E-ISSN: 2707-6644
P-ISSN: 2707-6636
www.computersciencejournals.com/ijcpdm
IJCPDM 2024; 5(2): 23-33
Received: 21-06-2024
Accepted: 30-07-2024

Bisme AS
Research Scholar, Department of Computer Science, Banasthali Vidyapith, Rajasthan, India

CK Jha
Professor, Department of Computer Science, Banasthali Vidyapith, Rajasthan, India

Sneha Asopa
Assistant Professor, Department of Computer Science, Banasthali Vidyapith, Rajasthan, India

Corresponding Author:
Bisme AS
Research Scholar, Department of Computer Science, Banasthali Vidyapith, Rajasthan, India

Optimizing CNN-based Deepfake detection with firefly algorithm: A Hybrid Approach

Bisme AS, CK Jha and Sneha Asopa

DOI: https://doi.org/10.33545/27076636.2024.v5.i2a.101

Abstract
Deepfake identification is crucial for ensuring the validity of digital material. This research seeks to improve the accuracy and efficacy of deepfake detection by improving a Convolutional Neural Network (CNN) using the Firefly Algorithm. We tested and compared the Firefly-optimized CNN model to a baseline CNN and the MobileNetV2 model on a dataset of authentic and fraudulent pictures. The CNN model was fine-tuned using the Firefly Algorithm, and MobileNetV2 served as a baseline for a more complicated design. The findings show that the Firefly-optimized CNN beats the baseline CNN and MobileNetV2 in classification accuracy, precision, recall, and ROC curve value. The Firefly-optimized CNN attained an accuracy of 99.09% and a ROC curve value of 0.99, outperforming the baseline CNN's accuracy of 95.98% and ROC value of 0.96, as well as MobileNetV2's accuracy of 93.48% and ROC value of 0.93. This research finds that adding optimization approaches such as the Firefly Algorithm may considerably improve CNN models' performance for deepfake detection, providing a solid answer for enhancing digital content verification procedures.

Keywords: CNN, deep fake, firefly algorithm, optimization, image classification

Introduction
Deepfake technology is a rapidly expanding subject that uses Artificial Intelligence (AI) and Machine Learning (ML) techniques to create incredibly realistic fake media [1]. These kinds of media can be used maliciously for things like identity theft, disinformation campaigns, and cyberbullying. While conventional methods like image editing software have been used for years to produce false media, deepfake's AI makes it much harder to identify and prevent these forgeries.

Deepfakes can be in many different forms, such as convincingly fake photos or movies that pass for the real thing. When it comes to Deepfake photos, artificial intelligence algorithms produce lifelike depictions of people who don't exist, which can be used for deceptive purposes [2]. Deepfake videos, on the other hand, use previously recorded video content to create new videos that are hard to distinguish from authentic ones. By fabricating video evidence of someone's actions, these tactics can be used to fabricate stories or blackmail them [3].

However, some markers, including face inconsistencies or lighting anomalies, can still be utilized to identify phony media, even with the growing sophistication of Deepfake technology [4]. Advanced techniques like machine learning algorithms can be utilized to spot patterns suggestive of phony media in order to detect deepfakes. Machine learning models, for instance, can be trained to distinguish between a genuine and artificial face [5]. To improve the model's accuracy in identifying Deepfake, optimization techniques like Particle Swarm Optimization (PSO) can be applied.

Deepfake technology's introduction has raised serious questions about how it might be abused. While conventional methods for identifying fraudulent media advanced techniques like machine learning algorithms and optimization algorithms like PSO, however they may not be as successful against deepfakes, have enormous potential for detecting fake media and reducing the negative effects of its use.

Lower-quality deepfakes are easy to identify since they frequently include features like uneven skin tones, poor lip-synching, and visible imperfections surrounding transposed faces [6]. Additionally, Deepfake technology still faces substantial challenges in replicating small features, including individual hair strands [7].

Inadequately depicted jewelry, teeth, and peculiar lighting effects including uneven illumination and iris reflections are other noteworthy identifiers [8]. Numerous organizations, including governments, academic institutions, and tech corporations, have started to sponsor research efforts to uncover and counteract this technology's negative effects in response to the growing worry over the prevalence and dangerous potential of deepfake [9]. One such project is the Deepfake Detection Challenge, which was launched to encourage healthy competition among international research teams in creating reliable detection algorithms and is sponsored by Microsoft, Facebook, and Amazon.

The main goal of deepfake research is to minimize false positives while achieving high accuracy in identifying different types of fake videos on the internet or social media. This is essential for fostering a safe atmosphere and limiting the dissemination of false information in the media and rumors, which can spark social unrest and political unrest [10]. The suggested strategy, which combines the firefly algorithm and CNN-based algorithms, was selected because it has the ability to increase deepfake detection rates in comparison to current models [11]. Because deepfakes are created using sophisticated AI and ML techniques, other methods, like conventional image and video analysis, might not be able to identify them. Because conventional detection techniques are not made to take into consideration the intricacy and sophistication of AI-generated material, they may be unable to detect deepfakes, which are incredibly lifelike [12]. Therefore, in order to increase the precision and dependability of deepfake detection, the suggested method of utilizing cutting-edge deep learning algorithms and optimization approaches is crucial.

Effective detection methods are essential to lessen the impact of deepfakes, given how widespread they are and the potential harm they might do, including identity theft, extortion, sexual exploitation, reputational damage, and harassment. By utilizing cutting-edge technology to increase deepfake detection accuracy, this research fills this requirement and eventually makes the internet a safer and more secure place.

Literature review

In order to increase detection rates above current models, we present a novel deepfake detection method in this research that makes use of the firefly algorithm and a CNN-based deepfake detection. The increasing quantity of deepfakes being created and the possible harm they may bring about—such as identity theft, extortion, sexual exploitation, reputational damage, and harassment are the driving forces behind our suggested method. The authors in [13] introduced a convolutional vision transformer (CVT) that is composed of a CNN (ViT) and a vision transformer in order to detect deepfakes. The CNN extracts the learnable features, and the ViT uses an attention approach to classify the learnt features as input. The model achieved an accuracy rate of 91.5 percent, an AUC value of 0.91, and a loss value of 0.32 after being trained on the Deepfake Detection Challenge Dataset (DFDC). The authors' addition of a CNN module to the ViT architecture and their accomplishment of a competitive outcome on the DFDC dataset constitute the contribution.

A method for identifying misleading movies (RNN) is proposed in [14] using convolutional neural networks (CNN) and recurrent neural networks combined with transfer learning in AutoEncoders. Test input data that have not yet been seen are analyzed to evaluate if the model is generalizable. Furthermore, the effect of residual picture input on the accuracy of the model is investigated. To illustrate the effectiveness of transfer learning, outcomes are presented for both scenarios in which it is used and in which it is not.

