

An evolutionary-convolutional neural network for fake image detection

Retaj Matroud Jasim, Tayseer Salman Atia

Department of Computer Engineering, Al Iraqia University, Baghdad, Iraq

Article Info

Article history:

Received Sep 4, 2022

Revised Oct 14, 2022

Accepted Oct 24, 2022

Keywords:

Convolutional neural network
Generative adversarial networks
Genetic algorithm
Multi-layer perceptron
Random forest
Support vector machine

ABSTRACT

The fast development in deep learning techniques, besides the wide spread of social networks, facilitated fabricating and distributing images and videos without prior knowledge. This paper developed an evolutionary learning algorithm to automatically design a convolutional neural network (CNN) architecture for deepfake detection. Genetic algorithm (GA) based on residual network (ResNet) and densely connected convolutional network (DenseNet) as building block units for feature extraction versus multilayer perceptron (MLP), random forest (RF) and support vector machine (SVM) as classifiers generates different CNN structures. A local search mutation operation proposed to optimize three layers: (batch normalization, activation function, and regularizes). This method has the advantage of working on different datasets without preprocessing. Findings using two datasets evidence the efficiency of the suggested approach where the generated models outperform the state-of-art by increasing 1% in the accuracy; this confirms that intuitive design is the new direction for better generalization.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Retaj Matroud Jasim

Department of Computer Engineering, Al Iraqia University

Saba'a Abkar, Baghdad, Iraq

Email: retaj.m.jasim@students.aliraqia.edu.iq

1. INTRODUCTION

Technology advancement is a double-edged sword. While this technology is being developed to enhance human life. On the one hand, attackers are threatening confidentiality, and causing damage, as was the case with deepfake. The term "Deepfake" was established in 2017 to describe the combination of deep learning with fake information that enables anybody to swap an individual's face with another's, including expressions. And generate realistic photographic false photographs or videos [1]. At the moment, fake detection is regarded as a critical security challenge. Because manufacturing has historically been constrained due to a lack of robust editing tools, domain expertise, and a time-consuming and challenging procedure [1]. It is getting easier to automatically generate fake faces or modify the face of a single person in a video or image. The advancement of deeplearning methods such as autoencoders (AE) [2] and generative adversarial networks (GAN) [3] has been facilitated by the availability of large-scale public data. As well as open software and applications that work on a variety of devices, such as PDAs. Some examples of different GAN generated synthetic faces seen in Figure 1 [4]. The statistics of increased deepfake images generated by different GANs are shown in Figure 2 [5], which Figure 2(a) shown the real images used, and Figure 2(b) shown number of face image generated by different GAN network. Thus, targeting individuals is no longer restricted to celebrities and those with expertise but has expanded to encompass the general population for any reason. As a result, it is necessary to design a more powerful and general fake image detector. To tackle these more complex and realistic attacks, researchers explored and presented a deep learning model for detecting fake images. The authors used CNN to develop models to detect deepfakes using a particular type of dataset created by GAN that requires preprocessing before inputting the detection model.

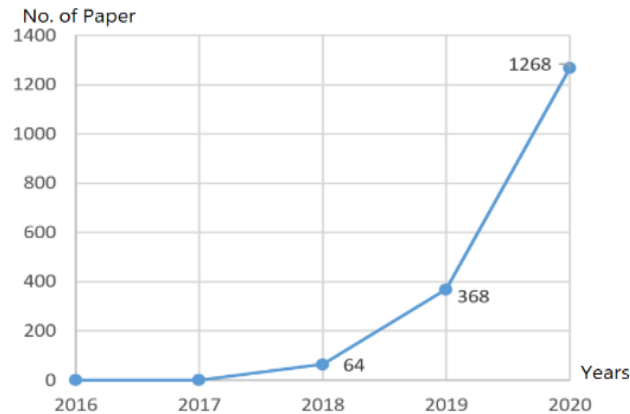


Figure 1. The number of papers in deepfake in years from 2016 to 2020 [4]

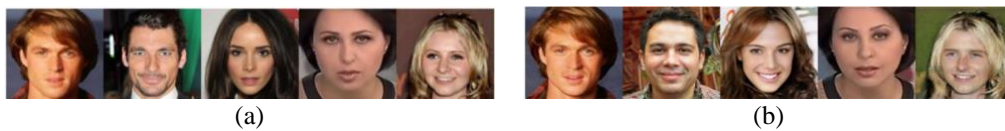


Figure 2. Different face image in (a) real face photo with variable resolution and (b) fake images made using Glow, StyleGAN, PGGAN, Face2Face, and StarGAN, from left to right [5]

Cozzolino *et al.* [6] a certain kind of residual-based descriptor might be seen as a primary convolutional neural network (CNN) with constrained connectivity. After that, may fine-tune and release some of the restrictions on the network on small sample size. He gained a significant increase in performance compared to a traditional detector. It used datasets image taken from nine sources, five from smartphones, and another four taken from a camera. These images are generated by five manipulations (median filtering, gaussian blurring, additive white gaussian noise (AWGN), resizing, and joint photographic experts group (JPEG) compression). The proposed CNN outperforms structural risk minimization (SRM)+support vector machine (SVM). However, advancing state of the art is slower and more expensive.

