Journal of Robotics and Control (JRC)
Volume 5, Issue 5, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i5.22473

Robust DeepFake Face Detection Leveraging Xception Model and Novel Snake Optimization Technique

Ahmed SAAD Al-Qazzaz [1*], Pedram Salehpour [2], Hadi S. Aghdasi [3]
[1, 2, 3] Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz Iran
Email: [1] Msc.ahmedalqazaz@gmail.com, [2] psalehpoor@tabrizu.ac.ir, [3] aghdasi@tabrizu.ac.ir
*Corresponding Author

***Abstract*—DeepFake technology has created an existential crisis around authenticity in digital media with the ability to create nearly imperceptible forgeries on a massive scale, such as impersonating public figures for nefarious reasons like misinformation campaigns, harassment, and fraud. In this thesis, a model Xception is combined with the Snake optimization technique to ensure efficient and accurate detection of ADOR in practice. The former is deep CNN architecture Xception which exploits depthwise separable convolutions to perform efficient feature extraction, and the latter is a novel snake optimization that borrows lessons from real-life predatory snakes to dynamically adapt parameters for better exploration of search space while avoiding local optima. The combined modality is systematically evaluated using multiple challenging DeepFake video datasets and shows significant improvement. A comparison of performance with other methods showed that a mean accuracy, precision, recall, and F1-score was 98.53% for the Snake-optimized Xception model while outperformed some state-of-the-art approaches and traditional Xception itself. This helps in reducing missing of misdetection and reduction of false positives, helping achieve a tool that is highly effective for digital media forensics. Such discoveries open the door for this method to unlock new levels of digital content integrity, necessary in media verification and legal evidence authentication, as well as assist individuals dealing with fake news or videos attempting identity theft online. This research highlights the strong efficacy of coupling the Xception model with Snake optimization for DeepFake detection; thus, establishes a new state-of-the-art and will inspire future studies and applications to protect genuineness in digital media.**

Keywords—DeepFake Face Detection; Xception Model; Snake Optimization; Digital Manipulation; Deep Learning; Media Forensics; Video Authentication.

I. INTRODUCTION

While there are different ways to identify DeepFakes, the widespread solutions have limitations around their reliability and accuracy. This paper opened the door to utilize Deep Learning in detecting fake videos, but there is still a research gap as no detection model has been found that works across all cases of DeepFakes. This paper fills this gap by proposing a novel detection solution implemented with the Xception model [5], then using Jungles learning techniques and in particular, Snake algorithm [6].

Xception model leverages depthwise separable convolutions [7] whose computations efficiently capture features from the individual video frames spanning both low-level and high-level semantics across all modalities. The work presented in [8] uses a variant of snake optimization method, which is an algorithm heuristically inspired by the movement characteristic snakes have and consequently named so [9]. The method adaptive tunes an optimization strategy to the exact form of the solution landscape, which can improve tuning in complicated parameter spaces such as those found for Xception [10]. They are more sensitive for each pixel to identify small differences within video materials and seamlessly integrated the Snake optimization principle, outperforming most existing models [11].

To address these challenges, we present a new object detection architecture which is based on the Xception model and Snake optimization. Slither extends the training steps to improve model's sensitivity in subtle video manipulations [11]. We perform extensive validation via experiments on a variety of highly public DeepFake videos datasets to show that our approach seamlessly integrates within the expansive, fast-growing literature in this space is robust and generalizable [12][13].

This paper makes three contributions to research. There are actually many firsts in this work; for one, it describes a new state-of-the-art approach to DeepFake detection by integrating an Xception model with Snake optimization [14]. The architecture not only increases the accuracy of detection but also makes subtle changes more sensitive to the model. Second, this research carries out extensive experimental on various datasets to thoroughly validate the proposed approach as a standard practical regarding integrity and authenticity preserving of digital media [16]. Together, these are two major steps forward for the field of digital forensics and FAKE detection tech in the battle against fake content proliferation to re-establish trust on digital media [17][18].

II. RELATED WORK

This paper [19] presented a study of detecting fake faces generated by GANs utilizing a new architecture called LBP-Net. LBP-Net has an advantage of using Local Binary Patterns that enable it to capture the textural disparity between fake and real faces, resulting in more robustness against the prevailing image augmentation techniques. The results showed LBP-Net gave accuracy and precision much higher than those of ResNet18 and Gram-Net. For example, with 10% blurred images, the accuracy of LBP-Net accuracy is 1.11% and 1.12% better compared to Gram-Net and

ResNet18, respectively. Concerning brightness-changed images, LBP-Net images increases the accuracy 3.9% and 8.37% and precision 12.17% 22.97% better in comparison to single models. In the end, the best results were achieved using a combination of different models rather than a single model.

In the paper [20], an investigation into deepfakes, a trending advanced digital manipulation method that employs deep learning to produce misleading images and videos, is carried out. Identifying deepfake images is a challenge to discern between the original content and the manipulated content, which is being used more regularly. It is becoming essential to be able to distinguish between authentic and manipulated videos because of the increasing proliferation of deepfakes. A study is conducted to test and experiment with different identifiers within and among fake and real images and videos. InceptionNet, a Convolutional Neural Network algorithm, is used as an identifier to identify deepfakes. Different convolutional networks were employed in the comparative study. The dataset utilized in this research is sourced from Kaggle and contains 401 training videos and 3,745 images generated through an augmentation process. Validation is done on test data, and the evaluation metrics include accuracy and the confusion matrix are reviewed. The obtained results revealed that the suggested model recognizes the deepfake images and videos in an exceptionally compelling manner with a 93% accuracy rate.

This paper [21] addresses the issue of deepfakes, which are realistic-looking fake media created by deep-learning algorithms that process large datasets to learn how to manipulate video and digital content, such as swapping faces or objects. The proliferation of deepfake content and modification technologies is significantly impacting public discourse and human rights protection. Deepfakes are increasingly being used maliciously as misinformation in legal contexts, aiming to influence court decisions. Given the critical role of digital evidence in legal cases, detecting deepfake media is vital and highly sought after in digital forensics. This study focuses on identifying and building a classifier capable of accurately distinguishing between authentic and manipulated media, with particular emphasis on facial recognition systems for identity protection. The research compares several state-of-the-art face-detection classifiers, including Custom CNN, VGG19, and DenseNet-121, using an augmented dataset of real and fake faces. Data augmentation techniques are employed to enhance performance and reduce computational demands. Preliminary results show that VGG19 outperforms the other models, achieving the highest accuracy of 95%.

This paper [22] explores the concept of deepfake technology, where digital manipulation techniques are employed to create fake images and videos or to swap faces using deep learning algorithms. Deepfake pornography, a particularly harmful application, contributes to hoaxes, fake news, and financial fraud. Deep learning, especially with the advancement of Generative Adversarial Networks (GANs), has made deepfakes more prevalent on social media, posing significant threats to the public. The necessity to detect these forged images and videos has led to extensive research; however, many existing methods are inefficient against new threats and have high computational demands. This study addresses these challenges by introducing a novel approach using Fuzzy Fisher Face with Capsule Dual Graph (FFF-CDG) to detect various types of fake images and videos. The datasets utilized in this research include FFHQ, 100K-Faces, DFFD, VGG-Face2, and Wild Deepfake. The proposed method demonstrates superior performance, achieving an accuracy of 95.82% on the FFHQ dataset, compared to 81.5%, 89.32%, and 91.35% obtained by existing systems.

The work [23] studies new techniques and methods of digital image and video processing – deepfake, which are based on deep learning technologies and create opportunities for disseminating true-to-life but actually false information. In some cases, the use of deepfake technologies can not be recognized by human properties and can only be established by comparing the images produced in the deepfake. At the same time, with the predictably rapid emergence of digital images and videos obtained by the use of deepfake technologies, it is assumed that the possibilities of identifying content obtained using them will be of paramount importance. The study describes and experimentally investigates various methods for identifying deepfakes in images and videos created using deepfake. For this convolutional neural network algorithm, InceptionNet was taken. A comparison of the application of different convolution networks was made. The basis of the research data is the Kaggle database which is represented by 401 training videos and 3745 augmented images. The research results are summarized in terms of accuracy and confusion matrix. The results of the proposed model on the dataset of this work showed the optimal result in identifying deepfake images in videos of 93%.