You Only Look Once convolution recurrent neural networks (YOLO-CRNNs) have been presented in [15] as a means of identifying deepfake films. Following the YOLO face detector's identification of the face areas in each video frame, a customized EfficientNet-B5 is utilized to extract the spatial features of these faces. These attributes are sent into a bidirectional long short-term memory (Bi-LSTM) as input sequences in order to retrieve the temporal properties. The new method is then tested on a new large-scale dataset, CelebDF-FaceForencics, which combines FaceForencies with Celeb-DF. It achieves 89.35% area under the receiver operating characteristic curve (AUROC), 89.38% accuracy, 85.5 percent precision, 83.15 percent recall, and 84.3 percent F1 measure with the pasting data technique.

In order to identify fraudulent movies using a neural network and expedite the process of locating deepfake videos, writers in [16] described how to use an integral video frame extraction approach. The Xception net was chosen to be paired with our classifier instead of InceptionV3 or Resnet50. The model provides a way to find visual anomalies. The CNN module's feature vectors are fed into the next classifier network, which then classifies the video. The top model was obtained using the Face Forensics and Deepfake Detection Challenge datasets. With the Face Forensics and Deepfake Detection Challenge datasets combined, the model achieved 92.33% accuracy; with the Face Forensics dataset, it achieved 98.5% accuracy. A specific self-distillation fine-tuning method was proposed by the authors in [17] to further improve the detector's durability and performance. The gathered characteristics can then be utilized to simultaneously describe the latent patterns of films across spatial and temporal frames, resulting in the development of a potent and precise deepfake video detector. The greatest area under curve (AUC) score of 99.94% on the Face Forensics benchmark and the surpassing of 12 accuracy levels of at least 7.90% and 8.69% on the challenging DFDC and Celeb-DF (v2) benchmarks, respectively, demonstrate the effectiveness of their methodology, which has been put through extensive testing and in-depth analysis.

In [18], a portable 3D CNN is recommended for deepfake detection. The channel transformation module is made to extract characteristics at a higher level with fewer parameters. 3D CNNs are used as a spatial-temporal module to combine spatial data with a time component. Spatial-rich model characteristics are taken from the input frames to suppress frame content and accentuate frame texture, hence improving the performance of the spatial-temporal module. According to experimental results, the suggested network uses less parameters than existing networks and outperforms other deepfake detection methods on widely used deepfake data sets. To detect fraudulent films, scientists proposed a vision transformer model in [19] that used a distillation method. In order to overcome the issue of false negatives, a CNN features and patch-based positioning model that

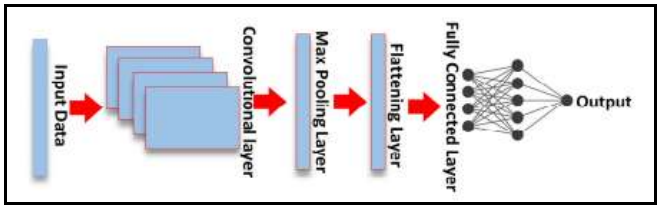
interacts with all locations to identify the artifact region are constructed. The suggested technique, which used patch embedding as input using the concatenated CNN features, obtained 91.9 f1 scores and 0.978 accuracy through a comparative examination of the Deepfake Detection (DFDC) Dataset.

Background

Convolutional Neural Network (CNN)

The CNN network is a deep learning model that can identify highly abstracted characteristics in objects [20]. It is thus excellent for visual picture analysis and identification [21]. On the other hand, a layer of the CNN model may also learn the characteristics of sequence data with numerous variables. It may thus be used to any prediction problem. Regarding fault identification, reference [22] demonstrates the careful density use of CNN in detecting and classifying the various kinds and intensities of rolling bearing faults. By analyzing the imbalance of a wind turbine's supervisory control and data acquisition (SCADA) data, the CNN architecture was employed in [23] to tackle the blade icing issue.

A typical CNN model, as seen in Figure 1, consists of the convolutional layer, pooling layer, flattening layer, and fully connected layer, as stated in reference [24]. The primary part of the CNN network is the convolutional layer, which uses weight sharing and sliding windows to minimize computing complexity. The kernel approach is used at this layer to extract different characteristics from the incoming data. The pooling layer is the following layer. By minimizing the connections between layers and executing each feature map individually, this layer aims to minimize the size of the feature map that is involved. Reducing dimensionality and extracting the dominating features is the primary objective of the pooling process in order to effectively train the model [25]. Max pooling and average pooling are two of the several forms of pooling processes. The flattening layer must be used to produce a one-dimensional vector before moving on to the fully connected linked layer (FC), since the FC layer comprises of weights, biases, and neurons to connect the neurons amongst the various layers. Occasionally, the FC layer is added as the last layer before the CNN network's output layer.



Firefly algorithm

In 2008, Xinshe Yang, a professor from Cambridge, proposed the firefly algorithm (FA) [13]. The FA is a swarm intelligence-based random search algorithm that mimics the natural attraction process between individual fireflies. When creating mathematical models of fireflies, the following idealization criteria are used to enhance certain firefly characteristics:

- The only factor attracting fireflies, male or female, across gender boundaries is light intensity;
- The brightness of the light affects the firefly' attraction to one another.
- The value of the objective function that has to be improved is correlated with the brightness of fireflies.

According to the firefly algorithm, fireflies have a special lighting mechanism and behavior. The light they emit can only be perceived by other fireflies within a specific range for two reasons: air can absorb light, and the light intensity *I* and distance from the light source r, are inversely correlated. The fundamental idea behind the standard FA is that, in a population of randomly distributed fireflies, brighter fireflies draw fuzzier fireflies toward them. Then, each firefly approaches a firefly in a solution space that has a high absolute brightness, updates its position, completes an iteration of positions, and locates an optimal position to achieve optimization. Firefly brightness and the value of the goal function are correlated. Fireflies will attract more if they are brighter and in better settings. The fireflies with the highest brightness travel at random because they are not attracted to other fireflies. It is necessary to establish the connection between firefly and the goal function value. The objective function value expresses the algorithm's absolute brightness; that is, the value of the objective function at position x_i^t equals the absolute brightness I_i of a firefly placed at $x_i^t\left(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD}\right)$

$$\varphi = \varphi_0 * \sigma \tag{9}$$

If a firefly with lower brightness is drawn to one with a higher brightness, then the relative brightness of the two fireflies is $I_{ij}\left(r_{ij}\right) = I_i e^{-\gamma r_{ij}^2}$. The absolute brightness of firefly i is represented by I_i; the cartesian distance between firefly i and firefly j is represented by r_{ij}; and the light absorption coefficient, γ, is often set as a constant. That is

$$r_{ij} = \left\| x_j^t - x_i^t \right\| = \sqrt{\sum_{k=1}^{D}\left(x_j^t - x_i^t\right)^2} \tag{10}$$

Assuming a proportionality between brightness and attraction between fireflies, the attraction is β_{ij}.