Afchar *et al.* [7] MesoNet, a convolutional neural network was designed to differentiate between authentic and Deepfake-modification faces. The system consists of two inception units: Meso-4 and MesoInception-4, and two convolution layers interconnected by max-pooling segments. Experiments indicate that the system has a 98% percent average identification rate for Deepfake films and a 95 % average detection rate for Face2Face videos under actual internet diffusion settings.

Tariq *et al.* [8] evaluated the use of different CNN architectures including the VGG16 [9], the ResNet [10], and the XceptionNet [11], to get real world face images. They used the CelebA database [12] for forgery images. Two ways were investigated: i) machine approaches based on GAN, namely ProGAN [13] and ii) manual approaches based on Adobe Photoshop CS6. Which comprised cosmetics, eyeglasses, sunglasses, hair, and hats. Nguyen *et al.* [14] suggested a deep CNN-based methodology for detecting Deepfake using a capsule network (CN). It cascades two CNNs: LNet and ANet, concurrently fine-tuned with attribute labels but differentially pre-trained. ANet is pre-trained by enormous facial identities for feature prediction. Whereas LNet is pre-trained by, large object classes for face identification. Additionally, it identifies replay attacks and computer-generated images.

Rössler *et al.* [15] proposed an automated benchmark for facial manipulation detection, that is based on deepfakes [16], face2face [17], faceswap [18], and neural textures as prominent representations of facial manipulations, at random compression levels and sizes. Conducted a comprehensive evaluation of data-driven forgery detectors. He established that extra domain-specific information increases the accuracy of forgery detection, even in the face of significant compression. They examined their proposed approach using actual faces from the CelebA-HQ [19] and flickr-faces-HQ (FFHQ) [20] databases, as well as synthetic faces created using InterFaceGAN [21] and StyleGAN [20], achieving an overall accuracy of 84.7% when using the FaceNet model.

Dolhansky *et al.* [22] established a baseline using three simple detection systems: i) a tiny CNN model with six convolutional layers and one fully connected layer for detecting low-level image modifications, ii) an

XceptionNet model trained only on face images, and iii) an XceptionNet model trained on the complete picture. When just the face image was analyzed, the detection system based on XceptionNet generated the best results, with a precision of 93.0%.

Rana *et al.* [23] designed a CNN that uses a multi-task learning strategy to recognize altered images and videos, and simultaneously identify the modified areas for each query. Sun *et al.* [24] explained how to evolve CNN architectures using a genetic algorithm automatically, and he demonstrated how its suggested technique is self-contained in its approach to CNN architecture design. Specifically, no pre- or post-processing in terms of CNNs is required.

Guarnera *et al.* [25] focusing on analyzing DeepFakes of human faces to develop a new detection method capable of detecting forensic traces hidden in images, a type of fingerprint during the image creation. The approach suggested employs an expectation-maximization (EM) algorithm, to extract a set of local characteristics tailored to model the underlying convolutional generative process. Guarnera *et al.* [26] they proposed a preliminary idea for combating deepfake images. By analyzing peculiarities in the frequency domain. Evaluation forensic scientists conducted samples of deepfake images using one of the most excellent well-known images. Forensic analysis applications, "Amplified Authenticate," were used to determine when it is capable of determining if a GAN-generated image. Seems realistic and contains anomalies focusing on what comes out of combining two technological processes: StarGAN [27] and StyleGAN [20].

Yang *et al.* [29] presented a forensic intelligence approach for detecting deepfake. They began by identifying the minor texture changes between actual and fabricated images using visual saliency, demonstrating the faces' texture. To emphasize this distinction, they used a guided filter using a feature map as a decision guided, to improve texture artifacts produced by post-processing, and to highlight probably forged characteristics. Guo *et al.* [5] developed an adaptive manipulating traces extraction network (AMTEN) that extracts modification traces, and a pre-processor to eliminate image content and show alteration traces. AMTEN employs an adaptive convolution layer to predict alteration traces in an image. AMTENnet, a fake face detector, is created by combining AMTEN with CNN. The experimental findings demonstrate that the suggested AMTEN accomplishes desired preprocessing. When recognizing faked facial photos created by different FIM methods. AMTENnet attains an average precision of 98.52%. Another model preprocesses data before sending it to the model via a primary network, and then trains naive classifiers to distinguish between authentic and GAN-generated images. Also provided AMTENnet and found a difference, which is constructed by adding AMTEN, and Experimental results demonstrate that the proposed AMTEN accomplishes desirable preprocessing.

Kaur *et al.* [29] the researchers calculated three features: i) features based on the photos content, ii) temporal features based on the frames at which the photos were shared, and iii) social context features based on the individuals who shared the images. These attributes were fed into machine learning algorithms to determine whether data is genuine. The dataset was used includes 810 K and 34 K photographs, respectively, posted on WhatsApp by 63 K and 17 K WhatsApp users in India and Brazil. Prediction for identifying false images was shown, to be much enhanced by including temporal and social context information. In contrast to expectations, they discovered that CNN-derived image content features utilizing raw pictures performed worse, than socio-temporal features but were still superior to random prediction. Using ensemble learning, the results of Support vector machines, random forest (RF), and logistic regression (LR) were combined with the help of socio-temporal characteristics to create the best model possible.