Detection technology of deep fake faces in videos is widely used in many areas of the virtual world and finance, to combat the spread of fake information. This paper [24] introduces a deep fake face detection in video method is a Bidirectional-LSTM approach based on temporal features. The Bidirectional-LSTM method is a deep learning approach that takes full advantage of temporal information among videos. It emphasizes the continuity features of facial expressions and muscle movements combined with the individual's talking style that can be recorded. The continuity features are cracked in deepfake videos and can be identified. The suggested method was substantially more successful than previously-used strategies, such as SVM and BP neural networks, which resulted in a shorter training time and improved performance. In practice, the accuracy of this method is 82.65%, according to our experimental data on the DFDC dataset. Another advantage of the Bidirectional-LSTM-based approach is that it can recognize deepfake videos of compressed quality and poor noise immunity.

The growing number of deepfake videos, including those made using face-swapping techniques, has given rise to the spread of dangerously authentic forged videos that threaten individuals and even whole nations. This paper [25] presents a new method for identifying true videos among deepfake videos based on the urgent problem associated with this kind of videos. The method is called YOLO-CNN-XGBoost and uses the YOLO face detector to select the face and screen area in each frame of a video sequence. In this case, each of the extracted faces is passed to InceptionResNetV2 CNN for

feature extraction. Next, XGBoost, acting as a recognizer, is activated at the top of the network. The proposed method demonstrates impressive performance on the CelebDF-FaceForensics++ (c23) merged dataset, achieving an AUC of 90.62%, accuracy of 90.73%, specificity of 93.53%, sensitivity of 85.39%, recall of 85.39%, precision of 87.36%, and F1-measure of 86.36%. The experimental results validate the superiority of the YOLO-CNN-XGBoost method compared to existing state-of-the-art techniques.

The ease of multimedia data manipulation and forgery has significantly increased with the advent of Artificial Intelligence (AI), leading to the widespread creation of AI-generated fake content known as deepfakes. This surge in deepfake content has presented new issues, concerns, and challenges for the research community, particularly in the area of detection. While deepfake detection has become a widely addressed task, existing methods often struggle with generalization. This paper [26] details the Face Deepfake Detection and Reconstruction Challenge, which tasked participants with two objectives: developing a deepfake detector capable of operating effectively "in the wild," and devising a method to reconstruct original images from deepfakes. The competition utilized real images from CelebA and FFHQ, along with deepfake images generated by StarGAN, StarGAN-v2, StyleGAN, StyleGAN2, AttGAN, and GDWCT. The winning teams were judged based on the highest classification accuracy for Task I and the "minimum average distance to Manhattan" for Task II. In Task I, deep learning algorithms, particularly those based on the EfficientNet architecture, achieved the best results, with VisionLabs securing first place with an accuracy of 93.61%. However, no winners were declared for Task II. The paper includes a detailed discussion of the methods proposed by the teams and their corresponding rankings.

Deep learning is a powerful technique widely applied in natural language processing, computer vision, image processing, and machine vision. Deepfakes, generated using deep learning methods like Generative Adversarial Networks (GANs), create synthetic images that are often indistinguishable from real ones, posing significant threats to the public. Detecting deepfake image content is crucial, but existing techniques often suffer from inaccuracy and high computational demands. This paper [27] presents a novel approach to deepfake detection using the Fisherface algorithm combined with Local Binary Pattern Histogram (FF-LBPH) and Deep Belief Networks (DBN) with Restricted Boltzmann Machines (RBM). The Fisherface algorithm reduces dimensionality in the face space, while LBPH aids in face recognition. The proposed FF-LBPH DBN model was tested on public datasets such as FFHQ, 100K-Faces, DFFD, and CASIA-WebFace. The approach includes pre-processing with a Kalman filter for refined fake image detection. The fusion of Fisherface and LBPH algorithms significantly reduced execution time and improved detection accuracy, achieving a rate of 98.82% on the CASIA-WebFace dataset and 97.82% on the DFFD dataset. The results indicate that the FF-LBPH DBN model effectively detects deepfake face images, potentially preventing defamation. Future work may explore various classifiers and distance metric measures to enhance deepfake detection.

This paper [28] introduces an adaptive manipulation traces extraction network (AMTEN), which acts as a pre-processing step to suppress image content and emphasize manipulation traces. AMTEN utilizes an adaptive convolution layer to predict and enhance manipulation traces, updating weights during the back-propagation pass to maximize detection. By integrating AMTEN with CNN, the paper presents AMTENnet, a highly effective fake face detector. Experimental results demonstrate that AMTENnet achieves superior pre-processing and detection capabilities, attaining an average accuracy of 98.52% for fake face images generated by various FIM techniques, and 95.17% for images with unknown post-processing operations, outperforming existing state-of-the-art methods.

This paper [29] introduces an attention-based DeepFake detection (ADD) method that leverages the fine-grained and spatial locality attributes of artificially synthesized videos for improved detection. The ADD framework consists of two main components: face close-up and face shut-off data augmentation methods, which can be applied to any convolutional neural network (CNN)-based classifier. ADD identifies potentially manipulated areas in the input image to extract representative features and emphasizes these regions during the decision-making process. The effectiveness of ADD is evaluated on two challenging DeepFake forensics datasets, Celeb-DF (V2) and WildDeepFake. The study demonstrates the generalizability of ADD by testing it with four popular classifiers: VGGNet, ResNet, Xception, and MobileNet. Results show that ADD significantly enhances the detection performance of all four classifiers on both datasets. Notably, ADD with a ResNet backbone detects DeepFakes with an accuracy of over 98.3% on Celeb-DF (V2), surpassing state-of-the-art DeepFake detection methods.

In Table I The comparison of deepfake detection methods with related works in different methodology, key features, dataset and results.

Paper [19] uses Local Binary Patterns to encode texture differences in fake faces through LBP-Net, and achieves the highest accuracy over ResNet18 as well as Gram-Net with higher precision. In contrast, Paper uses InceptionNet i.e. CNN and it give 93% accuracy on Kaggle dataset so the emphasis is done on comparisons with other DNNs Its proposed work, Facial recognition with custom CNN and VGG19, DenseNet-121 architectures on augmented dataset - Paper [21] had a little different focus but they were working more closely to the subject I am interested in; their highest accuracy was 95% utilizing VGG19 architecture (obtained from base import Inference) and lastly legal aspect of deepfakes. Paper [22] introduces FFF-CDG which use Fuzzy Fisher Face and Capsule Dual Graph, has 95.82% accuracy on FFHQ dataset solving drawbacks of current methods.

TABLE I. RELATED WORK

Paper	Methodology	Key Features	Dataset	Results	Highlights
[19]	LBP-Net	Leverages Local Binary Patterns (LBP) to capture texture differences between fake and real faces	Various (not specified)	LBP-Net outperforms ResNet18 and Gram-Net with higher accuracy and precision	Ensemble models further enhance detection performance
[20]	InceptionNet	Uses CNN (InceptionNet) for identifying deepfakes	Kaggle (401 training videos, 3,745 images)	Accuracy: 93%	Conducts comparative analysis with different CNNs
[21]	Custom CNN, VGG19, DenseNet-121	Focuses on facial recognition for deepfake detection	Augmented dataset of real and fake faces	VGG19 achieves highest accuracy: 95%	Addresses deepfakes in legal contexts
[22]	FFF-CDG	Uses Fuzzy Fisher Face with Capsule Dual Graph	FFHQ, 100K-Faces, DFFD, VGG-Face2, Wild Deepfake	Accuracy: 95.82% on FFHQ	Addresses inefficiencies of existing methods
[23]	InceptionNet	Uses CNN (InceptionNet) for deepfake detection	Kaggle (401 training videos, 3,745 images)	Accuracy: 93%	Focuses on distinguishing original from manipulated content
[24]	Bidirectional-LSTM	Exploits temporal features for deepfake detection in videos	DFDC dataset	Accuracy: 82.65%	Robust against compressed videos with noise interference
[25]	YOLO-CNN-XGBoost	YOLO for face detection, InceptionResNetV2 for feature extraction, XGBoost for recognition	CelebDF-FaceForensics++ (c23)	AUC: 90.62%, Accuracy: 90.73%	Superior performance compared to state-of-the-art techniques
[26]	EfficientNet	Face Deepfake Detection and Reconstruction Challenge	CelebA, FFHQ, deepfake images from various GANs	VisionLabs Accuracy: 93.61%	Tasks: detection in “wild” scenarios, reconstructing original images
[27]	FF-LBPH, DBN	Combines Fisherface with Local Binary Pattern Histogram (LBPH) and Deep Belief Networks (DBN)	FFHQ, 100K-Faces, DFFD, CASIA-WebFace	Accuracy: 98.82% on CASIA-WebFace, 97.82% on DFFD	Addresses high computational demands and inaccuracy
[28]	AMTENnet	Adaptive manipulation traces extraction network (AMTEN) integrated with CNN	Various FIM techniques	Accuracy: 98.52% (general), 95.17% (post-processed images)	Superior pre-processing and detection capabilities
[29]	ADD (Attention-based DeepFake Detection)	Utilizes face close-up and shut-off data augmentation methods	Celeb-DF (V2), WildDeepFake	Accuracy: >98.3% with ResNet on Celeb-DF (V2)	Enhances detection performance of multiple classifiers

Paper [23] leverages an InceptionNet on the Kaggle dataset used in Paper [20] and gains 93 % of accuracy also discriminating between If a photo is original or manipulated. Using Bidirectional-LSTM to dig temporal features for video deepfake detection, Paper [24] achieves an accuracy of 82.65% on the DFDC dataset and works well with compressed videos under noise interference. Paper [25] utilizes YOLO for detection of face, InceptionResNetV2 to extract features and XGBoost for recognition finally achieve 90.62% AUC over the same CelebDF-FaceForensics++ dataset which are better than other methodologies.

