Chapter One

Introduction

1.1 Background to the Study

In the science of image processing, determining blur type is essential for blind picture restoration. Picture blur type categorization is crucial for blur image enhancement, although it is a stimulating subject due to the diverse causes of image blurring. As the interaction of nature fog (haze blurring), optical lens deformation (unsharp mask blur), air disturbance (Probability distribution blur), and webcam relative movement during exposition (photo roughly equivalent movements) (motion blur). These grayscale images are commonplace in regular activities but incredibly challenging to detect and recognize[1, 2].

There are two approaches to recognizing image blur: handcraft feature-based approaches (dimensionality reduction-based) and learning functionality techniques. Dimensionality Reduction (DR), which can be categorized into two significant aspects, Feature Selection, and Feature Extraction, has enabled the separation of blurry elements, making the image in the dataset deblurred[3]. Algorithms that rely on feature extraction necessitate a prior understanding to recover blurring attributes that can recognize various categories of blurring photos, with the selected features of sample imageries then being used to train the assigned classifiers. On the other hand, the systems based on learned characteristics utilize only the innovative blur photographs to automatically study the distinctions between the various fuzziness kinds[4].

Character recognition from scene text images is challenging in computer imaging and rendering. When adaptive edge devices are incorporated into the procedure, the difficulty of

Do Not Copy, Lead City University, Nigeria

the task increases significantly. Text classification and detection are hampered by poor image resolution, which includes blurring, surface texture, and poor resolution[5].

Smart cities have exploded in popularity in recent years, particularly in the fields of health care, finance, education, video surveillance, and IoT-based autonomous cars. Edge and app-intelligence methods are required in such systems. Written data (Textual data) in images provide crucial information for content-based image repositories and a variety of other computer vision applications[6]. The textual information changes when there are differences in Arial size, style, configuration, and unpremeditated alignment, and its recognition (location and proof of identity) and classification (authentication) are made more difficult by the short disparity, little firmness, blur, and complicated background.

Text recognition and classification are frequently hampered by scenic qualities such as focus and related (stylistic and non-textual data). Variations in size, color, font, and orientation in the fore constitute a challenge in robustly detecting stylistic data from scene text photographs. On the other hand, images with complex backgrounds comprehend a variation of items with diverse colors, as well as atmosphere, grassland, blocks, and hedges, which reduces the strength and makes textual feature extraction difficult[7].

Several models, for example, deep and machine learning algorithms (DL and ML), are utilized in modern procedures for semantic visual tasks, including picture classification and semantic segmentation, and large annotated datasets of high-quality, artifact-free images are routinely used alongside these DL and ML models to train and assess these networks[8]. By demonstrating that ordinary pre-trained network models suffer a significant performance decrease when applied to blurred photos, the influence of one such artifact that is highly ubiquitous in natural capture conditions, blur, is of the essence[9,10].

Do Not Copy, Lead City University, Nigeria

Textual recognition has recently seen a surge in demand in various industries, including mailing sorting automation, authorization plate identification, and automated memo pads. In addition, in the aspect of picture recognition, approaches grounded on neural networks have been extended to textual recognition with excellent results. However, the number of learning and cognitive development aspects is growing due to the assortment of recognition application domains[11]. Because of the vast amount of data produced in the modern era, it is becoming extremely relevant to the insightful group, categorize, or classify data by concept for quick retrieval and lookup. However, the high dimensionality and inaccuracy of data, or even more broadly language, make unsupervised learning difficult[12]. Hence, there is a need to explore dimensionality reduction to read more text in the wild clearly.

To address the issues mentioned earlier, a hybrid dimensionality reduction technique is proposed in this study to retrieve essential information from a given dataset while minimizing the accuracy loss caused by data compression. As a result, this research offers an Independent Component Analysis (ICA) with an improved Genetic Algorithm (BA-GA) for blurred text recognition by dimensionality reduction. This technique proposes extracting the line segment data that makes up the image of input information and giving each segment an exclusive value. The results of the selected features are suggested to be classified using the Support vector machine (SVM), K-Nearest Neighbor (K-NN), and Ensemble Methods.

Figure 1.1 in Appendix B shows the simplified workflow of the proposed work.

Do Not Copy, Lead City University, Nigeria

1.2 Statement of the Problem

The recognition of word-based information from scene transcript images is a challenging problem in computer graphics and visualization. The task becomes considerably more problematic as it is already established with edge smart devices that are elaborate in the procedure. Text detection and classification are made more challenging by the low-quality image used in these edge innovative strategies, which has issues including blur, low resolution, and low contrast[13]. These challenges with blurred text have now birthed a series of studies that enhanced deblurring[14, 1]. This study is one such, and the blurred text will be addressed in this study.