Chakraborty *et al.* [30] the proposed technology could ascertain whether, the participants were paying attention and whether or not they were physically there. This system engages with the user, creates awareness, and begins verbal or visual communication based on the user's degree of visual focus of attention (VFOA). There is a near-perfect correlation between validation and testing accuracy of 99.24% and 99.43%, respectively, thanks to the presented ML methods.

Hsu *et al.* [31] they suggested utilizing deep learning and contrastive loss to spot false photos. To begin, numerous cutting-edge GANs used to create synthetic-natural picture pairings. The simplified DenseNet then expanded into a two-streamed network architecture to accept paired data. Pairwise learning then used to train the proposed shared fake feature network to recognize key differences between the two types of photos. In order to determine if an input picture is genuine or fake, a classifier was append to the suggested shared fake feature network.

Faruqui *et al.* [32] the optimized method implemented in three steps: the first step developed an algorithm to detect faces in 99 different resolutions. In the second step, generate a curve of characteristics to the accuracy. In the third step generated nine different mathematical models using curve fitting [33]. Despite the numerous works on the same subject, the previous studies focus on the previous detection, preprocessing, working with datasets generated by a particular kind of generative adversarial networks (GAN), and even cooperating with an expert in image processing and artificial intelligence. We propose a CNN network based on ResNet and DenseNet blocks, and we will use an evolutionary technique to evolve and fine-tune the CNN's topology. The presented method is self-contained in its approach to designing CNN architectures. There is no

necessity for pre-or post-processing with CNNs. Additionally, the recommended technique makes no assumptions about the user's prior knowledge of CNNs, the study's subject, or the evolutionary algorithm. The suggested technique is evaluated using datasets generated using various kinds of GANs.

2. SUGGESTED ALGORITHM

The proposed evolutionary CNN employs evolutionary learning techniques to achieve the goal of this work (by providing generalizations with a low degree of complexity). This technique generates models automatically and fine-tunes them (in terms of their number of layers, parameters, and classifier) to get a model with the best performance that works on different datasets. Figure 3 shows a block diagram that demonstrates the proposed evolutionary-CNN for fake image detection.

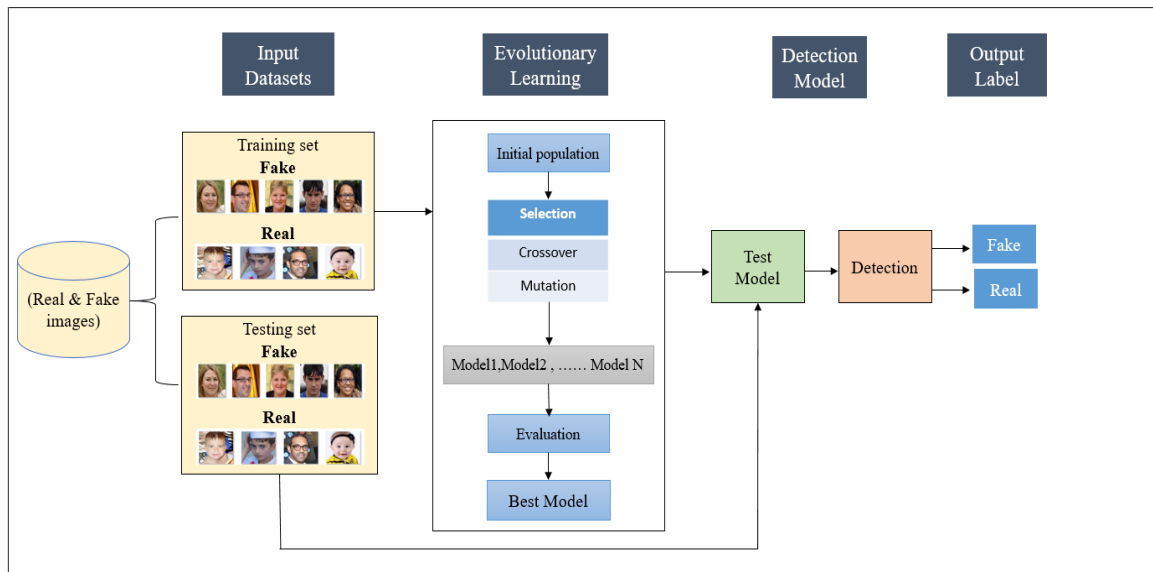


Figure 3. The suggested detection models

2.1. Population initialization

Chromosome (solution) in GA are constructed using four blocks of pertained networks, 2-ResNet blocks and 2-DenseNet blocks. These networks are used since they reduce the adverse effects of the gradient vanishing problems depending on which architecture can effectively train the hierarchical representations of the input data and improve the final detection performance (these blocks are shown in Figure 4). A fully connected layer (flatten) for feature extraction that is either stacked or ensemble, followed by a machine learning or deep learning classifier, namely multilayer perceptron (MLP), SVM, and RF. The MLP classifier used in this thesis has one hidden layer with 500 nodes in the dense layer and the sigmoid binary output activation function for the output layer. The nonlinear SVM was used with the RBF kernel, as well as the RF (25 trees). Finally, to initialize the first population, a permutation procedure with three vectors (feature extraction, classifiers, and learning) is used, as illustrate in Algorithm 1. The character encoding method is used to encode the necessary information about the CNN architecture. The letters R stand for Resnet, D for DenseNet, M for MLP, S for SVM, F for RF, E for Ensemble, and K for Stack. A fixed chromosome length of six gens (cells) proposed to represent three important design principles: feature extraction, classifiers, and learning, as illustrated in Figure 4.