In [26] authors have used EfficientNet in a contest which resulted into an accuracy of 93.61% utilizing diverse datasets adhering to wild scenarios and image reconstruction task. The work [27] integrates three methods (i) Fisherface with LBPH, and DBN which provides marginal accuracy of 98.82% in CASIA-WebFace; however, it is unable to reduce computational demands and the error during classification process In [28] a framework AMTENnet has been proposed that combines the operation of Gaussian basic image processing operators and CNNs, which in general can achieve 98.52% accuracy with test images without additional pre-processing and on post-processed images it is 95.17%. In contrast, Paper[29] introduces ADD based on face close-up data augmentation and shut-off strategy, resulting in more than 98.3% accuracy rate with ResNet over Celeb-DF(V2), significantly improving the detection performance under classifiers The table overall illustrates a wide spectrum of deepfake detection approaches with different tradeoffs and datasets, showcasing progress in terms of both accuracy, robustness as well specific testing scenarios -yi elding insights into the variety of strategies developed by researchers to address the difficult problem of spotting deepfakes.

III. PROPOSED APPROACH

As shown in Fig. 1, this proposed novel approach aims to provide a systematic mechanism for detecting fake identities on social networks utilizing advanced image editing tools. It starts off with this interesting dataset, the "Fake Face Photos by Photoshop Experts," which is an explicitly crafted facial image dataset for splitting real and unreal faces. The data undergoes preprocessing which includes: organizing, balancing resizing, normalizing or separating the data for training and testing to make sure that there is a good proportion of both real and fake images [32].

The essence of the proposed method can be found in its modeling stage, where a combination of deep learning using Xception architecture and fine-tuning model parameters utilizing Snake optimization are employed. Xception model by default which uses depthwise separable convolutions for efficient feature extraction. The proposed Snake optimization strategy imitates the way a snake moves and effectively changes its moving vertices in accordance with solution space features to improve the deep, complex hyper-parameters of Xception model. Due to the complexity needed

for image manipulations, this powerful combination helps train the model well from the dataset.

It is trained using the optimal parameter values found in Snake and tested for accuracy, precision, recall, F1-score to ensure it meets expectations. This evaluation includes a test performed on the training and validation datasets to detect an insight into real and fake images. Lastly, the proposed method is compared with other methods and models to demonstrate its efficiency over existing works in cybersecurity. This method of multiple evaluation not only solves the problem concerning technical difficulties but also makes significant progress in enhancing online autonomy and safety applications [30][31].

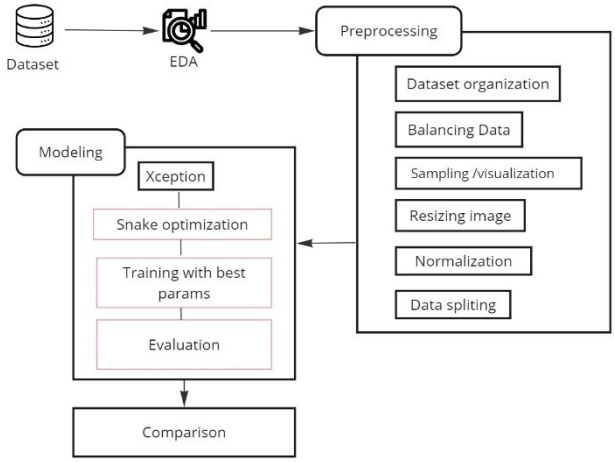

Fig. 1. Proposed approach

A. Dataset

The "Fake Face Photos by Photoshop Experts" dataset fills a crucial gap in identifying expertly manipulated face images. Unlike generative models, the dataset thus contains professionally manipulated images, which present varied and hard examples for training classifiers. It is a dataset with faces photoshopped professionally, i.e., some of the eyes, mouth, or whole face have been edited to make an actual but fake image. That makes it even more valuable for real-world scenarios because that reflects the sophistication in human-crafted fake face images.

The dataset [33] is divided into two main directories: training_real and training_fake. Finally, the training_real directory contains genuine face images that are used for training classifiers and rely upon a reference set to compare. The directory training_fake contains altered face, and the names of files show which parts were replaced or changed in a face. This makes it easier to analyze and train in a more controlled manner since users know exactly which perturbations are being utilized.

The dataset is preprocessed heavily to maintain data quality and balance it. Preprocessing: Organizing, Balancing, Resizing, Normalization, and Splitting into Train & Test Organizing while keeping real and fake images in tabbed categories for the same sake of separation. When balanced, the same number of real and fake images is there to protect against model training bias. Such changes may include resizing the images to make them all the same size, as not doing so would mean training your model on varying sizes of input times, which certainly will affect in return for output into prediction. Normalization scales pixel values to a standard range, which helps the model. We can validate a model with splitting the dataset into training and testing subsets. These preprocessing steps are essential to adequately prep the data for learning and investigation, thereby enhancing the generalizability and reproducibility of findings.

Because generative models like GANs can create an endless number of fake face images, and classifiers are likely to easily separate the real from the Facebook players. These methods are limited in their ability to capture features and patterns that generalize well for humans creating fake face images. These more sophisticated manipulations at the expert level in this dataset will have classifiers trained on it be able to better recognize fake faces that are generated by computers as well as human-created ones, making for easier real-world applications. Moreover, by specifying the manipulation type in detail, like identifying which part of a face has been modified, one can perform more specific analysis and training for building tailored methods to detect facial manipulations with high localization accuracy.

The "Fake Face Photos by Photoshop Experts" dataset can be applied not only in social networks, but also over to a variety of fields. For Online Authentications, we can use the dataset to develop better online authentication systems with more secured that prevent us from making a fake account. Authentication systems will better detect which identities actually belong to real users if you train classifiers using this dataset. Biometric-based authentication is another critical solution to bolster trustability in the verification process. This dataset is able to train models for face-based person verification with the ground truth being reliably preserved even in slightly tampered cases. As a huge margin of contemporaries and old guards in the field experiment with trained models for photo restoration prior to production, this dataset could be used to detect such manipulations, allowing professionals to stay up-to-date with new methods in the fast-evolving field of photo forgery.

The "Fake Face Photos by Photoshop Experts" dataset provides more than enough perfect images for training an extremely high-quality classification model, which makes it a rich, large-scale resource. This is very important for preventing the use of fake identities and information, which are core to security. It is so well-labeled and high-quality manipulated that classifiers trained on it are able to do a great job solving the problem of interest when they must be implemented in production.

The different possible use-cases for this dataset betray its significance in the realm of ensuring state credence through digital public commitments. In an increasingly internet-of-things world, this data will help maintain the trust of society at a fundamental level required to ensure digital integrity, helping detect and fend off fake imagery.

B. Preprocessing

We built a systematic preprocessing pipeline to process the “Fake Face Photos by Photoshop Experts” dataset for

training machine learning models. The main goal of this pipeline is to allow images being processed and tagged as correctly possible, thus optimizing learning process understanding the input data in an efficient way. The preprocessing consists of downloading the dataset from the access links supplied, after which on-decompressing all gathered files to extract necessary images. The dataset is divided into two sub-directories for the organization, "fake" and "real," which represent these classes. This first preprocessing process that organizes and balances the dataset is a fundamental step in order to have well-characterized classes without bias when training your model. Second, to make sure you are categorizing correctly here is a list of all subdirectories in the primary dataset folder [34]. The more we count the number of files in each folder corresponding to a class is instrumental in helping us understand how unbalanced or balanced our dataset. One key step is to visualize the number of images in a distribution across particular classes by using Seaborn. By doing this, you can spot any class imbalance at the beginning of the process which might degrade your model performance. The graph portrays a similarly distributed count for 1081 real and 960 fake images, thus implying that the dataset is balanced.

