therefore, CLAHE is better for intricate feature extraction tasks, especially in deepfake frames with low contrast, because it equalises the histogram and maximises entropy. In the future, researchers must improve preprocessing approaches to improve deepfake classification by using advanced AI. Our system can be tested on other datasets.

## REFERENCES


[1] X. Li, R. Ni, P. Yang, Z. Fu, and Y. Zhao, "Artifacts-disentangled adversarial learning for deepfake detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1658–1670, 2022.

[2] S. Das, S. Seferbekov, A. Datta, M. S. Islam, and M. R. Amin, "Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3776–3785.

[3] L. Stroebel, M. Llewellyn, T. Hartley, T. S. Ip, and M. Ahmed, "A systematic literature review on the effectiveness of deepfake detection techniques," *Journal of Cyber Security Technology*, vol. 7, no. 2, pp. 83–113, 2023.

[4] T. T. Nguyen *et al.*, "Deep learning for deepfakes creation and detection: A survey," *Computer Vision and Image Understanding*, vol. 223, p. 103525, 2022.

[5] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, "Deepfake detection: A systematic literature review," *IEEE access*, vol. 10, pp. 25494–25513, 2022.

[6] H. F. Shahzad, F. Rustam, E. S. Flores, J. Luís Vidal Mazón, I. de la Torre Diez, and I. Ashraf, "A Review of Image Processing Techniques for Deepfakes," *Sensors*, vol. 22, no. 12, p. 4556, 2022.

[7] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[8] A. Orooji and F. Kermani, "Machine learning based methods for handling imbalanced data in hepatitis diagnosis," *Frontiers in Health Informatics*, vol. 10, no. 1, p. 57, 2021.

[9] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *Journal of Engineering and Applied Sciences*, vol. 12, no. 16, pp. 4102–4107, 2017.

[10] P. Kwon, J. You, G. Nam, S. Park, and G. Chae, "Kodf: A large-scale korean deepfake detection dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10744–10753.

[11] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, IEEE, 2019, pp. 83–92.

[12] S. Dong, J. Wang, J. Liang, H. Fan, and R. Ji, "Explaining deepfake detection by analysing image matching," in *European Conference on Computer Vision*, Springer, 2022, pp. 18–35.

[13] S. Cao, Q. Zou, X. Mao, D. Ye, and Z. Wang, "Metric learning for anti-compression facial forgery detection," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1929–1937.

[14] Z. Sun, Y. Han, Z. Hua, N. Ruan, and W. Jia, "Improving the efficiency and robustness of deepfakes detection through precise geometric features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3609–3618.

[15] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.

[16] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process Mag*, vol. 35, no. 1, pp. 53–65, 2018.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[18] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[19] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[20] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," in *2014 international conference on advances in computing, communications and informatics (ICACCI)*, IEEE, 2014, pp. 2392–2397.

[21] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," in *2014 international conference on advances in computing, communications and informatics (ICACCI)*, IEEE, 2014, pp. 2392–2397.

[22] U. Kuran and E. C. Kuran, "Parameter selection for CLAHE using multi-objective cuckoo search algorithm for image contrast enhancement," *Intelligent Systems with Applications*, vol. 12, p. 200051, 2021, doi: https://doi.org/10.1016/j.iswa.2021.200051.

[23] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd international conference on image, vision and computing (ICIVC)*, IEEE, 2017, pp. 783–787.

[24] N. Dong, L. Zhao, C. H. Wu, and J. F. Chang, "Inception v3 based cervical cell classification combined with artificially extracted features," *Appl Soft Comput*, vol. 93, p. 106311, 2020, doi: https://doi.org/10.1016/j.asoc.2020.106311.