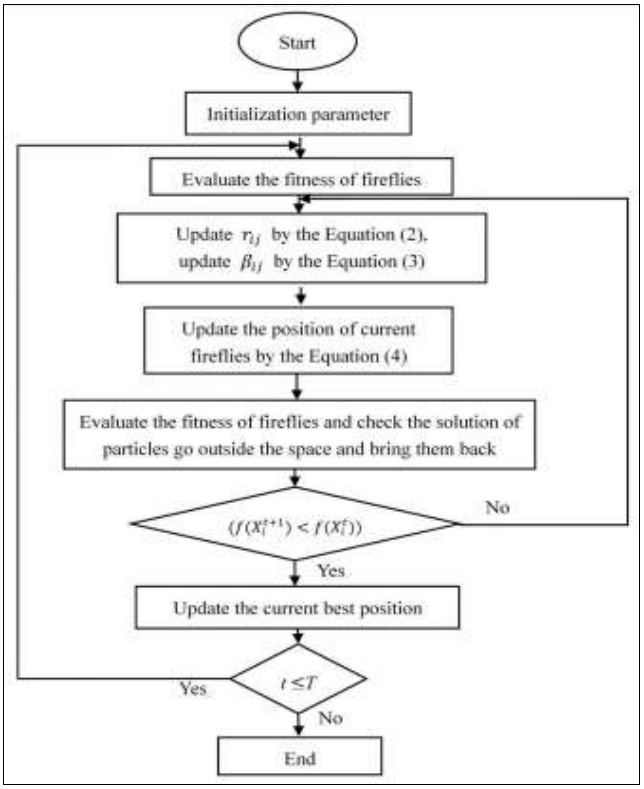
$$\beta_{ij}\left(r_{ij}\right) = \beta_0 e^{-\gamma r_{ij}^2} \tag{11}$$

The biggest appeal is β_0, and often. The algorithm's update formula is as follows:

$$x_i^{t+1} = x_i^t + \beta_{ij}\left(r_{ij}\right) \times \left(x_j^t - x_i^t\right) + \alpha * \delta_i \tag{12}$$

In this case, α is a random term coefficient, t is the number of iterations, and δ_i is a random integer generated from a uniform, Gaussian, or other distribution.

The flow chart of the standard firefly algorithm (FA) is shown in the Fig 1:



Yang used FA to optimize a test function with singularities in order to improve optimization performance because of the limitations of the regular FA. The pressure piping design global optimization issue was successfully solved using the FA, and the findings demonstrate that the FA is capable of solving this kind of global optimization problem. Nevertheless, the standard FA's parameters are predetermined, which might cause the algorithm to converge too soon or prevent it from converging altogether because of incorrect parameter values. Therefore, in order to get greater optimization performance, the standard FA has to be improved.

The primary factor influencing each firefly's location update is its ability to attract other fireflies with intense fluorescence to itself. The research demonstrates that after several location updates, the whole search space may converge to the local optimum since the global search is not random. In order to quantify group diversity, the average distance length from individual to group center must first be determined, as indicated in Equation

$$\beta^n = \frac{1}{|S|}\sum_{x=1}^{|S|}\sqrt{\sum_{y=1}^{m}\left(c_{xy} - \overline{c\gamma}\right)^2} \tag{13}$$

Where $|S|$ denotes the population size, $\overline{c\gamma}$ is the j-dimensional component of the average center of the population, and β^n is the diversity index of the n-th generation population.

The firefly location update approach is further modified in light of the diversity features previously mentioned. The following is the updated location update formula:

$$cx^{n+1} = cx^{n} + \mu 0 e^{-L\delta_{xy}^2}\left(c_x - c_y\right) + \rho * \left(c_x - c_b\right) + \varphi\left(r - \mathrm{n}\right) \tag{14}$$

Where ρ is a weight that varies with population variety and repetition periods and has some unpredictability, and c_b is the firefly individual that is now ideal. Here is the formula for calculating ρ.

$$\rho = \begin{cases} -r * \sigma & \beta^n \leq \sigma * \beta^0 \\ 0 & \beta^n > \sigma * \beta^0 \end{cases} \tag{15}$$

Where σ is a linear declining function that becomes less as the number of repetitions increases, and $\beta = 0$ is the population diversity index at the beginning of time. Here is the σ calculation formula: Equation, where T_{itex} is the number of iterations that are currently in progress and $T_{\max}$ is the maximum number of iterations.

$$\sigma = \frac{T_{\max} - T_{itex}}{T_{\max}} \tag{16}$$

In order to ensure a wider search range, the probability of a firefly individual will move toward the irregular direction away from the best in the initial stage of the algorithm. However, in the later stage of the algorithm, when the diversity requirements are reduced, the firefly will randomly search in the direction closest to the best in order to achieve a more focused local search. An algorithm's convergence to local optimization may be better avoided by using a dynamic adjustment mechanism that balances the issues of global optimization in the early stages and local optimization in the later stages.

The step factor φ is specified in the FA. Most fireflies will do local optimum search around the ideal point in the latter stages of the algorithm, and a set step size may cause the algorithm to fail to converge to the best point. Therefore, the step size factor is dynamically altered based on the number of iterations in order to balance the contradiction between the early and late phases of the algorithm convergence process as follows:

$$\varphi = \varphi_0 * \sigma \tag{17}$$

Where the first step factor is denoted by φ_0. It has finished improving the firefly algorithm so far.

Methodology

Data collection

For this study we collected the data from the kaggle which is from https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images/code. This dataset contains a collection of images classified as real and as well as fake, which provides a comprehensive basis for training and evaluating the deep fake detection model.

Data Pre-processing

Various preprocessing processes are used to guarantee consistency and enhance the performance of the deepfake

detection model, therefore preparing the dataset for efficient training and assessment. The following are the steps:

- **Normalization:** To enhance convergence during training, scale pixel values to a range of [0, 1]. By standardizing the input data, this technique facilitates the model's ability to learn from the characteristics and make generalizations.
- **Resizing:** To guarantee consistency in input size for the CNN, resize pictures to a standard dimension (e.g., 224x224 pixels). In order to preserve the integrity of feature extraction, this stage makes sure that every picture supplied into the model has the same spatial dimensions.
- **Data Augmentation:** Apply data augmentation methods to boost the dataset's variety, such as flipping, rotating, and color jittering. These methods lessen the chance of over fitting by exposing the model to a greater variety of picture changes, which improves generalization.

Train-Test Split: We will split our dataset into two parts training and testing subsets. Out of 100% 80% of data is utilized for training and remaining 20% of unseen data is utilized to evaluate the model efficiency.

