

Article

Not peer-reviewed version

---

# Improved Gaussian Filter Deep Learning Based General Adversarial Network Artificial Humming Bird Optimization Classifier for Hand Written Personality Prediction

---

[PUTTASWAMY B S](#)<sup>\*</sup> and Thillaiarasu N

Posted Date: 24 August 2024

doi: 10.20944/preprints202408.1684.v1

Keywords: kernel function; eigenvectors; covariance matrix; exploration and exploitation; discriminator; transfer function; generator



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

# Improved Gaussian Filter Deep Learning Based General Adversarial Network Artificial Humming Bird Optimization Classifier for Hand Written Personality Prediction

Puttaswamy B S <sup>1,2,\*</sup> and Thillaiarasu. N <sup>3</sup>

<sup>1</sup> Research Scholar, School of Computing and Information Technology, REVA University, Bengaluru, India.

<sup>2</sup> Assistant Professor, P E S College of Engineering, Mandya, India

<sup>3</sup> Associate Professor, School of Computing and Information Technology, REVA University, Bengaluru, India

\* Correspondence: bsputtaswamy2012@gmail.com;

**Abstract:** Deep learning-based handwriting personality prediction uses neural networks (CNNs) to consider and predict personality characters from handwritten text to improve applications such as adaptive technology and automated data access. This method requires high computational resources and widely classified data, and its accuracy may suffer from signature variation and overfitting. This study introduces a novel optimized deep learning for Hand Written Personality Prediction. First the input data is collected from given dataset and an enhanced Gaussian filter is used in pre-processing to improve image quality by reducing noise and increasing contrast. After pre-processing, features are extracted by PCA-based normalized GIST, which standardizes the features to improve image representation. These extracted features are generated using an adaptive horse herd optimization algorithm to select the most important features. Based on selected features Improved Generative Adversarial Networks with artificial hummingbird optimization (IGAN\_AHb) enables fast and efficient convergence of GANs. This reduce the mode collapse and ensures training with consistently fine-tuned parameters. The selected features are considered by characters such as openness, conscientiousness, extroversion, agreeableness and neuroticism, and finally classified using the IGAN\_AHb classifier. In the result section, the proposed model is compared with various models with the matrix precision, recall, F1-score. The proposed model attained the value of 97.3% of precision and a recall rate of 96%, respectively. By comparing with other models, the proposed model attained highest values.

**Keywords:** kernel function; eigenvectors; covariance matrix; exploration and exploitation; discriminator; transfer function; generator

## 1. Introduction

In today's digital age, information processing is heavily influenced by handwriting recognition. Paper contains a lot of information, and digital file processing is cheaper than processing traditional paper files [1]. The aims of handwritten characters are changed into machine-readable formats by handwriting recognition systems [2,3]. Applications of these systems contain vehicle license plate recognition, sorting of postal letters, check truncation system (CTS) scanning, preservation of historic documents in archaeology and automation of old documents in libraries and banks [4,5]. All these areas deal with large databases, which need high recognition accuracy, low computational complexity, and consistent performance from the recognition system. A deep neural structure has been shown to be more promising than shallow neural structures [6]. The field of deep learning continues to advance with deviations such as auto encoders, convolutional neural networks (CNNs), recurrent neural networks (RNNs), recurrent neural networks, deep belief networks, and deep Boltzmann machines. Introduced here is a convolutional neural network (CNN), a specific type of

deep neural network with wide applications in image classification, object recognition, recommender systems, signal processing, natural language processing, computer vision, and face recognition [7]. CNNs are more useful than their predecessors, multilayer perceptron (MLP), because they can automatically detect salient features of an object (such as an image, handwritten text, or face) without human supervision or involvement. The high performance of hierarchical feature learning results in a very efficient CNN [8].

A convolutional neural network (CNN) is a basic variant of the multilayer perceptron (MLP) network first used in 1980. Human inspired computing at CNN the brain, Humans observe or recognize objects by sight, and humans teach children to understand objects by showing them hundreds of pictures of those objects [9]. Similarly, CNN works in this manner and is recognized for its ability to analyze visual images. CNN architectures contain GoogLeNet (22 layers), AlexNet (8 layers), VGG (16-19 layers) and ResNet (152 layers). Adding the feature extraction and classification steps with CNNs needs minimal pre-processing and feature extraction efforts [10]. Also, CNN automatically extract interconnected features and small amount of training data can provide high recognition accuracy. The gathering design details and no previous knowledge of features is required [11]. The main advantage of using a CNN model is the use of topological information available in the input, which leads to better recognition results. In addition, the recognition results of the CNN model are independent of the rotation and translation of the input images. In contrast, MLP models do not use complete topological knowledge of the inputs. Also, for complex problems, MLPs are considered unsuitable, and their scaling for high-resolution images is delayed by the full interconnection between nodes, which is popularly called the curse of dimensionality.

In the last few years, the CNN model has been widely applied to handwritten digits. Authentication from the MNIST benchmark database. Some researchers have reported that accuracy is good as 98% or 99% for handwritten digit recognition. For MNIST digit recognition tests the researchers developed an ensemble model that combined multiple CNN models and achieved an accuracy of 99.73%, which was later improved to 99.77% with the 35-net ensemble method [13]. Thus, this work makes significant contributions in two key areas. First, it conducts a detailed evaluation of various parameters such as number of layers, striate size, kernel size, padding and expansion to improve handwritten digit recognition performance in CNN architecture. Second, it achieves better recognition performance by optimizing the learning parameters on the MNIST dataset [14]. This work uses the MNIST database because of the comprehensive output of results with different classifiers. Also, the standard database is popular and often serves as a benchmark for comparative studies in various tests connected to handwritten digit recognition in different regional and international languages [15].