Random image sampling and visualization are used to create a bird's eye view on what the dataset contains and how good it is. More precisely, 5 images are randomly sampled from both classes (fake and real), using random). sample function. This step is needed to get a wide perspective about what the dataset contains. For visualization, the selected images are presented in one row, and axes are removed to make it easier for you to see the difference between two classes. This review helps to validate the distribution and quality of your dataset, which is paramount for proper model training.

After the primary check, the Python imaging library (PIL) is used to load and prepare images for model input. PIL is an amazing library that allows you to open, edit and save image files in a very easy way using Python. We resize all images to 240×240 pixels, so that the model input is of similar size for every image. It is necessary to do this step so that all images in the dataset are of equal dimensions and moresepver errors from happening while training a model. Another very important step is to normalize pixel values into the [0, 1] range by dividing each pixel value with 255.0. These are the normalization steps which normalize input and help our model to converge fast in training Shuffling input allows the neural network to see images more effectively and give optimal training results. In order to improve model robustness and generalization, the fake samples are subject to heavier augmentation than real ones. Augmentation: Generating minimum realistic distortions augment means adding gaussian noise, which could be expected in the real world while the model inference. The process is achieved by adding noise to the images and then multiplying it with a random factor. The np. ImageClipEnsures the pixel values of image remain in range [0, 1] which is required to prevent numeric instability with some and zero gradients are not supported. This augmentation helps the model to catch subtle manipulations, thus increasing its accuracy in genuine and fake images differentiation.

After importing the proper libraries, we are going to label the images meant for training our (hopefully) faithfully will predict malaria or not using LabelEncoder from sklearn. Sklearn preprocessing module - which will convert string values (fake and real) into numeric values. For each single data point, we mark the integers for its output layer and pass one-hot representation of these numeric labels into the to_categorical method provided by tensorflow. keras. Multi-classification requires the utils module. These classes must be word-wise one-hot encoded to allow the model to properly learn and differentiate between these during training.

To perform efficient model training and testing, the dataset is split into 80-20 parts of training datasets respectively with a train_test_split function from sklearn. model_selection module. A set random state is kept so that the result of using this method are repeatable and a nearly identical split occurs in different calls to code. This step is required to test the efficiency and accuracy of the model so that it can work well with new samples, which have not been seen before by a trained imported model. Preprocessed the file consists an overall 2041 images of resolution 240x240 along with one-hot labels, ready for machine learningeaningful information. This extensive preprocessing pipeline means the "Fake Face Photos by Photoshop Experts" dataset is clean and ready for any deep learning related subjects. It organizes and balances classes, visualizes image distributions, and random sampling to make sure the dataset is quality/diverse to give a strong basis. The steps of image loading, resizing, normalization and augmentation in the pipeline make models robust and generalize. Which ensures good Model Training and Testing. This systematic approach maintained at the preprocessing pipeline level upholds a high role in driving future accuracy and reliability of machine learning models to detect fake identities when deployed in real-world applications.

C. Xception

The efficiency and performance of the proposed approach, based on the Xception model, were significantly enhanced. Xception, short for Extreme Inception, is built on the inception CNN architecture using depthwise separable convolutions [35][36]. Designed by François Chollet, the creator of Keras, Xception simplifies the inception block by using a separable convolutional operation, separating spatial and depth operations into two layers: depthwise convolution followed by point-wise convolution [37]. This simplification captures more complex patterns while reducing computational cost. The Xception model contains 36 convolutional layers organized into 14 modules, each with at least one residual connection similar to the ResNet architecture, aiding in training deeper layers by addressing the vanishing gradient problem [38]. Additionally, Xception can be pre-trained with ImageNet weights, enhancing the model's performance on smaller datasets. This architecture is known for its high accuracy and low computational demand, balancing complexity and efficiency [39]. Its strong feature discrimination powers make it an excellent base model for image classification and other vision tasks.

In this work, the deep learning model was built using Xception for image classification. The pre-trained weights on

the ImageNet dataset served as the base model. Additional layers tailored the model to the specific classification task, including a GlobalAveragePooling layer, two fully connected Dense layers with 128 and 64 neurons, respectively, and a Dropout layer to prevent overfitting. A Dense layer followed by a softmax function generated class probabilities.

During training, the pre-trained layers of the Xception model remained locked to retain the learned weights. The new layers were trained for three epochs on our dataset using the Adam optimizer [40] and categorical cross-entropy loss. The model was then trained for ten epochs on a test set. Training was monitored with accuracy and loss curves, showing results for each epoch [41]. Post-training, the model was tested on the test set, and its classification accuracy was calculated. Additionally, a confusion matrix combined with precision-recall at threshold values [42] provided insights into the model's performance across all classes. The evaluation confirmed that the model based on the Xception network architecture performed well in image classification, validating its effectiveness through various metrics.

D. Snake Optimization Algorithm

Snake Optimization Algorithm (SOA) [43] is a kind of nature-inspired optimization algorithm inspired by the behaviors snakes adopt to search and found their optimal solution while roaming in the searching space. Here, a "snake" is the current solution with position and velocity. Initially, snakes are placed in random locations and set to move with different velocities across the search space. In each iteration of the algorithm, snakes shift according to their current velocity, their position in the parameter space, and the behavior of the best snake. Additionally, a sinusoidal movement simulates the wriggling of a snake, preventing local minima entrapment and enhancing search space exploration.

Hyperparameter optimization [45] is crucial for fine-tuning SOA parameters to enhance performance accuracy. The snake-like SOA control allows the algorithm to dynamically explore and converge to high-quality solution regions in various optimization applications. In this study, the SOA method was applied to tune hyperparameters for the Xception architecture. SOA simulates snake movements to search for optimal parameter combinations, aiming to maximize testing accuracy. The hyperparameters include the number of filters in the first dense layers and the dropout rate.

Initially, a population of snakes with random positions and velocities is generated. For the new population, distances are calculated using a sine function as an offset to move towards optimal positions. In the SOA process, each snake is evaluated for fitness based on which an Xception-based model is trained with that snake's hyperparameters. Accuracy in classification for the validation set is then computed [47]. The performance of the best snake is averaged to update all snakes' positions and velocities iteratively.

The resulting model, using optimal hyperparameters determined by SOA, consists of a pre-trained Xception model with additional dense layers for classification. This fine-tuning process freezes pre-trained layers while training only the new layers [48]. Model performance is evaluated by classification accuracy, confusion matrix [49], and classification report [50]. The results demonstrate SOA's effectiveness in tuning hyperparameters, leading to a better deep learning model for image classification tasks.

IV. EVALUATION METRICS

To effectively assess how useful the model is in the classification of images, the following key evaluation metrics will be used. The aim is to comprehensively test the performance of the model.

A. Accuracy

Accuracy [51] is an important metric that helps to estimate the overall quality of the model's predictions. It is described as the proportion of correct classifications, consisting of true positives and true negatives.

$$ACC = \frac{TN + TP}{TP + TN + FP + FN}$$

B. Precision

Precision [52], this measure is critical in assessing whether a model can predict a class accurately. It is determined by the number of true positive predictions divided by the sum of true positive and false positive predictions.

$$PRE = \frac{TP}{FP + TP}$$

C. Recall

Recall [53], this metric accounts for how well the model was able to correctly spot all of the class instances. It comprises all of the True Positives divided by the True Positives ratio added to the False Negatives.

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

D. F1-Score

The F1-score [54] is a balanced non-positive metric that mathematically combines the harmonic mean of precision and recall. Indeed, this measure is used to assess the performance of the model in the selection of relevant instances and assess the increased relevance for both measures.

$$F1 - S = 2 \times \frac{PRE \times REC}{PRE + REC}$$

E. Area Under the ROC Curve (AUC)

The receiver operating characteristic curve (ROC) can be described by the area under the ROC curve (AUC), which gives an aggregate measure of performance across all possible classification thresholds. It shows the true positive rate versus false positive and represents sensitivity against 1-specificity. A high AUC represents a perform model.

$$AUC = \int_0^1 TPR(FPR)d(FPR)$$

Together, these metrics provide a comprehensive evaluation of the model's ability to detect fake identities in social networks. Accuracy is a general measure of model performance, while precision and recall offer nuanced insights into class-specific prediction accuracy and instance identification. The F1-score balances these aspects, and the

AUC adds an additional evaluation parameter by considering various classification thresholds. This complete review offers insights from both theoretical and practical standpoints, aiming for better performance-oriented solutions.