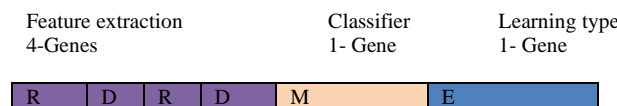


Figure 4. Chromosome structure

Algorithm 1. Initiate population

I/p: blocks matrix B, Classifier matrix C, learning matrix L, and number of solution n.
O/p: intial population P0.
 1. **While** P0 < n **Do**
 2. B1 ← entry from B;
 3. CL1 ← entry from CL;
 4. L1 ← entry from L;
 5. Chromosome ← B1 ∪ CL1 ∪ L1
 6. Add Chromosome to P0 population
 7. **End**
 8. Return P0

2.2. Crossover

Single point crossover is applied to exchange the information between two parents (classifier and learning), as shown in Figure 5. This is the simplest type. The single crossover location is selected at random and the parts of the two parents after the crossover position are exchanged to form two offsprings, Algorithm 2 illustrate the crossover operation.



Figure 5. The crossover operation

Algorithm 2. Crossover operation

I/p: Two Parent individual P1 and P2 selected by the binary tournament selection, crossover probability Cp
O/p: Two offspring
 1. X ← uniformly generate a number from [0, 1];
 2. **If** X < Cp **then**
 3. Set crossover a position at P1 and P2 respectively;
 4. Q1 ← combine the classifier part and leaning type from P1 with feature extraction from P2;
 5. Q2 ← combine the classifier and learning type from P2 with feature extraction from P1;
 6. **Else**
 7. Q1 ← P1;
 8. Q2 ← P2;
 9. **End**
 10. Return Q1 and Q2.

2.3. Mutation

In traditional GAs, the mutation operation is responsible for the global search, testing the search space for desirable performance in a single path generated with a fixed probability and the allowed mutation. Convergence may be achieved too soon if navigation of a search space is conducted in just one direction. A local search mutation is suggested to solve this issue by allowing algorithm to generate different pathways from a central starting point, as shown in Figure 6.

The proposed local search mutation technique is applied to optimize activation functions, Batch normalization, and regularization layers DenseNet and Resnet blocks. Optimizing one or more blocks simultaneously is possible. One or all of the necessary layers may also be subjected to optimization. To achieve this objective, a binary representation is used. For feature extraction blocks, there are four blocks, so their binary patterns are from 0000 to 1111, with bit1 representing block1, bit2 representing block2, and so on. if

the binary sequence in the block is 0011 for instance, then blocks 1 and 2 should be optimized. The same principle is used for the layer number, equivalent to three. These parameters range from 000 to 111, each bit denoting a different aspect of the activation function, normalization, and regularization. To keep things simple in Algorithm 3, integer representation is used. the various block numbers, with values from 0 to 15 (step 1) for the needed block parameters and 0 to 7 (step 2) for the required parameter values (step 2). Every time the method is run, it chooses a random number and configuration for the blocks. Mutation (steps 5 to step 27). (steps 5 to step 27). Then, the binary representation of the integer is obtained. The last step is to analyze these representations and conduct a local search. If the offspring are not exposed to mutation, it will continue to look the same.

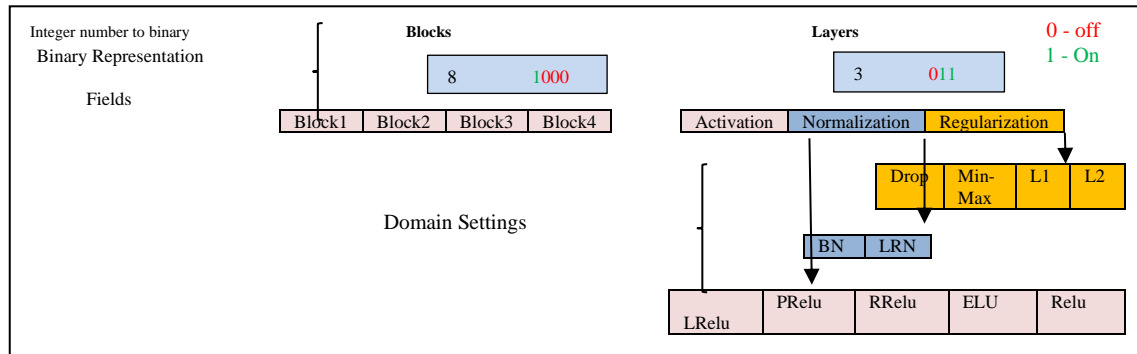


Figure 6. Mutation encoding

Algorithm 3. Mutation

I/p: The offspring Q , Activation, Normalization, Regularization

O/p: The mutation local population M_t .

```

1.  $X \leftarrow$  Uniformly generate a number from  $\{1, 16\}$ ;
2.  $Y \leftarrow$  Uniformly generate a number from  $\{1, 7\}$ ;
3.  $X_{bin} \leftarrow$  four bit binary of  $X$ ;
4.  $Y_{bin} \leftarrow$  three bit binary of  $Y$ ;
5. For each bit in  $X_{bin}$ 
6.   IF bit=1 then
7.     IF  $bit_1$  in  $Y_{bin}=1$  then
8.       For each A in Activation function
9.         model= New model with activation function at block;
10.        Mt=Mt  $\cup$  model;
11.      end
12.    end
13.    IF  $bit_2$  in  $Y_{bin} = 1$  then
14.      For each N in Normalization
15.        model= New model with Normalization in block;
16.        Mt=Mt  $\cup$  model;
17.      end
18.    end
19.  end
20.  Return  $M_t$ 
21. end

```

3. EXPERIMENT DESIGN

3.1. Benchmark data sets

These two types of datasets selected to achieve our work's goal for two reasons: its can work with datasets generated by different GAN networks, and these datasets have been worked on previously and produced results. Moreover, we compare our model's performance to that of existing models. They vary in size (large and medium) to demonstrate that our model can handle varying data.