### *1.1. Motivation*

The motivation to pursue this research lies in its ability to significantly improve the complementarity of advanced machine learning techniques for handwriting analysis and personality classification. By improving pre-processing methods, feature extraction and classification algorithms, it aims to achieve greater accuracy and flexibility in identifying different handwriting styles. This will reduce current limitations reporting and misclassification problems. The desire is to expand its practical applications in fields such as education, security, and personalized assessments, and to develop a model that performs robustly on diverse datasets and real-world scenarios. Also, the collaborative effort aims to expand the relationship between handwriting and personality characters, contributing to more effective and insightful analyses. The major contributions of the research is:

- To reduce the noise and improve the contrast of the input signature images using an improved Gaussian filter (IGF) in the preprocessing step.
- To extract textual features using hybrid PCA-NGIST method with enhanced feature representation
- Optimal features are selected by the Adaptive Horse Herd Optimization (AHHO) algorithm
- This selected features are classified using the IGAN\_AHb classifier, improving the accuracy and efficiency of the classification process.

The paper is ordered as follows: section 2 presents the related work and problem statement, section 3 provides the proposed methodology, section 4 shows the results and discussions, section 5 final parts summarized the conclusion and future scope, section 6 represent reference.

## 2. Related Works

Deore et al. [16] presented a Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset. Devanagari created a publicly available dataset of handwritten scripts. It contains 5800 isolated images of 58 separate letter classes: 12 vowels, 36 consonants and 10 numbers. In addition, appreciation of these characters was implemented using two advanced adaptive gradient methods using a two-stage VGG16 deep learning model. A two-stage approach of deep learning is developed to improve the overall success used in Devanagari Handwritten Character Recognition System (DHCRS). A test accuracy of 94.84% was achieved by the first model with a training loss of 0.18 on the new dataset. Furthermore, to achieve state-of-the-art performance on a much smaller dataset, the second fine-grained model requires significantly fewer trainable parameters and significantly less training time. However, Training deep convolutional networks requires significant computational power. Sophisticated deep learning models are complex and time-consuming.

An important and controversial problem is the development of a multilingual handwritten system. This problem is solved by CNN-based language-independent model. Fateh et al. [17] used the multilingual handwritten numeral recognition using a robust deep network joint with transfer learning. The model is designed to handle multi-script image recognition, language recognition and digit recognition. Transfer learning is used to improve the image quality and consequently the recognition performance. The performance of both the language and digit recognition procedures was verified through extensive tests. However, the existing model focuses on number recognition and has not been extended to the recognition of multilingual handwritten characters. In addition, it is challenging to increase the setting for optimal hyperparameters and number of layers without increasing the computational complexity.

He et al. [18] utilized a Deep adaptive learning for writer identification based on single handwritten word images. The advantages of features learned from an adaptive neural network are initially applied to secondary tasks such as explicit content recognition and transferred to the primary task of author recognition in a single procedure. A new adaptive convolutional layer is proposed to take advantage of these learned depth features. End-to-end training of a multi-task neural network, including one or more adaptive convolutional layers, is conducted to exploit robust common features for teacher identification. Three subtasks are assessed that relate to three expressive properties of handwritten word pictures: lexical content, word length, and letter attributes. However, more complex neural network architectures should be explored to more improve system performance.

Berriche et al. [19] used in Hybrid Arabic handwritten character segmentation using CNN and graph theory algorithm. This method produces candidate segmentation points and uses a graph theory-based technique to segment words into sub-words, achieving a correct sub-word segmentation rate of 96%. Additionally, a segmentation hypothesis graph (SHG) was created to make candidate characters for each subword. Both a manually designed convolutional neural network and a transfer learning-based CNN using a pre-trained Alexnet model are used. However, the method requires heuristic optimization for effective computation of segmentation hypothesis graph (SHG). In addition, generalization with different recognition approaches to different datasets presents significant challenges.

Khorsheed et al. [20] implemented the Handwritten Digit Classification Using Deep Learning Convolutional Neural Network. The recent summary of machine learning, particularly Convolutional Neural Networks (CNNs), has significantly improved accuracy in recognizing different handwriting patterns. Different filter sizes are used in this study to develop a deep CNN model with the aim of improving the handwritten digit recognition rate. The multi-layer deep architecture of the proposed model includes a fully connected layer (dense layer) for classification

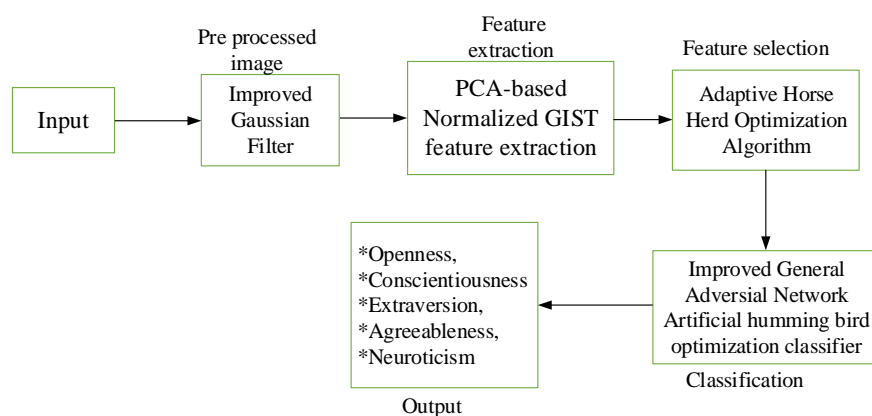
and a transformation and activation layer for feature extraction. However, the proposed method is characterized by its potential complexity and high computational cost, requiring a large amount of resources and extensive training time. In addition, increased complexity model makes tuning and optimization more demanding, leading to unreliable performance on different datasets.