V. EXPERIMENTAL RESULTS

A. Xception

The experimental findings in the confusion matrix [50] and classification report [55] show the ability of the Xception model to accurately differentiate real images from fake images. From the confusion matrix, Fig. 2, 190 out of 192 fake images were correctly identified, and only two were misclassified as real. Conversely, 210 out of 217 real images were accurately classified, while the remaining seven were misclassified as fake. This implies that the selected model is quite accurate in identifying fake and real images accurately [56].

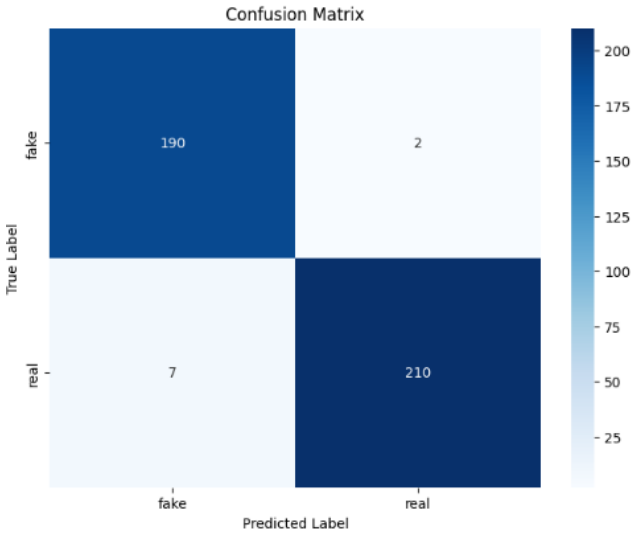

Fig. 2. Confusion matrix of Xception model

The classification report, Table II provides a detailed analysis of the model's performance, such as precision, recall, and F1-scores. The model has a precision for fake images of 0.96, meaning 96% of all images classified as fake are indeed fake. The recall for fake images is 0.99, suggesting that the model can detect 99% of fake images without false negatives. For real images, the precision and recall are 0.99 and 0.97, respectively, meaning that most images classified as real are accurate and the model detected 97% of actual real images.

TABLE II. CLASSIFICATION REPPORT OF XCEPTION MODEL

	Precision	Recall	F1-score
Fake	0.96	0.99	0.98
Real	0.99	0.97	0.98
Accuracy			0.98
Marco avg	0.98	0.98	0.98

The F1-scores offer additional insights by balancing precision and recall. Fake and real images have similar F1-scores of 0.98, indicating that the model has impressive accuracy in reducing false positives and false negatives [57]. Overall, the model has an accuracy of 0.98, which reflects the proportion of correct predictions out of all predictions made. Similarly, the macro and weighted averages for precision, recall, and F1-scores are all 0.98 for all classes, indicating consistent performance across all metrics. The macro average treats all classes equally, while the weighted average considers the number of instances in each class [58].

In summary, it can be concluded that the Xception model is reliable for deepfake detection. The high precision and recall rates for both classes show that this model performs accurate predictions for both real and fake images. Given the model's robust and accurate behavior, applications include digital forensics, social network security, and authentication. Moreover, the confusion matrix and classification report provide detailed analysis and validation of the model's performance.

B. Xception with Snake Optimization

The experimental findings demonstrate the significant performance enhancements of the Xception model using the Snake Optimization Algorithm (SOA). SOA optimized the hyperparameters with a filter size of 64 and a dropout rate of 0.1. These parameters were essential because a filter size of 64 extracted the critical features for image classification efficiently while maintaining computational feasibility. The dropout rate of 0.1 was crucial in reducing overfitting and maintaining the model's robustness through proper regularization.

These optimal hyperparameters for the Xception model, derived using SOA, resulted in a highly robust model in terms of classification. The confusion matrix, Fig. 3 shows that the model correctly identified 189 out of 192 fake images, misclassifying only three as real. For real images, 214 out of 217 were correctly classified, with the remaining three classified as fake. These results highlight the model's robustness in identifying both real and fake images accurately [59].

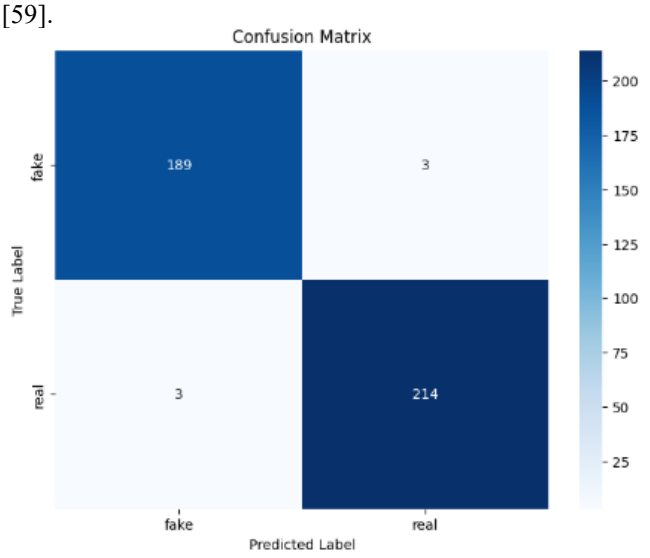

Fig. 3. Confusion matrix of Xception with snake optimization

In Table III, the classification report provides detailed metrics that prove the ability of the model to perform. For fake images, the precision is 0.98, which means that 98% of the images predicted as fake were actually fake. The recall rate for fake images is also 0.98, implying that 98% of all images meant to predict were caught. A high recall rate

indicates the ability of the model to identify fake images with a few or no misses. For the real-life images, the precision rate is 0.99, and it implies that 99% of the images predicted to be real were actually real. Similarly, the recall rate for real images is 0.99, indicating 99% of all real images predicted correctly.

TABLE III. CLASSIFICATION REPPORT OF XCEPTION WITH SNAKE OPTIMIZATION

	Precision	Recall	F1-score
Fake	0.98	0.98	0.98
Real	0.99	0.99	0.99
Accuracy			0.99
Macro avg	0.99	0.99	0.99

The F1-scores provide additional insights by balancing precision and recall. Both fake and real images have an F1-score of 0.98 for fake and 0.99 for real, indicating the model's good performance in managing precision and recall, resulting in fewer false positives and false negatives [60]. The model's overall accuracy rate is 99%, indicating the proportion of correct predictions out of all predictions made. Furthermore, the macro and weighted average values for precision, recall, and F1-scores are all 0.99, showing that the model performs consistently across different metrics. The macro average treats all classes equally, while the weighted average considers the number of instances for each class, ensuring balanced performance [60].

These results indicate the effectiveness of the Xception model when optimized with SOA, as it has demonstrated high robustness in identifying whether an image is real or fake. Due to the high precision, recall, and F1-scores of both classes, we can use this model for many real-life problems that require established recognition patterns in images, such as detecting deepfakes. Overall, the Xception model optimized with SOA showed high gains compared to the standard model. Experimental results and analysis show significant improvements in performance according to precision, recall, F1-score, and accuracy. The comparative analysis between the standard Xception model and the optimized version with Snake Optimization reveals the optimization gains in deepfake detection and its superiority for these tasks.

VI. COMPARATIVE ANALYSIS

The results comparison effectively, Table IV demonstrates how the performance of the Xception model has improved after optimization with the Snake Optimization Algorithm (SOA) and subsequently compares it to the standard, unoptimized version of the Xception model. This comparison provides systematic insight into how the proposed method fares against other approaches for deepfake detection.

Consistent improvements across all key metrics are observed, showcasing the superior performance of the Snake-optimized Xception model. In the non-optimized Xception model, it scored an accuracy of 97.80%, precision of 97.75%, recall of 97.87%, and an F1-score of 97.79%. These performance metrics indicate a reasonably performing model, but not as reliable and accurate as required.

After optimization with SOA, the Xception model showed significant improvements: accuracy, precision, recall, and F1-score all increased to 98.53%. This uniform improvement across all measures suggests that Snake optimization made the model more accurate and consistent in its predictions.

These performance differences become more apparent when examining their confusion matrices. The confusion matrix for the base Xception model showed true positives of 190 for fake images and 210 for real images, with minimal misclassifications (2 fakes as real and 7 reals as fake). Despite these strong performance indicators, there was room for improvement. Conversely, the Snake-optimized Xception model exhibited positive improvements with a true positive rate of 189 for fake images and 214 for real images, with even fewer misclassifications (3 fake images misclassified as real and vice versa). This demonstrates the high accuracy and robustness of the Snake-optimized model.