Model Building: We use an architecture of Convolutional Neural Networks (CNNs) specifically designed for deepfake detection. In order to take use of learnt representations, the CNN is built using pre-trained feature extraction layers. These are followed by specially constructed classification layers, which contain fully linked layers and a sigmoid activation function for binary classification. Using the preprocessed dataset, the model is adjusted to take into account properties unique to deepfake. The Firefly Algorithm is used to tune critical hyper parameters, such learning rate, batch size, and dropout rate, in order to improve model performance and avoid over fitting. This method guarantees that the CNN can discern between real and deepfake pictures with accuracy.

Optimization using Firefly algorithm: We use the nature-inspired optimization method Firefly Algorithm to maximize CNN model by adjusting hyper parameters like learning rate, batch size, and dropout rate. This method simulates the flashing behavior of fireflies to identify optimum solutions by repeatedly changing hyper parameters depending on model performance measures like accuracy and F1-score. The Firefly Algorithm improves the generalizing and performance accuracy of the model by assessing it with many sets of hyper parameters and updating them depending on their efficacy, therefore enhancing the whole performance deepfake detecting capability.

Model Evaluation: The process of "model evaluation" looks at whether a produced model can be applied to fresh data in order to determine its generalizability. Among the many criteria for categorization performance are accuracy, precision, recall, F1 score, specificity, and sensitivity. By comparing estimates many times, cross validation ensures their accuracy. This implies that confusion matrices, for example, have a wealth of information to provide. Assessment is essential for development and deployment.

Performance Metrics

Accuracy: The simplest way to measure how often the classifier makes correct predictions is by using accuracy. Another way of looking at it is that this is the percentage of correct forecasts relative to all guesses.

$$Accuracy = \frac{TP + TN}{S} \ (eq-1)$$

Precision: In contrast to this ratio in addition to one minus from it, i.e., (1-Precision), which presents the percentage false negatives; 1/Precision yields recall.

$$\Pr ecision = \frac{TP}{TP + FP} \ (eq-2)$$

Recall: On other hand there are called false negatives in relation with True Negatives.

$$\mathrm{Re}\, call = \frac{TP}{TP + FN} \ (eq-3)$$

F1-Score: It is calculated by taking the accuracy and recall scores and squaring them. For this.

$$F1 = \frac{2 * \Pr ecision * \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call} \ (eq-4)$$

Results and Discussion

Results

In this part, we report the results of our trials utilizing the CNN model improved using the Firefly Algorithm, comparing its performance against a baseline CNN model with fixed hyper parameters and the MobileNetV2 model. Each model was trained and assessed on a dataset comprising of genuine and fake photos. The assessment measures employed include accuracy, precision, recall, and validation accuracy. The comparison focuses on how the Firefly-optimized CNN model improves upon the baseline CNN in terms of performance while being more computationally economical, and how it matches up against the more complicated MobileNetV2 architecture in deepfake detection tests.