### 2.1. Problem Statements

Important challenges are aligning feature extraction methods with differences in multilingual handwriting styles and accurately translating personality behaviors into actionable insights in different cultural contexts. Effective reducing of these problems necessary to improve the credibility and compatibility of the proposed method in real-world situations. Despite advances in handwriting testing and personality classification, along with these techniques for multilingual applications (especially English and Hindi handwriting) faces difficulties. Extracting different handwriting styles, improving classification accuracy with complex deep learning frameworks such as IGAN\_AHb and handling computational demands are critical.

## 3. Proposed Method

Handwriting analysis is a systematic study of preserved graphic structures. Which are generated in the human brain and produced on paper in cursive or printed style. The style in which a text is written reflects an array of meta-information. Personality is a combination of an individual's behavior, emotion, and motivation and thought-pattern characteristics. It has an impact on one's life choices, well-being, health, and numerous other preferences. The pre-processing of input handwriting image includes noise reduction and contrast enhancement using the IGF. After the pre-processing, the textual features are extracted using PCA-based normalized GIST feature extraction method (hybrid PCA-NGIST) approach. Furthermore, the feature selection is performed using an AHHO. The features are delivers to the deep learning method based on Improved General Adversarial network Artificial humming bird optimization (IGAN\_AHb) classifier used to classify the selected features. The OCEAN model, which stands for Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism, is the methodology used to categories the Big Five personality traits and is the development for personality trait images. Here, the personality classification is analyzed for both English and Hindi language handwriting. Figure 1 represents the block diagram of basic proposed model.



**Figure 1.** Block diagram of basic proposed model.

### 3.1. Pre-Processing

The pre-processing images with an enhanced Gaussian filter helps improve the performance of noise reduction, edge protection, image enhancement, feature extraction and image analysis algorithms.

The Gaussian filter is commonly used to reduce noise in image processing and computer vision. It works by smoothing a noisy image, efficiently generating pixel values in a Gaussian distribution.

This filter is direct and local, meaning it smooths the image without focusing on edges and details. Its efficiency lies in handling noise following a standard distribution by applying the normalization of each pixel to its neighbors. This method helps reduce noise while maintaining reliable overall image quality. Edge blurring some filters, like as Gaussian blurring, make edges less defined and harder to distinguish. So, in order to enhance the filter, this research uses improved Gaussian filter for more appropriate results.

### 3.1.1. Improved Gaussian Filter

This paper uses an anisotropic directional Gaussian filter to improve edge structure and reduce noise. The kernel of this filter is provided,

$$DG(a,b;\theta, p, \mu_a, \mu_b) = z - \left( \frac{a_\theta^2}{2\mu_a^2} + \frac{b_\theta^2}{2\mu_b^2} \right) \quad (1)$$

Decompose to shorten the activation process  $DG$  Enable as two filters. From  $\mu_b^2$  much smaller than  $\mu_a^2$ , decompose the filter into a compact isotropic filter.

$$IG(b; \mu_b) = z - \frac{a^2 + b^2}{2\mu_b^2} \quad (2)$$

$$AG(a_\theta; \theta, \mu_\theta) = z - \frac{a_\theta^2}{2\mu_\theta^2} \quad (3)$$

It is defined by the formula  $\mu_\theta^2 = \mu_a^2 - \mu_b^2 \approx \mu_a^2$  and  $a_\theta = a \cos \theta + y \sin \theta$ . Isotropic 2-D Gaussian scale filter  $\mu_b$ , the resulting image corresponds to a 1-D Gaussian  $\theta$  direction. By varying  $\theta$  From pixel to pixel, an improved Gaussian filter is generated, yielding an anisotropically smoothed image. After pre-processing, the features are extracted using PCA-NGIST method.

### 3.2. Feature Extraction Using PCA-NGIST Method

Feature extraction is a method of converting raw data into a set of significant features that can be used for further analysis such as classification or clustering. These features reduce the dimensionality of data while making it easier to capture, process and analyse essential information. PCA-NGIST combines Gabor filters for spatial feature extraction and PCA is used for dimensionality reduction, improving image representation by focusing on key spatial patterns by reducing noise and refining computational efficiency. The PCA-NGIST method uses Principal Component Analysis (PCA) and Normalized GIST (NGIST) interpretation to extract and discard significant features from images. NGIST improves on traditional GIST by normalizing features to smooth out light and shadow variations, providing a concise representation of image directions and measurements without segmentation. PCA secure hard feature extraction and dimensionality reduction increases classification accuracy by creating a more adaptive feature set while preventing overfitting.

Let  $f(a,b)$  be the 2D Gabor filter of the image in  $e$  dimensions and  $z$  orientations, is calculated as:

$$f(a,b) = \left( \frac{1}{2\pi\mu_a\mu_b} \right) \exp \left[ -\frac{1}{2} \left( \frac{a^2}{\mu_a^2} + \frac{b^2}{\mu_b^2} \right) + 2\pi n \gamma a \right] \quad (4)$$

Here,  $\gamma$  denotes the radial frequency of the Gabor filter,  $n$  is the composite number  $\sqrt{-1}$ ,  $\mu_a$  and  $\mu_b$  are the Gabor filter Absolute and non-orthogonal bases.