3.1.1. HFF

The hybrid fake face (HFF) dataset [5] includes eight facial images. Three kinds of facial images are pick a random from 3 free datasets for use with actual face images. Face with poor resolution photos are from

CelebA [20], very realistic images of people's faces are from CelebA-HQ [19], also, video stills of people's heads are from FaceForensics. Thus, facial photos on the Internet are replicated as realistically as feasible. PGGAN and StyleGAN for detect manipulation, Face2Face and Glow for face expression modification, and StarGAN for face features transmitted are chosen to generate false face pictures. Since StarGAN may transfer facial qualities such as hair color (black, blond, brown), and age (young or elderly) to other fields, StarGAN manipulates 5 kinds of face features. It has been stated that face photos with varied qualities produced by the same GAN have the same artifacts or fingerprints [34].

3.1.2. 140 K

We employed 140 K [35] authentic and synthetic images in a dataset for our studies. This dataset contains 70 K authentic images taken from the flickr-faces-HQ (FFHQ) [18] data gathered by NVidia, as well as 70 K synthetic images selected from the one million synthetic images dataset (created by StyleGAN); Figure 7 shows this network generates some images. The database is divided into 100k training sets, 20 K test sets, and 20 K validation.

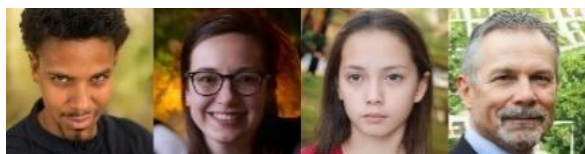


Figure 7. Fake faces generated with StyleGAN [36]

4. EXPERIMENTAL SETUP

4.1. Evaluation metrics

Accuracy (ACC) and error rate (ERR) were considered in this work to evaluate the model's efficiency. Accuracy helps in finding out errors in the measurement values of the models. The error rate is directly proportional to the accuracy. The ACC is derived using the ratio of correct predictions to total observations in the dataset (shown in (1)). ERR is calculated by dividing the total number of inaccurate predictions by the total data set (shown in (2)). Where (TP+TN) the true prediction, (FP+FN) the false prediction, (P+N) the total number of data sets.

$$ACC = (TP + TN)/(P + N) \quad (1)$$

$$ERR = (FP + FN)/(P + N) \quad (2)$$

4.2. Experiment results and discussion

It used this experiment to detect forged images, and worked on two data sets (140 A and HHH). When experimenting with these datasets, the 'sigmoid' output activation function is used when an MLP classifier is used. The evaluation results for the trial data set are shown in Table 1; We get her best models in binary detection. The first column in the table ("Model") relates to the structure of the model in which we train datasets (R1, R2) to represent two blocks of ResNet50, while (D1, D2) represent two blocks of DenseNet121. The second column ("building blocks") is the block that creates this structure. The structural elements that build this architecture, the number of layers (Conv2D) and the assembly layer (the pooling intermediate layer) help reduce a number of parameters and thus increase processing. It performs downsampling using an average or median of the width and height inputs. The third column ("parameters") indicate the settings of the model: N represents the Normalization which has two value (Batch normalization, local response normalization), AF (Activation function), which have (ReLU, LeakyReLU, PReLU), Ks represent kernel size, a relatively small number matrix. When applied to the input images, the compression between the filter value and the pixels of the input image generates new feature maps. "St" denotes "Stride." Filter indicates the number of filters that may be used, and o/p AF represents the output activation function ("sigmoid"). The last parameter is the optimizer ("Adam"), which has the default-learning rate (0.001) and momentum (0.9). The fifth column indicates the kind of classifier used in this model; we use three different types (MLP, SVM, and RF. The sixth column ("model type") indicates how these blocks may be interconnected (parallel or sequential). The seventh and eighth columns represent the datasets employed and the model's accuracy on these datasets, respectively.

Table 1 shows three of the best models generated by the proposed GA. What is interesting about the data in the table is that: High accuracy for all datasets was found when an MPL was used as a classifier, which is expected since it correlates the feature non-linearly to generate all possible patterns. No significant difference

in accuracy was found between sequential or parallel models in SVM and RF for the datasets HFF. On the other hand, unexpected results were found for 140k datasets. These findings indicate that the classifiers are unaffected by model type and relate to the size of the datasets, which is related to the number of extracted attributes, where the classifier accuracy decreases as the number of extracted features increases. We can infer a good correlation between the activation function and MLP; the ML classifier is unsuitable for ensembles with deep learning. This method show best performance in detection compared with other one. The accuracy of CNN of the first model on the test dataset results shown in Figure 8, which Figure 8(a) illustrates the accuracy ROC of the HFF dataset, while Figure 8(b) illustrates the accuracy ROC of 140 K datasets. Figure 9 represent the loss in both two datasets, which Figure 9(a) illustrates the loss in HFF dataset, while Figure 9(b) illustrates the loss in 140 K dataset.