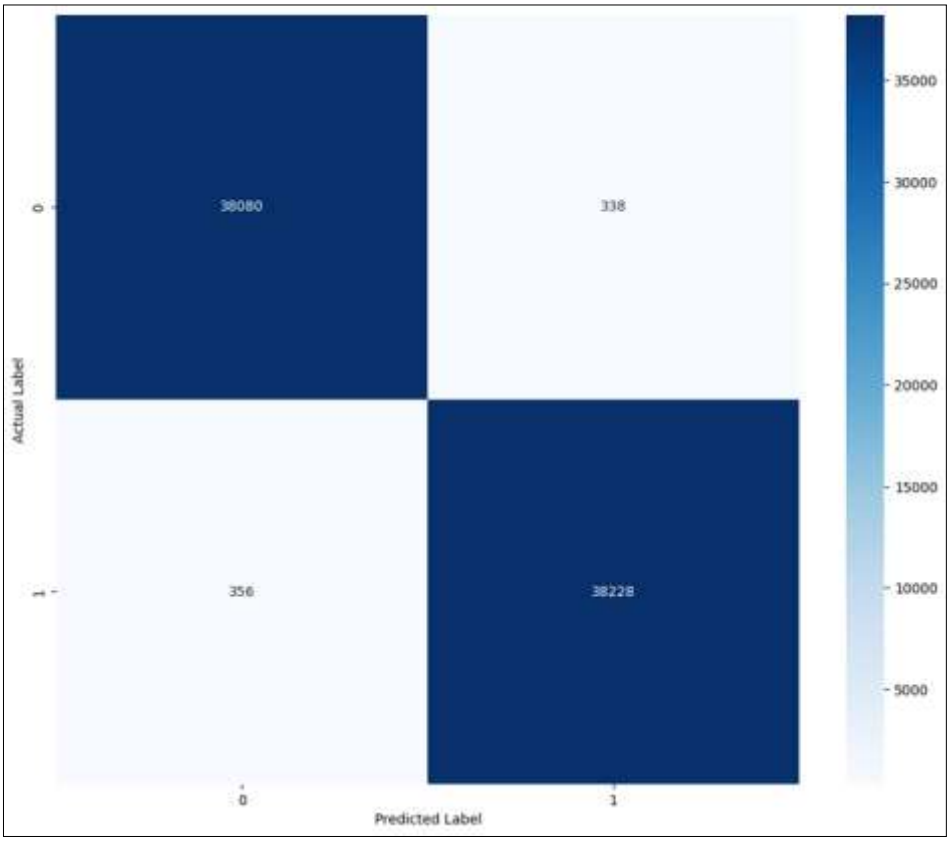


Fig 1: CNN with Firefly Algorithm

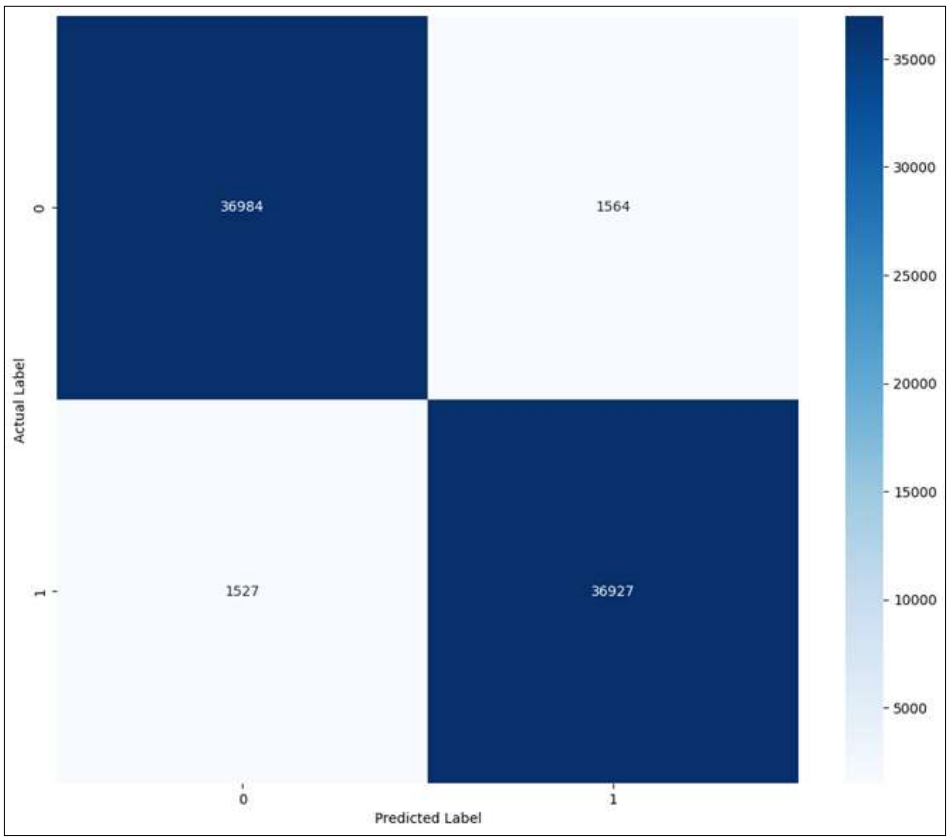


Fig 2: CNN confusion matrix

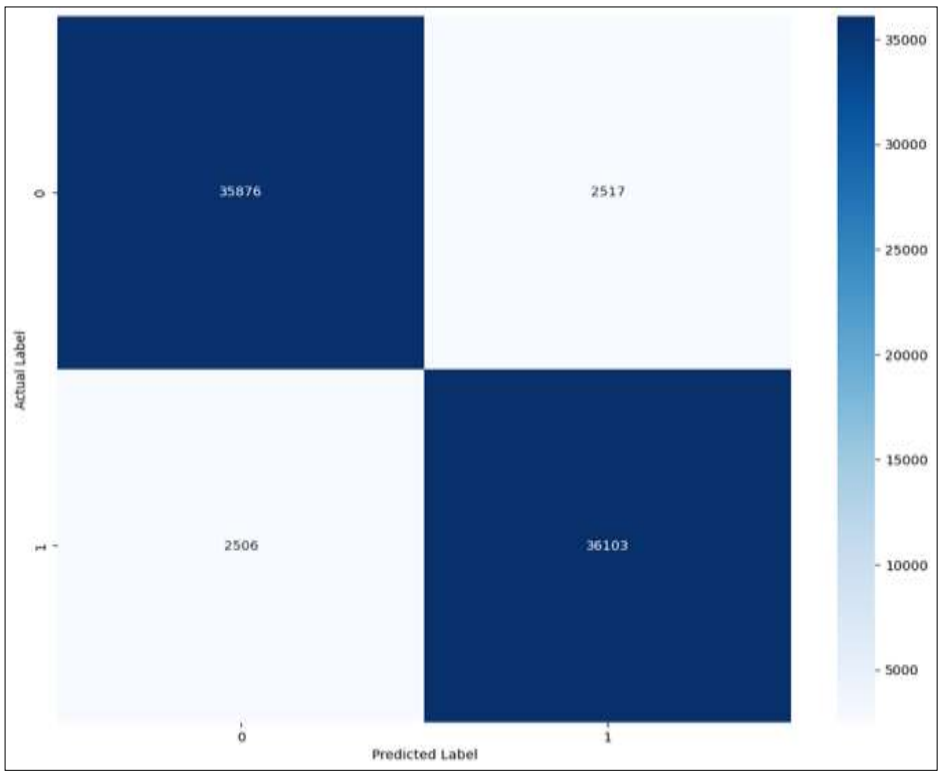


Fig 3: Mobile NetV2 confusion matrix

From the above confusion matrices, there are two classes: 0 (Fake) and 1 (Real). Out of 77,002 samples, our suggested model, CNN with Firefly Algorithm, properly identified 38,080 times as Fake (0) and marginally misclassified 338 times as Real (1). Similarly, it categorized 38,228 occasions as Real and misclassified 356 Real occurrences as Fake. This demonstrates excellent performance and little misclassification.

In contrast, the baseline CNN model properly recognized 36,984 occurrences as Fake but misclassified 1,564 as real, while correctly categorizing 36,927 as Real but misclassifying 1,527 as Fake. This reveals a greater misclassification rate than our recommended model.

On the other hand, the MobileNetV2 model detected 35,876 instances as Fake but misclassified 2,517 as real, while properly categorizing 36,103 as Real but misclassifying 2,506 as Fake. This exhibits the largest misclassification across the models, demonstrating that the CNN with Firefly Algorithm surpasses both the baseline CNN and MobileNetV2 in terms of correct classification of both Fake and Real data, displaying a more balanced and precise performance in deepfake detection.