This improvement can also be seen in the classification report. For the fake class, precision increased from 0.96 to 0.98, recall remained at 0.98, and the F1-score stayed at 0.98. For the real class, both precision and recall improved from 0.97 to 0.99, resulting in an F1-score of 0.99. These improvements show the optimized model's enhanced ability to accurately identify both fake and real images.

The macro average for precision, recall, and F1-score also improved to 0.99 from the standard Xception's 0.98, indicating the enhanced performance and robustness of the Snake-optimized model. This is vital for efficient deployment in real-life systems where high accuracy is imperative.

Finally, when employing the Snake optimization shared here, the Xception model notably enhances all performance metrics by reducing misclassifications and improving precision, recall, and F1-score across all classes. This Snake-optimized model is highly effective at accurately distinguishing deepfake faces from real ones, demonstrating that advanced optimization techniques can significantly boost the performance of deep learning models.

TABLE IV. COMPARATIVE TABLE

Metric	Xception	Xception with
Snake		
Accuracy	97.80%	98.53%
Precision	97.75%	98.53%
Recall	97.87%	98.53%

To further enhance the model's performance and applicability, future research could explore additional optimization techniques, investigate the model's performance on larger and more diverse datasets, and address specific challenges in real-world deployment. These directions could provide valuable insights and contribute to the continuous improvement of deepfake detection technologies.

VII. DISCUSSION

In Table V, Compared to other works in the deepfake detection domain utilizing the Xception model, the Snake-optimized solution is the most practical and robust approach.

Scholars have taken distinct approaches to tackle the issue of deepfake detection, and each was successful to some extent.

A pragmatic solution by [19] combines LBP-Net, which finds texture differences between fake and real faces through Local Binary Patterns. The resultant image is then input to an efficient Lightweight Bridge Proposal network (LB-Net), outperforming ResNet18 and Gram-Net with higher accuracy and precision, although the exact metrics and dataset specifics were not mentioned, causing some ambiguity regarding its applicability.

In another research work, [20] applies InceptionNet, a type of Convolutional Neural Network (CNN), for detecting deepfakes on a Kaggle dataset with 401 training videos and 3,745 images. This study reports an accuracy of 93%, which is respectable but does not match the performance of our optimized model. Similarly, [23] also uses InceptionNet on the same dataset and achieves an identical accuracy of 93%, focusing mostly on distinguishing original from manipulated content.

The study by [21] investigates multiple CNN architectures, including a custom-built CNN, VGG19, and DenseNet-121, on synthetic face images that visually simulate real or fake identities. VGG19 achieves the highest accuracy at 95%, emphasizing the importance of facial recognition in deepfake detection. However, this accuracy is significantly lower than the 98.53% achieved by our Snake-optimized Xception model.

Further advancements are seen in [22], which introduces FFF-CDG, using Fuzzy Fisher Face with Capsule Dual Graph on datasets like FFHQ and 100K-Faces, achieving an accuracy of 95.82% on FFHQ. This method addresses inefficiencies in existing techniques but still falls short of our results. Similarly, [24] uses Bidirectional-LSTM to exploit temporal features in the DFDC dataset, achieving an accuracy of 82.65%. Despite its robustness against compressed videos with noise interference, its accuracy is considerably lower compared to our method.

TABLE V. COMPARATIVE TABLE WITH RELATED WORK

Paper	Methodology	Key Features	Dataset	Results	Highlights
[19]	LBP-Net	Leverages Local Binary Patterns (LBP) to capture texture differences	Various (not specified)	Outperforms ResNet18 and Gram-Net	Ensemble models further enhance detection performance
[20]	InceptionNet	Uses CNN (InceptionNet) for identifying deepfakes	Kaggle (401 training videos, 3,745 images)	Accuracy: 93%	Comparative analysis with different CNNs
[21]	Custom CNN, VGG19, DenseNet-121	Focuses on facial recognition for deepfake detection	Augmented dataset of real and fake faces	VGG19 achieves highest accuracy: 95%	Addresses deepfakes in legal contexts
[22]	FFF-CDG	Uses Fuzzy Fisher Face with Capsule Dual Graph	FFHQ, 100K-Faces, DFFD, VGG-Face2, Wild Deepfake	Accuracy: 95.82% on FFHQ	Addresses inefficiencies of existing methods
[23]	InceptionNet	Uses CNN (InceptionNet) for deepfake detection	Kaggle (401 training videos, 3,745 images)	Accuracy: 93%	Focuses on distinguishing original from manipulated content
[24]	Bidirectional-LSTM	Exploits temporal features for deepfake detection in videos	DFDC dataset	Accuracy: 82.65%	Robust against compressed videos with noise interference
[25]	YOLO-CNN-XGBoost	YOLO for face detection, InceptionResNetV2 for feature extraction, XGBoost for recognition	CelebDF-FaceForensics++ (c23)	AUC: 90.62%, Accuracy: 90.73%	Superior performance compared to state-of-the-art techniques
[26]	EfficientNet	Face Deepfake Detection and Reconstruction Challenge	CelebA, FFHQ, deepfake images from various GANs	VisionLabs Accuracy: 93.61%	Detection in “wild” scenarios, reconstructing original images
[27]	FF-LBPH, DBN	Combines Fisherface with Local Binary Pattern Histogram (LBPH) and Deep Belief Networks (DBN)	FFHQ, 100K-Faces, DFFD, CASIA-WebFace	Accuracy: 98.82% on CASIA-WebFace, 97.82% on DFFD	Addresses high computational demands and inaccuracy
[28]	AMTENnet	Adaptive manipulation traces extraction network (AMTEN) integrated with CNN	Various FIM techniques	Accuracy: 98.52% (general), 95.17% (post-processed images)	Superior pre-processing and detection capabilities
[29]	ADD (Attention-based DeepFake Detection)	Utilizes face close-up and shut-off data augmentation methods	Celeb-DF (V2), WildDeepFake	Accuracy: >98.3% with ResNet on Celeb-DF (V2)	Enhances detection performance of multiple classifiers
Our Work	Xception, Xception with Snake Optimization	Uses CNN (Xception) with Snake optimization to enhance detection performance	Real and fake faces dataset	Xception: Accuracy: 97.80%, Precision: 97.75%, Recall: 97.87%, F1-score: 97.79% Xception with Snake: Accuracy: 98.53%, Precision: 98.53%, Recall: 98.53%, F1-score: 98.53%	Significant improvement in all metrics with Snake optimization

The approach in [25], combining YOLO for face detection, InceptionResNetV2 for feature extraction, and XGBoost for recognition, achieves an AUC of 90.62% and an accuracy of 90.73% on the CelebDF-FaceForensics++ (c23) dataset. While superior to many state-of-the-art techniques, it still does not reach the high accuracy of our Snake-optimized model. In [26], EfficientNet achieves an accuracy of 93.61% on various datasets, including CelebA and FFHQ, which is impressive but not as high as our results.

Moreover, [27] jointly uses Fisherface with Local Binary Pattern Histograms (LBPH) and Deep Belief Networks (DBN), achieving an accuracy of 98.82% on CASIA-WebFace and 97.82% on DFFD. Although these results are competitive, they still slightly lag behind our model's performance. AMTENnet, described in [28], integrates Adaptive Manipulation Traces Extraction Network (AMTEN) with CNN and achieves an accuracy of 98.52% on general datasets and 95.17% on post-processed images, indicating superior pre-processing and detection capabilities but not surpassing our model.

Finally, [29] proposes ADD (Attention-based DeepFake Detection), which applies close-up and blackout face methods on data augmentation, achieving over 98.3% accuracy with ResNet on Celeb-DF (V2). While highly effective, it is still marginally outperformed by our method.

The Snake-optimized Xception model not only achieves a higher accuracy of 98.53% but also excels in precision, recall, and F1-score, each at 98.53%. These metrics underscore the model's enhanced capability in accurately identifying deepfakes compared to existing approaches. The consistent performance across these key evaluation metrics highlights the robustness and reliability of our proposed method in deepfake detection.

Our approach demonstrates the highest accuracy and F1-scores, surpassing many state-of-the-art methods, especially after incorporating Snake optimization. This highlights the potential of our method in achieving superior deepfake detection performance compared to existing techniques. Our work sets a new benchmark in the field, showcasing the effectiveness of combining advanced CNN architectures with innovative optimization techniques to achieve unparalleled results in deepfake detection.