Table 1. Best model results

| Model | Building blocks | Parameter | Dataset | Accuracy (%) |
|--------------------|---|---|---------|--------------|
| R1-D1-R2-D2 | R1 | N (Batch normalization, local response normalization) AF (leaky relu) Ks (7x7) St (2,2) Filter (64) | 140 K | 99.16 |
| | Conv2D $\left[\begin{matrix} (1 \times 1, 64) \\ (3 \times 3, 64) \\ (1 \times 1, 56) \end{matrix} \right] \times 3$ | | HFF | 98.45 |
| | R2 | Ks (2x2) St (2x2) Filter1(56) Filter2(28) | | |
| | Conv2D $\left[\begin{matrix} (1, 1, 128) \\ (3, 3, 128) \\ (1, 1, 512) \end{matrix} \right] \times 3$ Pooling (average pooling) | | | |
| R1-R1-D2-D2 | D1 | Pad (zero padding) Reg (dropout=0.5) o/p AF (sigmoid) classifier MLP model type Sequential | | |
| | Conv2D $\left[\begin{matrix} (1 \times 1) \\ (3 \times 3) \end{matrix} \right] \times 6$ | | | |
| | D2 | | | |
| | Conv2D $\left[\begin{matrix} (1 \times 1) \\ (3 \times 3) \end{matrix} \right] \times 12$ Pooling (average pooling) | | | |
| R1-R1-D2-D2 | Same settings above | N (Batch normalization local response normalization) AF (leaky relu) Ks (7x7) St (2,2) Filter(64) Ks (2x2) St (2x2) Filter1(56) Filter2(28) Pad (zero padding) Reg (dropout=0.5) classifier SVM model type parallel | 140 K | 67.34 |
| | | | HFF | 87.60 |
| | | | | |
| | | | | |
| D1-R1-D2-R2 | same | Same | 140 K | 67.74 |
| | | Classifier SVM | HFF | 87.54 |
| | | Classifier RF | 140 K | 63.45 |
| | | | HFF | 85.34 |

4.3. Evaluation with related works

To evaluate further the accuracy of the evolutionary CNN models. The comparisons are compared among proposed evolutionary CNN and some works. Note that Handcrafted-AMNET Guo *et al.* [5] and LBP Wang *et al.* [35] are developed for other forensics tasks, so We need them to be divergent for our forensics investigation. We use three datasets: the HFF, and 140 K. Table 2 reports the accuracy of detection on two datasets. We can see that evolutionary CNN can obtain the best results in binary classification tasks. Evolutionary CNN outperforms all other architectures on the 140 K and HFF data sets regarding detection error. specifically, evolutionary-CNN obtains a detection error rate of 0.41% for HFF and 0.68% for 140 K.

In addition, the detection error rate is lower than that of LBP and AMNET. On 140 K and HHF, evolutionary-detection CNN's error is 5.15% less than AMNET and LBP. Furthermore, the comparison demonstrates that the suggested method obtains the highest level of performance among the architectures to which it belongs.