The Fourier transformation of the 2D Gabor filter function  $F(i, j)$  and  $f(a, b)$  can be expressed as:

$$F(i, j) = \exp \left\{ -\frac{1}{2} \left[ \frac{(i - \gamma)^2}{\mu_i^2} + \frac{j^2}{\mu_j^2} \right] \right\} \quad (5)$$

Here  $\mu_i$  and  $\mu_j$  can be calculated as:

$$\mu_i = \frac{1}{2} \pi \mu_a \text{ and } \mu_j = \frac{1}{2} \pi \mu_b \quad (6)$$

Consider  $f(a, b)$  as the source function of the Gabor wavelet the transformation is obtained by setting the dictionary of the Gabor filter the orientation  $\theta$  and scaling factor  $\delta$  parameters are as follows:

$$f_{ez}(a, b) = \delta^{-e} f(a', b'), \quad (7)$$

where  $\delta > 1, e$  and  $z$  are integers Magnitude number and orientation number,  $\theta = z\pi/Z$  and  $a', b'$  Calculated:

$$a' = \delta^{-e} (a \cos \theta + b \sin \theta) \quad (8)$$

$$b' = \delta^{-e} (-a \sin \theta + b \cos \theta) \quad (9)$$

Let  $R$  be the number of orientations and calculate the value of  $\theta$  as the  $\theta = z\pi/R$ . The quantity  $\delta$  Equations (6), (7), and (8) are designed to maximize the energy freedom.

$$D_m = \frac{D_m}{\sqrt{\sum_{n=1}^{512} |D_m|^2}} \quad (10)$$

Let  $H$  be an  $Z \times 512$  array of NGIST feature vectors  $D_m$ . A PCA unsupervised learning algorithm is used to compute the selected eigenvectors  $SJ_{512 \times C}$  in a redundancy-minimized manner on these feature vectors. These eigenvectors  $H_{Z \times 512}$  are then used to introduce a new small feature matrix  $B_{Z \times C}$  through the following equation:

$$B_{Z \times C} = H_{Z \times 512} \cdot SJ_{512 \times C} \quad (11)$$

The steps of the PCA algorithm for computed  $SJ_{512 \times C}$  are given: Let  $Q$  be the number of hand writing/results in the training dataset  $H$  of  $Z$  NGIST vectors.  $\{D_1, D_2, \dots, D_Z, D_m \subset \text{RealNumber}\}$ ; each exercise belongs to the class A vector  $n \subset \{1, 2, \dots, R\}$ . The covariance sequence is defined as:

$$O = \frac{1}{Z-1} \sum_{m=1}^Z (D_m - \bar{D}) \cdot (D_m - \bar{D})^H \quad (12)$$

The vector  $\bar{D}$  denoted the mean of training data vectors and is assumed as follows:

$$\bar{D} = \frac{1}{Z} \sum_{u=1}^Z D_m \quad (13)$$

Choose from  $K$  eigenvectors  $SJ_{512 \times c}$ , the matrix of original eigenvectors  $SJ_{512 \times 512}$  is related. The upper  $C$  eigenvalues result from decomposition Covariance matrix  $O$ , which is denoted as:

$$SJ_{512 \times c} = SJ(m, n), \quad (14)$$

Where  $m = 1, \dots, 512$  and  $n = 1, \dots, C$ .

The PCA-NGIST method for feature extraction reduces the amount of data and removes similar features. Furthermore, PCA mainly deals with linear relationships and may miss more complex connections between features. After feature extraction, the feature selection is done by using AHHO algorithm.

### 3.3. Features Selection Using AHHO

The Adaptive herd Horse Optimization algorithm (AHHO) helps to remove most of the redundant information by selecting the most important features. AHHO can detect and exploit non-linear relationships, improving the overall quality of features. It finds the best features for specific tasks, makes the model perform better and avoids overfitting. AHHO is inspired by the herding behavior of horses. Social behavior in horses is classified into six groups based on age: sociability among (R). As with most metaheuristic algorithms, this procedure improves the accuracy and efficiency of the subsequent classification by the IGAN\_AHB classifier equation by (15)

$$\text{Classification Accuracy} = \frac{Nccl}{Tns} \quad (15)$$

$Nccl$  denotes the number of correct classifications and  $Tns$  denotes the total number of samples in the dataset. The dimension reduction rate is determined using the equation:

$$\text{Dimension reduction Ratio} = \frac{NoSF}{Tnf} \quad (16)$$

Where  $NoSF$  number of specific features and  $Tnf$  indicates the total number of features in the dataset.

AHHO begins by setting and modifying control parameters such as the maximum number of repetitions and population size. At (S), hierarchy (H), sociability (S) grazing (G), defense mechanism (T), imitation (I) and both iteration, the horses move according to equation 17.

$$C_o^{e, Age} = \vec{W}_o^{e, Age} + C_o^{(e-1), Age}, \quad (17)$$

A new level  $o$  on the  $o^{th}$  horse  $e$  is denoted as the  $e^{th}$  iteration is signified as  $C_o^{e, Age}$ . The old status of  $o^{th}$ , there is another horse  $C_o^{(e-1), Age}$ . Where, velocity vector  $o$  referred to as the  $o^{th}$  horse  $\vec{W}_o^{e, Age}$ .