VIII. CONCLUSION

In this thesis, we propose an end-to-end solution using the Xception model, further improved by our new feature called Snake optimization. In response to ever more sophisticated digital forgeries, our research aims to fulfill the strong demand for innovative detection technologies. With the introduction of depthwise separable convolutions, the Xception model can efficiently extract features, and we employ Snake optimization to dynamically tune parameters, navigating through this complex and high-dimensional parameter space. This integrated approach has demonstrated significant improvements in performance metrics, achieving an accuracy, precision, recall, and F1-score of 98.53%.

The rigorous testing on diverse datasets, selected for their complexity and sophisticated manipulations, highlights the robustness and adaptability of our proposed method. These results surpass both the standard Xception model and many state-of-the-art methods, establishing our framework as a leading solution in the realm of digital media forensics. In comparison to related works, our Snake-optimized Xception model stands out as the most effective and reliable. Previous studies, including those utilizing LBP-Net, InceptionNet, VGG19, and other CNN architectures, have achieved varying degrees of success but do not match the high performance of our optimized model. Advanced methodologies like FFF-CDG and AMTENnet still fall short of our accuracy metrics. The use of Bidirectional-LSTM for temporal feature exploitation and the combination of YOLO, InceptionResNetV2, and XGBoost for recognition, though robust against certain challenges, also do not achieve the high accuracy of our Snake-optimized model. Even competitive approaches such as ADD (Attention-based DeepFake Detection) with over 98.3% accuracy are marginally outperformed by our method.

Thus, despite the magnitude of our findings, there are naturally some inherent constraints to this study. However, the model is only impressive if it works well on particular datasets that do not encompass the entire range of possible deepfake techniques and manipulations in practice. Common reasons for failure include the model being vulnerable to new adversarial attacks and relying on high-quality input data. Its diversity in terms of the dataset, lack of real-world data, and slow-release cycle raise questions that should be addressed by future work aiming to make this model robust against new deepfakes or any potential modifications/updates on contemporary fake videos.

Moreover, the ethical consequences and societal effects of deepfake detection methods are substantial. Deepfakes rightfully raise privacy, misinformation, and trust concerns in digital media. It is structured to be sensible with respect to these ethical considerations; respecting the privacy of data, not reinforcing algorithmic biases, and independently preventing misuse of detection technologies. Responsible research and technology development and difficult dialogues on these ethical issues are a must.

The conclusion suggests that the Snake-optimized Xception model is better than other methods; however, additional validation work in independent studies and external-internal evaluation seem important after reading this paper. Comparisons to paired analyses by others would also add credibility to our conceptions. Although, it is very important to know the explainability of the model. The identification of these mechanisms increases transparency and reinforces accountability, which ensures the model's predictions are trustable and subject to review.

Despite how much this proposed framework does to advance deepfake detection, it is not as though the battle has been won. Digital manipulations involve continuous novelty and mutation, and in order to stay ahead of this transformation, real-time authenticity is essential. Every day, we hear about some new program that cannot be digitally counterfeited. In future work, we should focus on the efficiency of Snake optimization, various deepfake enhancements with this model, and investigation of real-world content stations. It could add another level of acumen

to next research work on interpretability and applicability in different situations.

To summarize, this research establishes a new state-of-the-art in deepfake detection and demonstrates the potential efficacy of using both sophisticated CNN architectures along with elaborate optimization. Further investigation and cooperation are important for exploring antidotes to better, sustainable deepfake detection. We can build a more secure and sustainable digital media environment by understanding the complexity of this phenomenon and addressing ethical issues.

Therefore, our work offers great promise for understanding and fighting digital forgery to ensure that the authenticity of all online content is upheld. We call upon future researchers to extend our work and explore new avenues, solutions, or tools that can help solve the existing problem of deepfake detection so as to prevent fake media from going viral.

REFERENCES

[1] J. Banumathi *et al*., "An Intelligent Deep Learning Based Xception Model for Hyperspectral Image Analysis and Classification," *CMC*, vol. 67, no. 2, pp. 2393–2407, 2021, doi: 10.32604/cmc.2021.015605.

[2] B. U. Mahmud and A. Sharmin, "Deep Insights of Deepfake Technology: A Review," *arXiv preprint arXiv:2105.00192*, 2021.

[3] M. Dang and T. N. Nguyen, "Digital Face Manipulation Creation and Detection: A Systematic Review," *Electronics*, vol. 12, no. 16, p. 3407, 2023, doi: 10.3390/electronics12163407.

[4] E. Ferrara, S. Cresci, and L. Luceri, "Misinformation, manipulation, and abuse on social media in the era of COVID-19," *J. Comput. Soc. Sci.*, vol. 3, no. 2, pp. 271–277, 2020, doi: 10.1007/s42001-020-00094-5.

[5] E. U. H. Qazi, T. Zia, and A. Almorjan, "Deep Learning-Based Digital Image Forgery Detection System," *Appl. Sci.*, vol. 12, no. 6, p. 2851, 2022, doi: 10.3390/app12062851.

[6] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma, and Y. Liu, "Countering malicious deepfakes: Survey, battleground, and horizon," *International Journal of Computer Vision*, vol. 130, no. 7, pp. 1678-1734, 2022.

[7] L. Peng *et al*., "A Multi-strategy Improved Snake Optimizer Assisted with Population Crowding Analysis for Engineering Design Problems," *J. Bionic Eng.*, vol. 21, no. 3, pp. 1567–1591, 2024, doi: 10.1007/s42235-024-00505-7.

[8] D. Pan, L. Sun, R. Wang, X. Zhang, and R. O. Sinnott, "Deepfake detection through deep learning," in *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pp. 134-143, 2020.

[9] A. Diwan and U. Sonkar, "Visualizing the truth: a survey of multimedia forensic analysis," *Multimed. Tools Appl.*, vol. 83, no. 16, pp. 47979–48006, 2024, doi: 10.1007/s11042-023-17475-3.

[10] G. Hu, R. Yang, M. Abbas, and G. Wei, "BEESO: Multi-strategy Boosted Snake-Inspired Optimizer for Engineering Applications," *J. Bionic Eng.*, vol. 20, no. 4, pp. 1791–1827, 2023, doi: 10.1007/s42235-022-00330-w.

[11] K. Shaheed, A. Mao, I. Qureshi, M. Kumar, S. Hussain, I. Ullah, and X. Zhang, "DS-CNN: A pre-trained Xception model based on depth-wise separable convolutional neural network for finger vein recognition," *Expert Systems with Applications*, vol. 191, p. 116288, 2022.

[12] M. M. El-Gayar, M. Abouhawwash, S. S. Askar, and S. Sweidan, "A novel approach for detecting deep fake videos using graph neural network," *J. Big Data*, vol. 11, no. 1, pp. 1–27, 2024, doi: 10.1186/s40537-024-00884-y.

[13] O. A. H. H. Al-Dulaimi and S. Kurnaz, "A Hybrid CNN-LSTM Approach for Precision Deepfake Image Detection Based on Transfer Learning," *Electronics*, vol. 13, no. 9, p. 1662, 2024, doi: 10.3390/electronics13091662.

[14] M. Tian, M. Khayatkhoei, J. Mathai, and W. AbdAlmageed, "Unsupervised Multimodal Deepfake Detection Using Intra- and Cross-Modal Inconsistencies," *arXiv*, vol. 2311.17088, 2023.

[15] H. Jeon, Y. Bang, and S. S. Woo, "FDFtNet: Facing Off Fake Images Using Fake Detection Fine-Tuning Network," *ICT Systems Security and Privacy Protection*, 2020, doi: 10.1007/978-3-030-58201-2_28.

[16] M. Masood, M. Nawaz, K. M. Malik, A. Javed, A. Irtaza, and H. Malik, "Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward," *Appl. Intell.*, vol. 53, no. 4, pp. 3974–4026, 2023, doi: 10.1007/s10489-022-03766-z.

[17] X. Wu, X. Liao, B. Ou, Y. Liu, and Z. Qin, "Are Watermarks Bugs for Deepfake Detectors? Rethinking Proactive Forensics," *arXiv preprint arXiv:2404.17867*, 2024.

[18] M. Tayseer, J. Mohammad, M. Ababneh, A. Al-Zoube, and A. Elhassan, "Digital Forensics and Analysis of Deepfake Videos," in *11th International Conference on Information and Communication Systems (ICICS)*, 2020.