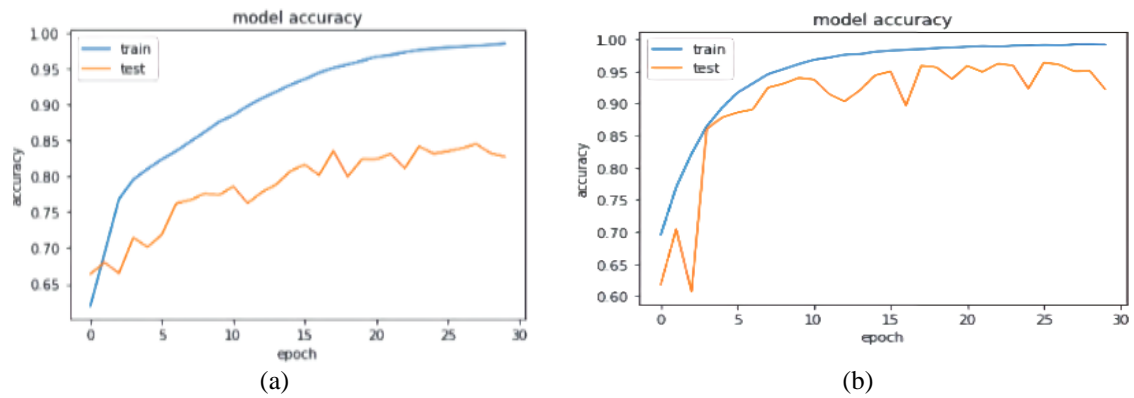


Figure 8. The model accuracy of two datasets

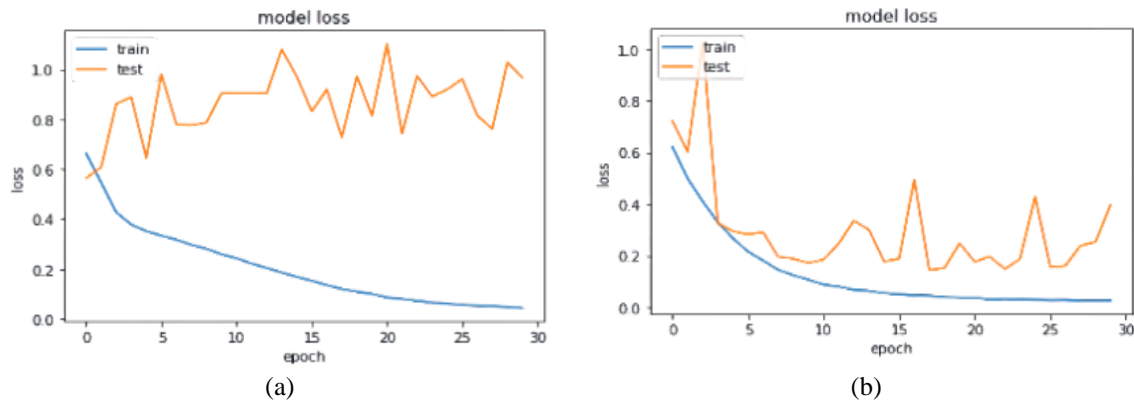


Figure 9. The loss results of two datasets, (a) HHF and (b) 140 K

Table 2. Comparison of proposed model with LBPNet and AMTENnet

| Model | Datasets | Accuracy (%) | Proposed model Accuracy (%) |
|----------|----------|--------------|-----------------------------|
| LBP-Net | HHF | 98.21 | 98.45 |
| AMTENnet | 140k | 98.52 | 99.75 |

5. CONCLUSION

This work aims to create and evolve a CNN architecture design algorithm utilizing genetic algorithms capable of automatically creating and evolving the optimum CNN architecture with minimal resources. The proposed architecture has three phases: the first is the CNN network. The second phase is the genetic algorithm that CNN feeds into to choose the best model from many models. The third phase represents the classifier connected to CNN after choosing the best CNN model to detect whether the image is real or fake. This study deals with three types of classifiers (MLP, SVM, and RF), each giving different results. The suggested encoding approach based on Resnet and Desnet blocks with a constant-length representation has accomplished this objective. It provides a multipoint crossover operator for the chromosome and a local search mutation operator for the best parameter settings. The offered operator of crossovers and the developed mutation operators give the suggested effective local and global search algorithm capabilities. They enable it to locate the optimal CNN designs. On the HHF and 140 K fake image detection datasets. The proposed technique yields the best results. The model performs well on various manipulations and can recognize fake images created by various kinds of GANs; unlike other models, it does not need pretreatment and postprocessing. Our subsequent work will be capable of spotting fraudulent video content.