The Herd Optimization Algorithm (HOA) operates in continuous space, while the feature selection problem operates in discrete space. To use HOA to select a feature, a continuous interval must be converted into a discrete space. A U-shaped transfer function is used in this transformation proposed method. A U-shaped transfer function is assumed  $d$  (the slope) and  $a$  (the width). The function is calculated as follows:  $d$  (decline) and  $a$  (width). The function is calculated as follows:

$$U(C_{o.z}^e) = \gamma |C_{o.z}^e|^\beta \quad (18)$$

$$C_{o.z}^e = \begin{cases} 1 & \text{rand} < C(C_{o.z}^e) \\ 0 & \text{rand} \geq C(C_{o.z}^e) \end{cases} \quad (19)$$

The probability value of the U-shape is connected by the U-shape transfer function  $U(C_{o.z}^e)$ . The values of the elements are changed to 1 or 0 using equation (19). The movement of horses in AHHO in the adaptive-representative search space is calculated using equation (20).

$$c_i^{k+1} = c_i^k + \theta * (C_{Hors_{q_{est}}}^k - C_i^k) \quad i = 1, 2, \dots, N \quad (20)$$

Positions  $i$ th Horse again  $k$  and  $k + 1$  denoted as  $c_i^k$  and  $c_i^{k+1}$ , individually. The position of the  $C_{Hors_{q_{est}}}^k$  is indicated in the search space  $c$ , A random number  $r$  is equally distributed between 0 and 1,  $N$  signifies the number of horses. AHHO simulation initiates by generating a randomly generated population of horses. Every horse is in position  $d$  Denoted as  $C_p = (C_{p1}, C_{p2}, C_{p3}, \dots, C_{pn})$  where  $n$  represents the horse in position  $p = 1, 2, \dots, N$ ,  $z = 1, 2, \dots, P, U(0,1)$  generate a function on a continuously distributed random value between 0 and 1.  $C_z^{\min} C_z^{\max}$  defining the search space boundaries of the boundaries in  $z$  position.

$$C_{p.z} = C_z^{\min} + (C_z^{\max} - C_z^{\min}) * U(0,1) \quad (21)$$

Improves the overall quality of AHHO features. However, it does not classify or perform calculations based on these features alone.

### 3.4. Improved Generative Adversarial Networks Using Classification

Improved General Adversarial Network with Artificial Hummingbird Optimization (IGAN\_AHb) classifier improves the accuracy and display of predictions by optimizing the best quality features set for better classification results. Generative Adversarial Networks (GANs) use a probability theory method to generate data. The basic architecture consists of two feed-forward neural networks: a generative model  $A$  and a discriminative model  $B$ . In this system, the generator  $A$  takes random inputs  $n$  and  $C(n)$  transforms them into real data  $C_{Data}(k)$  usage information  $A(n)(X)$ . A discriminator  $B(x)$  checks these data points and decides whether they are true or false  $A$ . The generator  $A$  trains the discriminator  $B$  to prove that the samples it generates come from the true data distribution. Therefore,  $A$  and  $B$  are trained by countermeasures similar to a two-player min-max game, as shown in Equation (22).

$$\min_A \max_B f(B, A) = E_{k \approx C_{Data}(k)} [\log B(k)] + E_{k \approx C(n)} [\log(1 - B(A(n)))] \quad (22)$$

Improved variation of GANs is proposed to solve the problem of sequence of data generated by random and unconstrained mode or basic GANs. Supplementary or conditional information  $t$  is used to stabilize the generating process by being fed  $t$  as input to both  $A$  and  $B$ . Previous entry  $n$  Attached with additional information  $t$  and fed into the generator  $A$ , in entry  $k$  is the discriminant  $B$  and  $t$ . This results in an improved version of the opposite intention.

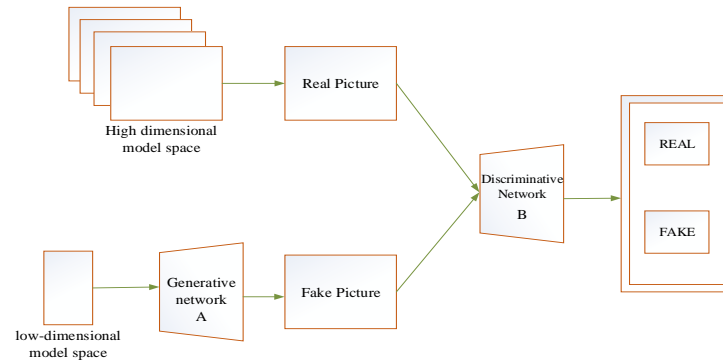
$$\min_A \max_B f(B, A) = E_{k \approx C_{Data}(k)} [\log B(k/t)] + E_{k \approx C(n)} [\log(1 - B(A(n/t)))] \quad (23)$$

Improved GANs face many challenges such as optimal solution problems and mode collapse problems, where the generator produces limited output types. The training may be unbalanced, leading to oscillations and failure to effectively balance the generator and discriminator. In addition, GANs are too complex for hyperparameter selections, which makes the tuning process complicated and time-consuming. Training GANs is labor intensive, demanding significant computational power and time.

In order to fine tune the proposed classifier, AHHO is used in this research. The determination of loss function is given as

$$LF = \frac{1}{x} \sum_{a=1}^A (a_x - a'_x) \quad (24)$$

Where  $a_x$  signify the actual value,  $a'_x$  denote the predicted value, and  $X$  denotes the total number in the training set. To reduce the loss, the proposed study assumed a metaheuristic based AHHO algorithm. Figure 2 represents the GAN architecture.