[19] L. Bojic, N. Prodanovic, and A. D. Samala, "Maintaining Journalistic Integrity in the Digital Age: A Comprehensive NLP Framework for Evaluating Online News Content," *arXiv preprint arXiv:2401.03467*, 2024.

[20] Y. Wang, V. Zarghami, and S. Cui, "Fake Face Detection using Local Binary Pattern and Ensemble Modeling," in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3917-3921, 2021, doi: 10.1109/ICIP42928.2021.9506460.

[21] P. Theerthagiri and G. Nagaladinne, "Deepfake Face Detection Using Deep InceptionNet Learning Algorithm," in *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pp. 1-6, 2023, doi: 10.1109/SCEECS57921.2023.10063128.

[22] M. Taeb and H. Chi, "Comparison of Deepfake Detection Techniques through Deep Learning," *J. Cybersecur. Priv.*, vol. 2, no. 1, pp. 89–106, 2022, doi: 10.3390/jcp2010007.

[23] P. M. Arunkumar, Y. Sangeetha, P. V. Raja, and S. N. Sangeetha, "Deep Learning for Forgery Face Detection Using Fuzzy Fisher Capsule Dual Graph," *ITC*, vol. 51, no. 3, pp. 563–574, 2022, doi: 10.5755/j01.itc.51.3.31510.

[24] V. Nagagopiraju, K. Ayyappa, P. Anshulalitha, J. Srikanth, and K. T. Teja, "A Effciet Deep Fake Face Detection Using Deep Inception Net Learning Algorithm," *Turcomat*, vol. 15, no. 1, pp. 138–141, 2024, doi: 10.61841/turcomat.v15i1.14555.

[25] S. Pei *et al*., "A bidirectional-LSTM method based on temporal features for deep fake face detection in videos," in *Proceedings Volume 12346, 2nd International Conference on Information Technology and Intelligent Control (CITIC 2022)*, vol. 12346, pp. 311-318, 2022, doi: 10.1117/12.2653461.

[26] A. Ismail, M. Elpeltagy, M. S. Zaki, and K. Eldahshan, "A New Deep Learning-Based Methodology for Video Deepfake Detection Using XGBoost," *Sensors*, vol. 21, no. 16, p. 5413, 2021, doi: 10.3390/s21165413.

[27] L. Guarnera *et al*., "The Face Deepfake Detection Challenge," *J. Imaging*, vol. 8, no. 10, p. 263, 2022, doi: 10.3390/jimaging8100263.

[28] S. T. Suganthi, M. U. A. Ayoobkhan, N. Bacanin, K. Venkatachalam, H. Štěpán, and T. Pavel, "Deep learning model for deep fake face recognition and detection," *PeerJ Comput. Sci*., vol. 8, p. e881, 2022, doi: 10.7717/peerj-cs.881.

[29] Z. Guo, G. Yang, J. Chen, and X. Sun, "Fake face detection via adaptive manipulation traces extraction network," *Comput. Vision Image Understanding*, vol. 204, p. 103170, 2021, doi: 10.1016/j.cviu.2021.103170.

[30] A. Khormali and J.-S. Yuan, "ADD: Attention-Based DeepFake Detection Approach," *Big Data Cogn. Comput*., vol. 5, no. 4, p. 49, 2021, doi: 10.3390/bdcc5040049.

[31] A. Shah, S. Varshney, and M. Mehrotra, "Detection of Fake Profiles on Online Social Network Platforms: Performance Evaluation of Artificial Intelligence Techniques," *SN Comput. Sci*., vol. 5, no. 5, pp. 1–15, 2024, doi: 10.1007/s42979-024-02839-9.

[32] H. Else, "Publishers unite to tackle doctored images in research papers," *Nature*, 2021, doi: 10.1038/d41586-021-02610-7.

[33] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge

Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, p. 652801, 2021, doi: 10.3389/fenrg.2021.652801

[34] SQL-Injection-Extend. (2024, May 17). Retrieved from https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection

[35] V. W. de Vargas, J. A. S. Aranda, R. D. S. Costa, P. R. da Silva Pereira, and J. L. V. Barbosa, "Imbalanced data preprocessing techniques for machine learning: a systematic mapping study," *Knowl. Inf. Syst.*, vol. 65, no. 1, pp. 31–57, 2023, doi: 10.1007/s10115-022-01772-8.

[36] S. Sharma and S. Kumar, "The Xception model: A potential feature extractor in breast cancer histology images classification," *ICT Express*, vol. 8, no. 1, pp. 101-108, 2022.

[37] A. Dhillon G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85-112, 2020.

[38] S. Pashine, S. Mandiya, P. Gupta, and R. Sheikh, "Deep fake detection: survey of facial manipulation detection solutions," *arXiv preprint arXiv:2106.12605*, 2021.

[39] A. Satapathy and J. Livingston LM, "A lightweight convolutional neural network built on inceptio-residual and reduction modules for deep facial recognition in realistic conditions," *The Imaging Science Journal*, vol. 71, no. 1, pp. 14-32, 2023.

[40] C. E. S. Rex, J. Annrose, and J. J. Jose, "Comparative analysis of deep convolution neural network models on small scale datasets," *Optik*, vol. 271, p. 170238, 2022.

[41] P. Sumari *et al*., "A Precision Agricultural Application: Manggis Fruit Classification Using Hybrid Deep Learning," *Rev. d'Intelligence Artif.*, vol. 35, no. 5, pp. 375-381, 2021.

[42] R. Adityatama and A. T. Putra, "Image classification of Human Face Shapes Using Convolutional Neural Network Xception Architecture with Transfer Learning," *Recursive Journal of Informatics*, vol. 1, no. 2, pp. 102-109, 2023.

[43] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[44] F. A. Hashim and A. G. Hussien, "Snake Optimizer: A novel meta-heuristic optimization algorithm," *Knowledge-Based Systems*, vol. 242, p. 108320, 2022.

[45] "CSDL | IEEE Computer Society," May 08, 2024.

[46] B. Bischl *et al*., "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, 2023.

[47] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295-316, 2020.

[48] G. P. Bhandari, R. Gupta, and S. K. Upadhyay, "An approach for fault prediction in SOA-based systems using machine learning techniques," *Data Technologies and Applications*, vol. 53, no. 4, pp. 397-421, 2019.

[49] H. Shavit, F. Jatelnicki, P. Mor-Puigventós, and W. Kowalczyk, "From Xception to NEXcepTion: New Design Decisions and Neural Architecture Search," *arXiv preprint arXiv:2212.08448*, 2022.

[50] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label classifier performance evaluation with confusion matrix," *Computer Science & Information Technology*, vol. 1, pp. 1-14, 2020.

[51] J. Görtler *et al*., "Neo: Generalizing Confusion Matrix Visualization to Hierarchical and Multi-Output Labels," *arXiv preprint arXiv:2110.12536*, 2021.

[52] S. A. Macskassy, F. Provost, and S. Rosset, "ROC confidence bands: An empirical evaluation," in *Proceedings of the 22nd international conference on Machine learning*, pp. 537-544, 2005.

[53] A. Tharwat, "Classification assessment methods," *Applied computing and informatics*, vol. 17, no. 1, pp. 168-192, 2021.

[54] A. M. Carrington *et al*., "Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 329-341, 2022.

[55] O. Saidani *et al*., "Student academic success prediction in multimedia-supported virtual learning system using ensemble learning approach," *Multimedia Tools and Applications*, pp. 1-26, 2024.

[56] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F1 and macro-averaged F1 scores," *Appl. Intell.*, vol. 52, no. 5, pp. 4961–4972, 2022, doi: 10.1007/s10489-021-02635-5.

[57] S. M. Abdullah *et al*., "An Analysis of Recent Advances in Deepfake Image Detection in an Evolving Threat Landscape," *arXiv preprint arXiv:2404.16212*, 2024.

[58] D. J. Hand, P. Christen, and N. Kirielle, "F*: an interpretable transformation of the F-measure," *Mach. Learn.*, vol. 110, no. 3, pp. 451–456, 2021, doi: 10.1007/s10994-021-05964-1.

[59] T. Gowda, W. You, C. Lignos, and J. May, "Macro-Average: Rare Types Are Important Too," *arXiv preprint arXiv:2104.05700*, 2021.

[60] G. Pei *et al*., "Deepfake Generation and Detection: A Benchmark and Survey," *arXiv preprint arXiv:2403.17881*, 2024.

[61] K. Riehl, M. Neunteufel, and M. Hemberg, "Hierarchical confusion matrix for classification performance evaluation," *arXiv preprint arXiv:2306.09461*, 2023.