REFERENCES

- [1] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, Dec. 2020, doi: 10.1016/j.inffus.2020.06.014.
- [2] R. Lopez, J. Regier, M. I. Jordan, and N. Yosef, "Information constraints on auto-encoding variational Bayes," *Advances in Neural Information Processing Systems*, vol. 2018-December, pp. 6114–6125, 2018.
- [3] I. Goodfellow et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.
- [4] H. A. Khalil and S. A. Maged, "Deepfakes creation and detection using deep learning," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, May 2021, pp. 1–4, doi: 10.1109/MIUCC52538.2021.9447642.
- [5] Z. Guo, G. Yang, J. Chen, and X. Sun, "Fake face detection via adaptive manipulation traces extraction network," *Computer Vision and Image Understanding*, vol. 204, p. 103170, Mar. 2021, doi: 10.1016/j.cviu.2021.103170.
- [6] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *IH and MMSEC 2017 - Proceedings of the 2017 ACM Workshop on Information Hiding and Multimedia Security*, Jun. 2017, pp. 159–164, doi: 10.1145/3082031.3083247.
- [7] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *10th IEEE International Workshop on Information Forensics and Security, WIFS 2018*, Dec. 2019, pp. 1–7, doi: 10.1109/WIFS.2018.8630761.
- [8] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo, "Detecting both machine and human created fake face images in the wild," in *Proceedings of the ACM Conference on Computer and Communications Security*, Jan. 2018, pp. 81–87, doi: 10.1145/3267357.3267367.
- [9] J. Tao, Y. Gu, J. Z. Sun, Y. Bie, and H. Wang, "Research on vgg16 convolutional neural network feature classification algorithm based on transfer learning," in *CISS 2021 - 2nd China International SAR Symposium*, Nov. 2021, pp. 1–3, doi: 10.23919/CISS51089.2021.9652277.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [11] B. Wang, Y. Li, X. Wu, Y. Ma, Z. Song, and M. Wu, "Face forgery detection based on the improved siamese network," *Security and Communication Networks*, vol. 2022, pp. 1–13, Feb. 2022, doi: 10.1155/2022/5169873.
- [12] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, vol. 2015 Inter, pp. 3730–3738, doi: 10.1109/ICCV.2015.425.
- [13] Z. Zhang, M. Li, and J. Yu, "D2PGGAN: two discriminators used in progressive growing of GANs," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, vol. 2019-May, pp. 3177–3181, doi: 10.1109/ICASSP.2019.8683262.
- [14] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: using capsule networks to detect forged images and videos," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, May 2019, vol. 2019-May, pp. 2307–2311, doi: 10.1109/ICASSP.2019.8682602.
- [15] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2019, vol. 2019-October, pp. 1–11, doi: 10.1109/ICCV.2019.00009.
- [16] M. Westerlund, "The Emergence of Deepfake Technology: A Review," *Technology innovation management review*, vol. 9, no. 11, pp. 39–52, Jan. 2019, doi: 10.22215/timreview/1282.
- [17] A. Groshev, A. Maltseva, D. Chesakov, A. Kuznetsov, and D. Dimitrov, "GHOST-a new face swap approach for image and video domains," *IEEE Access*, vol. 10, pp. 83452–83462, 2022, doi: 10.1109/ACCESS.2022.3196668.
- [18] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image Synthesis using Neural Textures," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, Aug. 2019, doi: 10.1145/3306346.3323035.
- [19] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2018.
- [20] Y. Shen, C. Yang, X. Tang, and B. Zhou, "InterFaceGAN: Interpreting the disentangled face representation learned by GANs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2004–2018, Apr. 2022, doi: 10.1109/TPAMI.2020.3034267.
- [21] Y. Shen, C. Yang, X. Tang, and B. Zhou, "InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2004–2018, Apr. 2022, doi: 10.1109/TPAMI.2020.3034267.
- [22] B. Dolhansky et al., "The deepfake detection challenge (DFDC) dataset," *arXiv preprint*, Jun. 2020, [Online]. Available: <http://arxiv.org/abs/2006.07397>.
- [23] M. S. Rana and A. H. Sung, "DeepfakeStack: A deep ensemble-based learning technique for deepfake detection," in *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, Aug. 2020, pp. 70–75, doi: 10.1109/CSCloud-EdgeCom49738.2020.00021.
- [24] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Completely automated CNN architecture design based on blocks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1242–1254, Apr. 2020, doi: 10.1109/TNNLS.2019.2919608.
- [25] L. Guarnera, O. Giudice, and S. Battiato, "DeepFake detection by analyzing convolutional traces," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, vol. 2020-June, pp. 2841–2850, doi: 10.1109/CVPRW50498.2020.00341.
- [26] L. Guarnera, O. Giudice, C. Nastasi, and S. Battiato, "Preliminary forensics analysis of deepfake images," in *12th AEIT International Annual Conference, AEIT 2020*, Sep. 2020, pp. 1–6, doi: 10.23919/AEIT50178.2020.9241108.
- [27] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: unified generative adversarial networks for multi-domain image-to-image translation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8789–8797, doi: 10.1109/CVPR.2018.00916.
- [28] J. Yang, S. Xiao, A. Li, G. Lan, and H. Wang, "Detecting fake images by identifying potential texture difference," *Future Generation Computer Systems*, vol. 125, pp. 127–135, Dec. 2021, doi: 10.1016/j.future.2021.06.043.
- [29] M. Kaur, P. Daryani, M. Varshney, and R. Kaushal, "Detection of fake images on WhatsApp using socio-temporal features," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 58, Dec. 2022, doi: 10.1007/s13278-022-00883-y.
- [30] P. Chakraborty, S. Ahmed, M. A. Yousuf, A. Azad, S. A. Alyami, and M. A. Moni, "A human-robot interaction system calculating visual focus of human's attention level," *IEEE Access*, vol. 9, pp. 93409–93421, 2021, doi: 10.1109/ACCESS.2021.3091642.




- [31] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, "Deep fake image detection based on pairwise learning," *Applied Sciences*, vol. 10, no. 1, p. 370, Jan. 2020, doi: 10.3390/app10010370.
- [32] N. Faruqui and M. A. Yousuf, "Performance-accuracy optimization of face detection in human machine interaction," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, Sep. 2019, pp. 38–43, doi: 10.1109/ICAEE48663.2019.8975661.
- [33] A. Gelman and A. Vehtari, "What are the most important statistical ideas of the past 50 Years?," *Journal of the American Statistical Association*, vol. 116, no. 536, pp. 2087–2097, 2021, doi: 10.1080/01621459.2021.1938081.
- [34] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in GAN fake images," in *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec. 2019, pp. 1–6, doi: 10.1109/WIFS47025.2019.9035107.
- [35] 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)*, Sep. 2021, vol. 2021-Sept, pp. 3917–3921, doi: 10.1109/ICIP42928.2021.9506460.

BIOGRAPHIES OF AUTHORS



Retaj Matroud Jasim    was born in kut, Iraq, in 1995. She received a B.Sc. degree in computer engineering from Al-Imam Al kadum university Collage, Wasit, 2017. She is currently studying M.Sc. in computer engineering at Al Iraqia University College of engineering, Iraq. She can be contacted me at the e-mail: matroudretaj@gmail.com.



Tayseer Salman Atia    is a professor at the department of computer engineering, Al Iraqia University, Iraq. Where she has been a faculty member since 2012. From 2013-2014 she was the head of the computer-engineering department. From 2014-2015 she was the dean's assistant for scientific affairs. Tayseer graduated with the first class B.Sc. degree in computer science in 2004 and a M.Sc. in data security in 2007 from the university of Technology, Iraq. She completed her Ph.D. in computer science from Al Mosul University, Iraq. Her research interests are data security and artificial intelligence, especially computational intelligence techniques. She can be contacted at e-mail: tayseer.salman@aliraqia.edu.iq.