**Figure 2.** Generative Adversarial Networks Architecture.

### 3.4.1. The Artificial Hummingbird Optimizer

Artificial hummingbird optimization (AHO) enables fast and efficient convergence of GANs. This minimizes mode collapse and ensures training with consistently fine-tuned parameters. AHO simplifies hyperparameter tuning and reduces computational requirements, improving overall performance. The AHO algorithm optimizes the parameters using the following fitness function.

$$Fitness = Maximum[accuracy] \quad (25)$$

The AHO, recently introduced, draws inspiration from the specialized flight and foraging behaviors of hummingbirds in nature. Just as hummingbirds select flowers based on factors such as nectar quality, content of individual flowers, nectar filling rate, and time of last visit, this biologically inspired optimization approach involves early stage selection of food sources. In the AHA optimizer, each solution vector is characterized by a food source, and the fitness value is solved by the honey replenishment rate associated with each source. Higher fitness values were associated with higher honey replenishment rates. Each "hummingbird" in the optimization is directed to a specific food source, successfully placing the hummingbird and the food source in the same location. The algorithm then remembers the positions of the food sources and their fluid replenishment rates. Hummingbirds in a population share information about their last visit to each food source and their path. A visit list table records the frequency of visits to each food source. Food sources with high visit frequencies are selected in future optimization steps.

$$q_a = Fu + rand \times (C_z - Fu), a = 1, 2, \dots, n \quad (26)$$

Where, denotes  $q_a$  th level of the food source,  $Fu$  and  $C_z$  represent the minimum level, the upper hard search-space constraints, and the  $rand$  represents a random number between  $[0, 1]$ . The arrival schedule of food sources is initialized as follows:

$$DB_{a,b} = \begin{cases} 0, & a \neq b \\ null, & a = b \end{cases} \quad i = 1, 2, \dots, n \text{ and } b = 1, 2, \dots, n \quad (27)$$

AHO tends to have a faster convergence rate compared to traditional optimization algorithms because of its dynamic and intelligent search patterns.

The AHO model which used to fine-tune the classifier improves the classification accuracy and assists to achieve better results.

## 4. Result and Discussion

The suggested model is implemented in this research using the python tool. The research is conducted on a computer with an x64-based processor, 8 GB of installed RAM, a 64-bit operating system, and an Intel(R) Core(TM) i5-3470 CPU running at 3.20 GHz. The proposed model compared with VGG 16, ResNet 50 and Google Net in terms of accuracy, precision, recall, F1 Score.

#### 4.1. Performance Metrics

These criteria help determine the algorithm's overall performance and suitability for specific applications. Assessment can be done using a variety of tools at python platform. Together, these metrics help assess the strengths and weaknesses of the classification model.

**Accuracy:** The percentage of correctly classified, segmented images.

$$\text{Accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{true negative} + \text{false positive} + \text{false negative}} \quad (28)$$

**Recall (R):** Recall is the ratio of properly classified images to the total number of images belonging to x class.

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \quad (29)$$

**Precision (P):** Fraction of images and Total number of correctly classified images categorized:

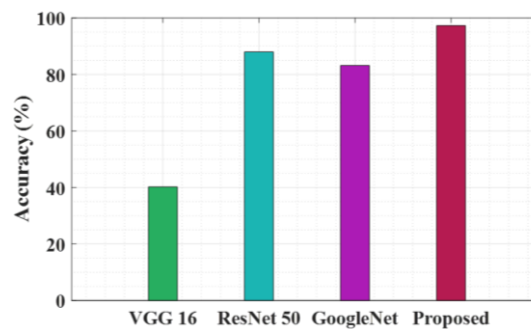
$$\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \quad (30)$$

**F1 Scale:** It is a scale that correlates recall and precision:

$$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (31)$$

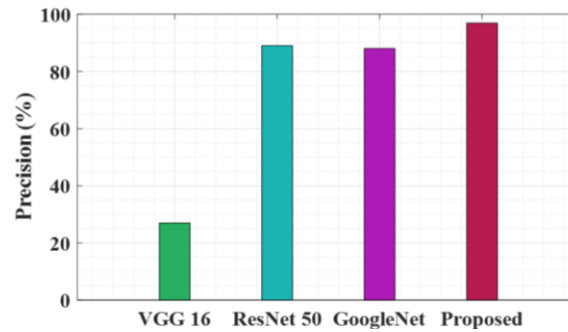
#### 4.2. Performance Evaluation

Figure 3 represents the accuracy of various architecture in four models: VGG 16, ResNet 50, Google Net and the proposed model. VGG 16 indicates very low accuracy, above 20%. ResNet 50 and Google Net perform well with 90% accuracy. The proposed model has a peak accuracy of 97.3, which is close to 100%.



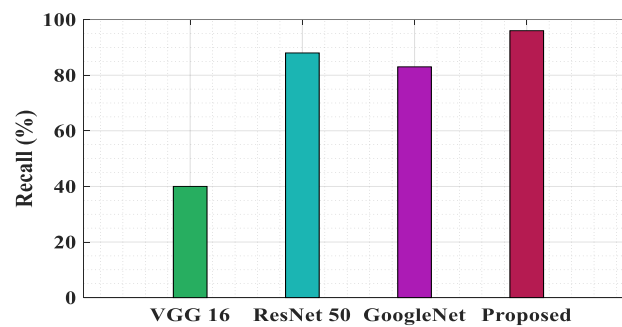
**Figure 3.** Accuracy of various architecture.

Figure 4 compares the precision of various architecture in four models: VGG 16, ResNet 50, GoogleNet and the proposed model. VGG 16 has very low accuracy, above 20%. ResNet 50 and GoogleNet Expression 90% higher accuracy. The proposed model achieves a maximum precision of 96.9, almost reaching 100%.