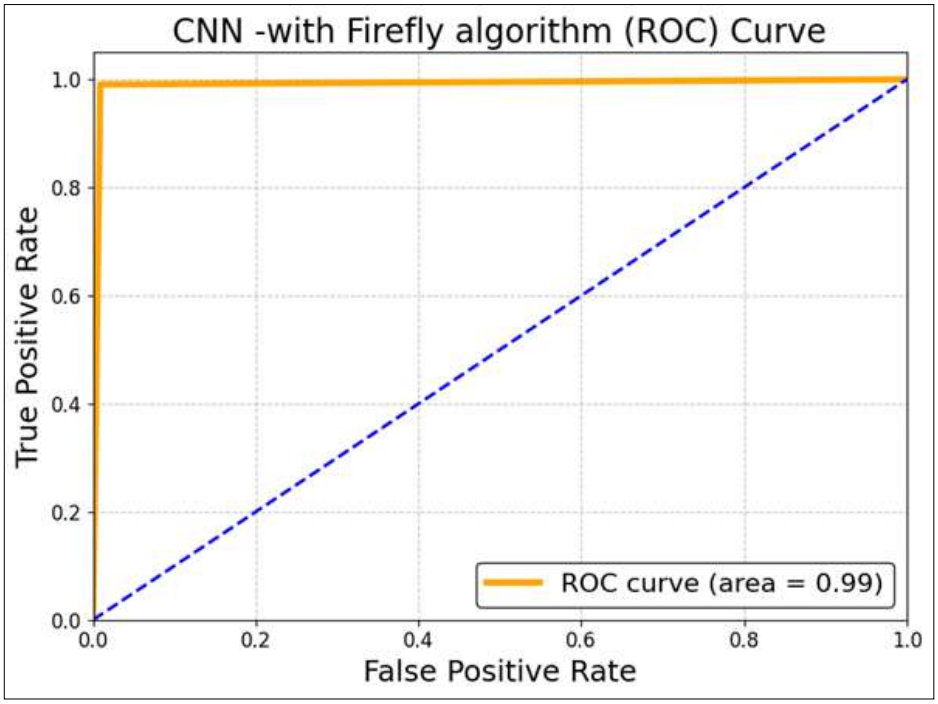


Fig 4: CNN with Firefly algorithm ROC

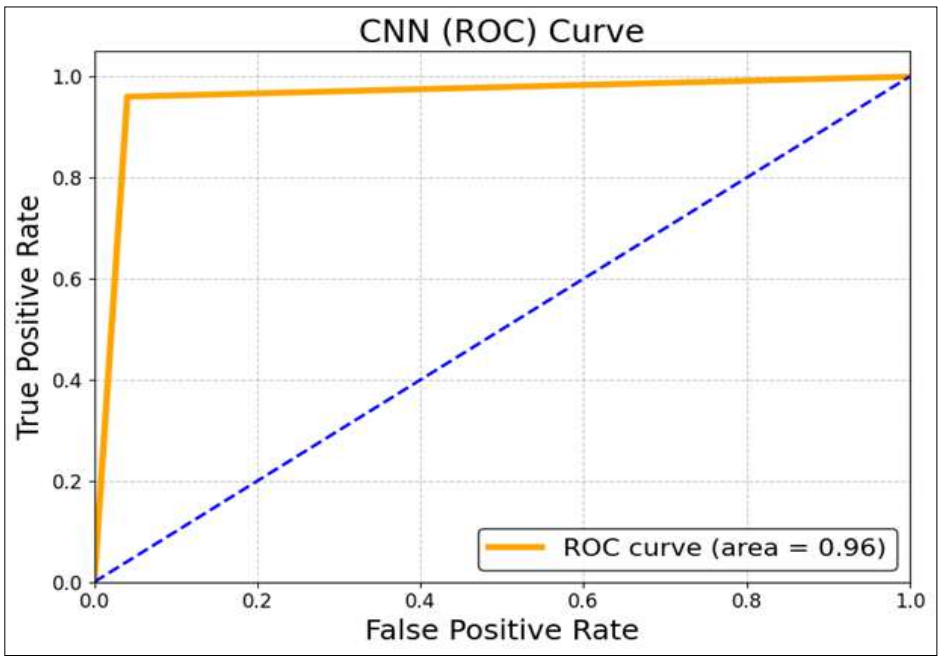


Fig 5: CNN ROC

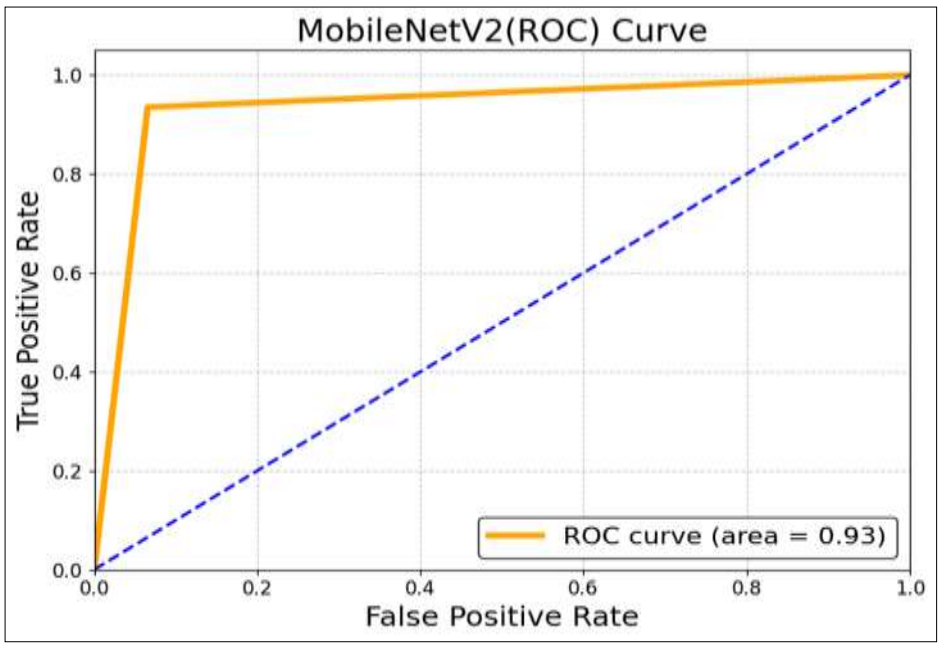


Fig 6: MobileNetV2 ROC

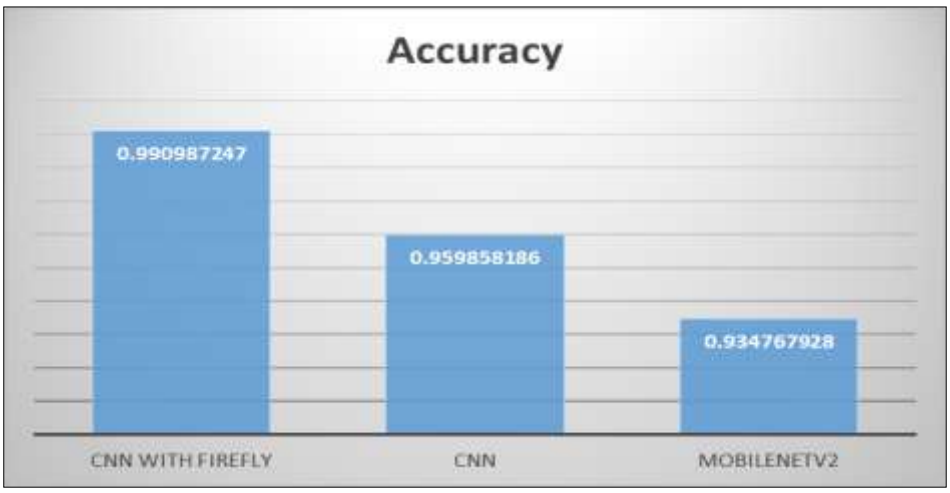
The ROC curve results illustrate the performance of each model in discriminating between fake (0) and Real (1) classes. Our suggested model, CNN with Firefly Algorithm, produced an amazing ROC value of 0.99, showing good discriminating between the two classes and near-perfect classification.

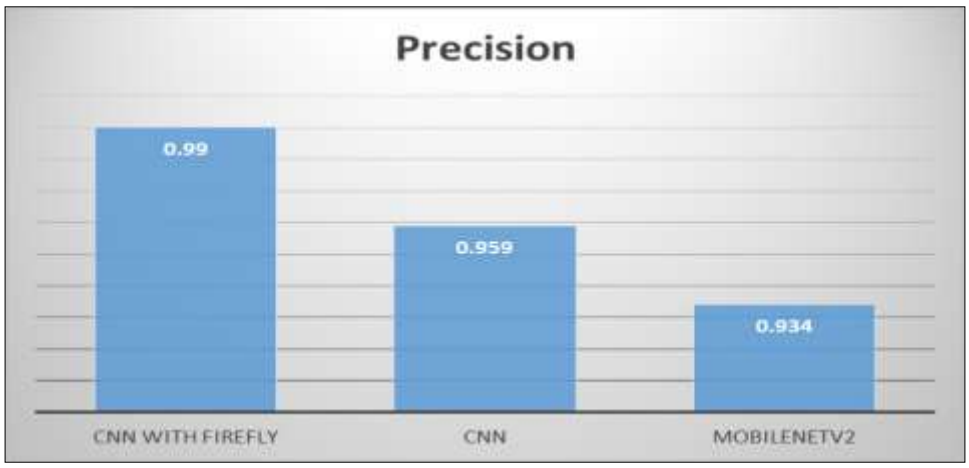
In contrast, the baseline CNN model obtained a slightly lower ROC value of 0.96, which still implies high classification performance but with somewhat less accuracy than the suggested model. The MobileNetV2 model earned a ROC value of 0.93, which, while excellent, demonstrates the worst performance among the models.