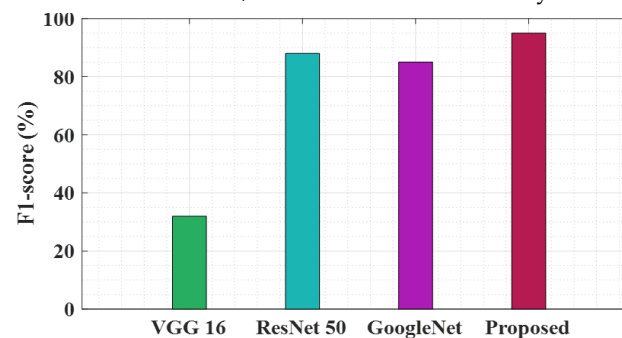
**Figure 4.** Precision of various architecture.

Figure 5 is a graph comparing the recall percentages of four different architectures: VGG 16, ResNet 50, Google Net, and the proposed model. The vertical axis indicates recall ranges from 0 to 100. The VGG 16 bar is very small, indicating a low recall percentage. ResNet 50 and Google Net have comparable heights, showing comparable recall percentages that are higher than VGG 16. The proposed model has the highest bar, indicating the highest recall 96% among the four. This graph highlights the best performance of the proposed model in terms of recall, a key metric for evaluating classification models.



**Figure 5.** Recall of various architecture.

Figure 6 compares the F1 scores of four different network architectures: VGG 16, ResNet 50, GoogleNet, and the proposed model. The vertical axis is the "F1-score range from 0 to 100. The VGG 16 bar is very narrow, indicating a low F1-score. ResNet 50 and Google Net have comparable heights, giving comparable F1-scores that are higher than VGG 16. This graph highlights the best performance of the proposed model in terms of F1-score, a fraction of the accuracy of a test.



**Figure 6.** F1 score of various architecture.

The Figure 7 illustrates a receiver operating characteristic (ROC) curve plot that measures the performance of the classification model. The horizontal axis plots the false positive rate and the vertical axis plots the true positive rate. There are five ROC curves, each with separate area under the curve (AUC) values for a different class (from 0 to 4): Class 0 (0.9732), Class 1 (0.9587), Class 2 (0.9187), Class 3 (0.8987), and Class 4 (0.9012). The dashed diagonal line represents the line of no discrimination

where the true positive rate equals the false positive rate. This diagram helps to combine the classifier's ability to distinguish between true positives and false positives through multiple classes.

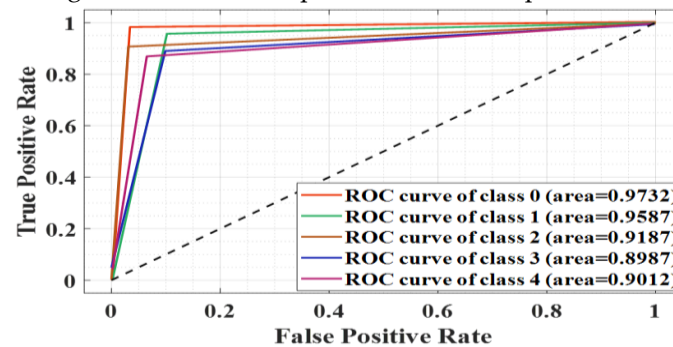


Figure 7. Receiver operating characteristic curve.

The Figure 8 shows graphs comparing the performance of the proposed model over 100 training and validation epochs. The left graph shows the training and validation accuracy, with the new model performing better and achieving higher accuracy faster. The right graph shows the training and validation loss, the new model reduces the error quickly and maintains a low loss. In general, the new model is more accurate and has fewer errors than the existing model.

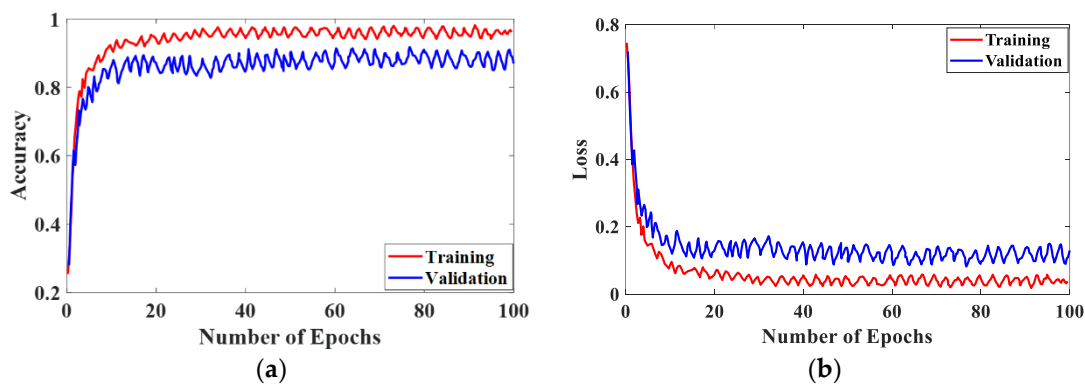


Figure 8. proposed model for training and validation.

## 5. Conclusion

In this research, a new deep learning algorithm that is designed for handwritten personality prediction is presented. Initially, the input data is gathered from the provided dataset and then pre-processing is performed using an improved Gaussian filter to increase image quality by decreasing noise and boosting contrast. Following pre-processing, PCA-based normalized GIST is used to extract features and standardize them in order to enhance image representation. An adaptive horse herd optimization technique is used to identify the most significant features from extracted data. GANs may converge quickly and effectively thanks to IGAN\_AHb, which is based on specific features. In the result section, the model achieved a peak precision of 97.3%, a maximum precision of 96.9% and a recall rate of 96%, the highest among the four models tested. These results suggest that the proposed model is more accurate and has less errors compared to existing models, making it more useful for classification tasks. In future studies, this research intent to use more advanced model for classification and feature extraction. Furthermore, this research likes to include model advanced dataset contain more features to improve the classification performance.

## References

1. Avis, Giulia Rosemary, Fabio D'Adda, David Chierigato, Elia Guarnieri, Maria Meliante, Andrea Primo Pierotti, and Marco Cremaschi. "A Neural Network for Automatic Handwriting Extraction and Recognition in Psychodiagnostic Questionnaires." In *ICT4AWE*, pp. 140-151. 2024.

2. Siranjeevi, P., R. Ruthran, R. Ramchandru, and R. Kaladevi. "Enhancing Hand Script Digitization using Deep Learning." In *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS)*, pp. 1-6. IEEE, 2024.
3. Jain, Jaishree, Garima Saroj, Ayushi Gautam, Bhavya Agrawal, Sarthak Gupta, and Yogendra Narayan Prajapati. "Modernising Medical Records: Region-based Convolutional Recurrent Neural Network and Connectionist temporal classification-Based Doctor's Handwriting Recognition." (2024).
4. Preethi, P., R. Asokan, N. Thillaiarasu, and Thangavel Saravanan. "An effective digit recognition model using enhanced convolutional neural network based chaotic grey wolf optimization." *Journal of Intelligent & Fuzzy Systems* 41, no. 2 (2021): 3727-3737.
5. Barati, Ramin. "Incorporating locally linear embedding and multi-layer perceptron in handwritten digit recognition." *e-Prime-Advances in Electrical Engineering, Electronics and Energy* 2 (2022): 100081.
6. Ahlawat, Savita, Amit Choudhary, Anand Nayyar, Saurabh Singh, and Byungun Yoon. "Improved handwritten digit recognition using convolutional neural networks (CNN)." *Sensors* 20, no. 12 (2020): 3344.
7. Alom, Md Zahangir, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen, Abdul AS Awwal, and Vijayan K. Asari. "A state-of-the-art survey on deep learning theory and architectures." *electronics* 8, no. 3 (2019): 292.
8. Albahli, Saleh, Marriam Nawaz, Ali Javed, and Aun Irtaza. "An improved faster-RCNN model for handwritten character recognition." *Arabian Journal for Science and Engineering* 46, no. 9 (2021): 8509-8523.
9. Drott, Beatrice, and Thomas Hassan-Reza. "On-line handwritten signature verification using machine learning techniques with a deep learning approach." *Master's Theses in Mathematical Sciences* (2015).
10. El Khayati, Mohsine, Ismail Kich, and Youssef Taouil. "CNN-based Methods for Offline Arabic Handwriting Recognition: A Review." *Neural Processing Letters* 56, no. 2 (2024): 115.
11. Samsuryadi, Rudi Kurniawan, and Fatma Susilawati Mohamad. "Automated handwriting analysis based on pattern recognition: A survey." *Indonesian Journal of Electrical Engineering and Computer Science* 22, no. 1 (2021): 196-206.
12. Rabby, AKM Shahariar Azad, Md Majedul Islam, Nazmul Hasan, Jebun Nahar, and Fuad Rahman. "Borno: Bangla handwritten character recognition using a multiclass convolutional neural network." In *Proceedings of the Future Technologies Conference*, pp. 457-472. Cham: Springer International Publishing, 2020.
13. Saqib, Nazmus, Khandaker Foysal Haque, Venkata Prasanth Yanambaka, and Ahmed Abdelgawad. "Convolutional-neural-network-based handwritten character recognition: an approach with massive multisource data." *Algorithms* 15, no. 4 (2022): 129.
14. MUNSARIF, MUHAMMAD, Edi Noersasongko, Pulung Nortantiyo Andono, A. Soeleman, and Muhammad Sam'an. "An improved convolutional neural networks based on variation types of optimizers for handwritten digit recognition." Available at SSRN 4055758 (2022).
15. Rahman, Md Moklesur, Md Shafiqul Islam, Roberto Sassi, and Md Aktaruzzaman. "Convolutional neural networks performance comparison for handwritten Bengali numerals recognition." *SN Applied Sciences* 1, no. 12 (2019): 1660.
16. Deore, Shalaka Prasad, and Albert Pravin. "Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset." *Sādhanā* 45, no. 1 (2020): 243.
17. Fateh, Amirreza, Mansoor Fateh, and Vahid Abolghasemi. "Multilingual handwritten numeral recognition using a robust deep network joint with transfer learning." *Information Sciences* 581 (2021): 479-494.
18. He, Sheng, and Lambert Schomaker. "Deep adaptive learning for writer identification based on single handwritten word images." *Pattern Recognition* 88 (2019): 64-74.
19. Berriche, Lamia, Ashjan Alqahtani, and Siwar RekikR. "Hybrid Arabic handwritten character segmentation using CNN and graph theory algorithm." *Journal of King Saud University-Computer and Information Sciences* 36, no. 1 (2024): 101872.
20. Khorsheed, Eman Ahmed, and Ahmed Khorsheed Al-Sulaifanie. "Handwritten Digit Classification Using Deep Learning Convolutional Neural Network." *Journal of Soft Computing and Data Mining* 5, no. 1 (2024): 79-90.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.