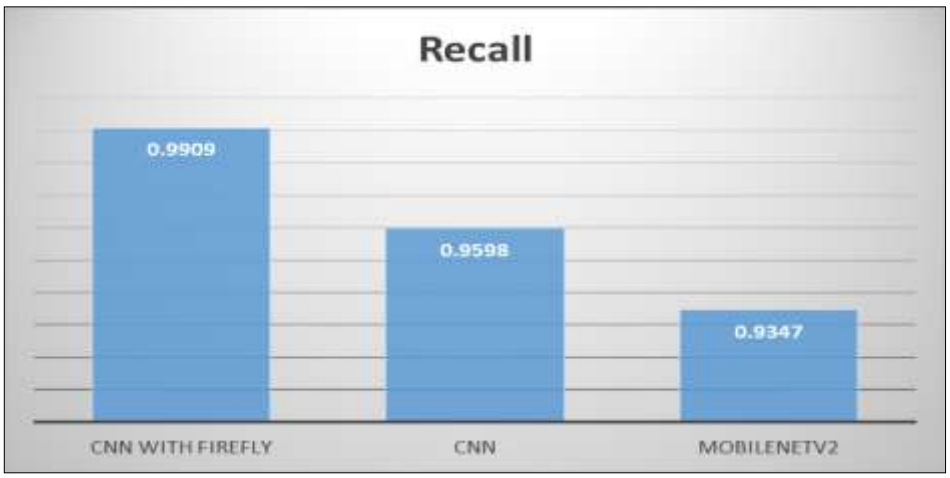
This comparison illustrates that the CNN with Firefly Algorithm beats both the baseline CNN and MobileNetV2 models in terms of overall classification accuracy and efficacy, as evidenced by the larger ROC value.

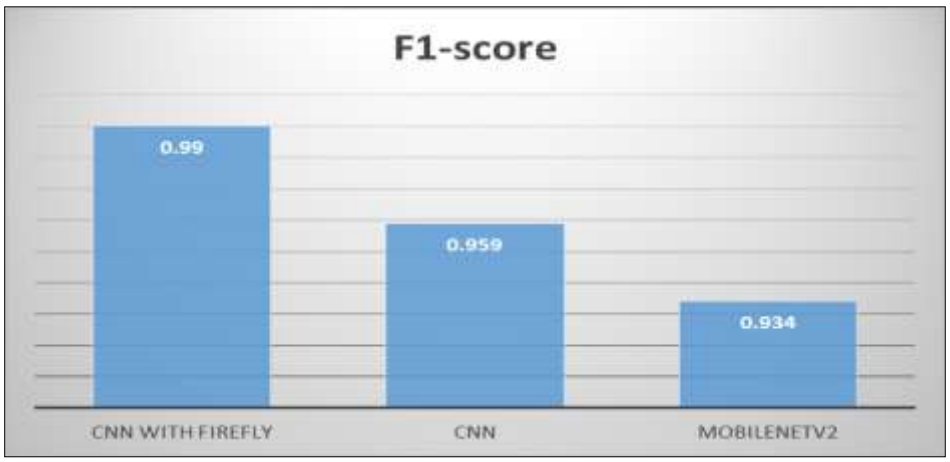
Table 1: Evaluation metrics

Models	Accuracy	Precision	Recall	F1-score
CNN with Firefly	0.99099	0.99	0.9909	0.99
CNN	0.95986	0.959	0.9598	0.959
MobileNetV2	0.93477	0.934	0.9347	0.934


Accuracy
0.990987247
0.959858186
0.934767928
CNN WITH FIREFLY
CNN
MOBILENETV2


Precision
0.99
0.959
0.934
CNN WITH FIREFLY
CNN
MOBILENETV2


Recall
0.9909
0.9598
0.9347
CNN WITH FIREFLY
CNN
MOBILENETV2


F1-score
0.99
0.959
0.934
CNN WITH FIREFLY
CNN
MOBILENETV2

The performance comparison shows that the CNN with Firefly Algorithm beats both the baseline CNN and MobileNetV2. The suggested model has an accuracy of 0.9909, precision of 0.99, recall of 0.9909, and F1-score of 0.99, indicating good classification capacity. In comparison, the baseline CNN model had an accuracy of 0.9598, with precision, recall, and F1-score values of 0.959, indicating effective but slightly less accurate performance. MobileNetV2, although still successful, has the lowest metrics: 0.9347 accuracy, 0.934 precision, 0.9347 recall, and an F1-score of 0.934. Overall, the CNN with Firefly Algorithm performs better in terms of accuracy and classification than the other models.

Discussion

The studies' findings show that the CNN model modified with the Firefly Algorithm outperforms both the baseline CNN and MobileNetV2 models in deepfake detection. The Firefly-optimized CNN has a fantastic accuracy of 0.9909, with precision and recall values of 0.99 and an F1-score of 0.99. This high level of performance is confirmed by a ROC curve value of 0.99, which indicates almost flawless classification abilities. This model's ability to accurately distinguish between fake and real pictures implies that the Firefly Algorithm considerably enhances CNN performance by tweaking hyper parameters for greater accuracy and dependability.

In contrast, the basic CNN, although effective, achieves significantly lower performance numbers. It has an accuracy of 0.9598, precision and recall values of 0.959, and an F1-score of 0.959. The ROC curve value of 0.96 shows good classification performance, however it falls short of the optimal CNN's capabilities. MobileNetV2, despite its more complicated design, has the lowest metrics of the three models, with an accuracy of 0.9347, a precision of 0.934, a recall of 0.9347, and an F1-score of 0.934. The ROC curve value of 0.93 supports its relative underperformance.

Overall, the CNN with Firefly Algorithm outperforms the baseline CNN and the MobileNetV2 model in terms of deepfake detection effectiveness and efficiency. This improvement emphasizes the effectiveness of using optimization approaches to improve model performance in difficult classification problems.

Conclusion

Finally, combining the Firefly Algorithm with the CNN model has shown to be a very successful method for detecting deepfakes. The upgraded CNN model outperformed the baseline CNN and MobileNetV2 models in terms of accuracy, precision, recall, F1-score, and ROC curve value. With an amazing accuracy of 0.9909 and a near-perfect ROC curve value of 0.99, the CNN with Firefly Algorithm performs well in distinguishing Fake and Real pictures. This increase is due to the Firefly Algorithm's capacity to properly tune hyper parameters, which refines the model's performance.

The baseline CNN, although effective, had somewhat lower metrics, with an accuracy of 0.9598 and a ROC curve value of 0.96, suggesting that, while it works well, it falls short of the Firefly-optimized CNN in terms of precision. Despite being a more complex model, MobileNetV2 had the lowest performance metrics of the tested models, highlighting its limitations when compared to the more streamlined and optimized CNN approach.

Overall, the findings show that improving CNN models using approaches such as the Firefly Algorithm may greatly improve their performance in deepfake detection tasks. This shows that such optimization procedures are useful for enhancing classification accuracy and reliability, and they represent a potential area for future study and application in image classification and other related fields.

References

1. Peipeng Y, Xia Z, Fei J, Lu Y. A Survey on Deepfake Video Detection. IET Biometrics. 2021 Apr;10. DOI: 10.1049/bme2.12031.
2. Heidari A, Navimipour N, Dag H, Unal M. Deepfake detection using deep learning methods: A systematic and comprehensive review. Wiley Interdiscip Rev Data Min Knowl Discov; c2023 Nov, 14. DOI: 10.1002/widm.1520.
3. van der Sloot B, Wagensveld Y. Deepfakes: regulatory challenges for the synthetic society. Comput Law Secur Rev. 2022;46:105716. DOI: 10.1016/j.clsr.2022.105716.
4. Rana M, Nobi M, Murali B, Sung A. Deepfake Detection: A Systematic Literature Review. IEEE Access. 2022 Jan;10:1. DOI: 10.1109/ACCESS.2022.3154404.
5. Thing VLL. Deepfake Detection with Deep Learning: Convolutional Neural Networks versus Transformers. In: Proceedings of the 2023 IEEE International Conference on Cyber Security Resilience, CSR 2023; c2023, p. 246-253. DOI: 10.1109/CSR57506.2023.10225004.
6. Guhagarkar N, Desai S, Vaishyampayan S, Save A. Deepfake Detection Techniques: a Review. Int J Res Innov ISSN. 2021;1(4):2581-7280. Available from: www.viva-technology.org/New/IJRI.
7. Sarpate D, Mungi A, Borkar S, Mane S, Shaikh K. A Deep Approach to Deep Fake Detection. Int J Sci Res Sci Eng Technol. 2024;11(2):530-534. DOI: 10.32628/ijsrset2411274.
8. Tolosana R, Vera-Rodriguez R, Fierrez J, Morales A, Ortega-Garcia J. Deepfakes and beyond: A Survey of face manipulation and fake detection. Inf Fusion. 2020;64:131-148. DOI: 10.1016/j.inffus.2020.06.014.
9. Gambín Á, Yazidi A, Vasilakos A, Haugerud H, Djenouri Y. Deepfakes: current and future trends. Artif Intell Rev. 2024 Feb. DOI: 10.1007/s10462-023-10679-x.
10. Gupta G, Raja K, Gupta M, Jan T, Whiteside ST, Prasad M. A Comprehensive Review of DeepFake Detection Using Advanced Machine Learning and Fusion Methods. Electronics. 2024;13(1):1-27. DOI: 10.3390/electronics13010095.
11. Li M, Liu B, Hu Y, Zhang L, Wang S. Deepfake Detection Using Robust Spatial and Temporal Features from Facial Landmarks. In: Proceedings of the 9th International Workshop on Biometrics Forensics, IWBF 2021; 2021. DOI: 10.1109/IWBF50991.2021.9465076.
12. Akhtar Z. Deepfakes Generation and Detection: A Short Survey. J Imaging. 2023;9(1). DOI: 10.3390/jimaging9010018.
13. Wodajo D, Atnafu S. Deepfake Video Detection Using Convolutional Vision Transformer. 2021. Available from: http://arxiv.org/abs/2102.11126.

14. Suratkar S, Kazi F, Sakhalkar M, Abhyankar N, Kshirsagar M. Exposing DeepFakes Using Convolutional Neural Networks and Transfer Learning Approaches. In: 2020 IEEE 17th India Council International Conference (INDICON); 2020. p. 1-8. DOI: 10.1109/INDICON49873.2020.9342252.
15. Ismail A, Elpeltagy M, Zaki M, El-Dahshan KA. Deepfake video detection: YOLO-Face convolution recurrent approach. PeerJ Comput Sci. 2021;7:1-19. DOI: 10.7717/PEERJ-CS.730.
16. Habeeba SMA, Lijiya A, Chacko AM. Detection of deepfakes using visual artifacts and neural network classifier. In: Innovations in Electrical and Electronic Engineering; c2021, p. 411-422.
17. Ge S, Lin F, Li C, Zhang D, Wang W, Zeng D. Deepfake video detection via predictive representation learning. ACM Trans Multimed Comput Commun Appl; c2022 May, 18. DOI: 10.1145/3536426.
18. Liu Y, Wang SL, Zhang JF, Zhang W, Li W. A neural collaborative filtering method for identifying miRNA-disease associations. Neurocomputing. 2021;422:176-185. DOI: 10.1016/j.neucom.2020.09.032.
19. Heo YJ, Choi YJ, Lee YW, Kim BG. Deepfake Detection Scheme Based on Vision Transformer and Distillation; c2021. Available from: http://arxiv.org/abs/2104.01353.
20. Ghosh A, Sufian A, Sultana F, Chakrabarti A, De D. Fundamental concepts of convolutional neural network. In: Recent trends in Advancements of Artificial Intelligence and Internet of Things; c2020, p. 519-567.
21. Mishra B, Shahi TB. Deep learning-based framework for spatiotemporal data fusion: an instance of landsat 8 and sentinel 2 NDVI. J Appl Remote Sens. 2021;15(3):34520.
22. Habeck C, Gazes Y, Razlighi Q, Stern Y. Cortical thickness and its associations with age, total cognition and education across the adult lifespan. PLoS One. 2020;15(3).
23. Chen L, Xu G, Zhang Q, Zhang X. Learning deep representation of imbalanced SCADA data for fault detection of wind turbines. Measurement. 2019;139:370-379.
24. Suresh V, Janik P, Rezmer J, Leonowicz Z. Forecasting solar PV output using convolutional neural networks with a sliding window algorithm. Energies. 2020;13(3):723.
25. Aslam S, Herodotou H, Mohsin SM, Javaid N, Ashraf N, Aslam S. A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids. Renew Sustain Energy Rev; c2021 Mar, 144. DOI: 10.1016/j.rser.2021.110992.