On the other hand, natural visual feature classifications are a complex problem due to the many kinds of image features, interference, reduced juxtaposition, arbitrary placement of the opening scene (font, style, dimensions, and perspective), and context attributes. Most of all, the excellent quality materials of the information picture's higher dimensional space are a big problem in these cases, but deep learning (DL) makes it a walk-through[15,16]. Most scholars have recommended the use of dimensionality reduction and Machine Learning procedures that can be employed to extract features for pattern recognition. This recommendation makes it essential to reduce dimensions using ML and DL algorithms. Hence, there is a need to use dimensionality reduction (DR) for feature selection and extraction in this study.

Also, numerous ways exist to discover the finest combination of features, choose a set of valuable and different features, and reduce the number of dimensions, all of which help pattern classification work. The selection of features in learning algorithms is a way to choose a subcategory of the great attributes that are more critical to the implementation. GA is effective, inspired by a biological probability optimization algorithm that can be used for an inclusive variability of image computational requirements, such as image augmentation,

Do Not Copy, Lead City University, Nigeria

segmentation techniques, classification techniques, and (of course) variable selection. On the contrary, collaborating to retrieve its iterative process will enhance its efficiency much more[17]. Scholars have suggested that researchers can improve this study by improving GA[18]. As a result, this study proposes an enhanced GA for the blurred text categorization model.

1.3 Aim and Objectives of the Study

This study aims to advance the machine-learning dimensionality reduction model for blurred text detection in natural scene images. The specific objectives are to:

1. design dimensionality reduction (DR) and hybrid dimensionality reduction (DR) models.
2. classify the designed models in 1.
3. assess the performance of the models in terms of accuracy, precision, and f1 score.
4. compare the obtained results with the state of the art with respect to its accuracies.

1.4 Research Questions

1. How can the very high-dimensional sparse vector be fetched from the dataset, and how can the model used to fetch out very high-dimensional sparse vector be improved to enhance the text deblurring?
2. How will the model developed be tested?
3. How will the developed models be evaluated for superior performance and features?
4. What is the outcome of comparing the result obtained from performance evaluation with the related state-of-the-art?

1.5 Significance of the Study

This study will be immensely beneficial in retrieving precise textual information from the internet, akin to finding a needle in a haystack, especially in a blurry text situation. The

Do Not Copy, Lead City University, Nigeria

haystack is a big data warehouse built up on the web over a long period, and machine learning will aid in finding a single piece of information a user requires. On the other hand, text classification is quickly becoming one of the most popular research issues; hence, a study that enhances this field of study is always welcomed.

1.6 Scope of the Study

This study proposes a machine learning approach, with an open-source dataset to be obtained from repositories. It proposes to evaluate the results obtained in terms of evaluation metrics such as accuracy, f1- score, and precision.

1.7 Limitations of the Study

This study is limited to scenic images and not static images. Once scenic images can be used for this study, static images can also pass for any researcher who desires to work on those. Also, this study works only on a dataset ICDAR 2019 SLVT, though very robust and numerous; any other researcher could decide to work on other datasets.

1.8 Operational Definition of Terms

1. **Algorithm:** An algorithm is a procedure for solving a problem or performing a computation. Algorithms act as an exact list of instructions that conduct specified actions step by step in hardware- or software-based routines.
2. **Scene Text:** text that appears in an image captured by a camera in an outdoor environment. The detection and recognition of scene text from camera-captured images are computer vision tasks that became important after smartphones with good cameras became ubiquitous. The text in scene images varies in shape, font, color, and position. The recognition of scene text is further complicated sometimes by non-uniform illumination and focus.

Do Not Copy, Lead City University, Nigeria

3. **Pattern Recognition:** In computer science, it is the process of recognizing patterns by using a machine learning algorithm. It can be defined as the classification of data based on prior knowledge or statistical information extracted from patterns and/or their representation. One of the essential aspects of pattern recognition is its application potential. Examples include speech recognition, speaker identification, multimedia document recognition (MDR), and automatic medical diagnosis. In a typical pattern recognition application, the raw data is processed and converted into an amenable form for a machine to use. Pattern recognition involves the classification and cluster of patterns.
4. **Dimensionality Reduction:** is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse due to the curse of dimensionality, and analyzing the data is usually computationally intractable (hard to control or deal with). Dimensionality reduction is typical in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics.
5. **Blurred Text:** An unclear text seen in images. The text is embedded in scenic images in this case study.
6. **Text Detection:** Text detection is detecting text in the image, followed by surrounding it with a rectangular bounding box. Text detection can be carried out using image-based techniques or frequency-based techniques.
7. **Feature Selection:** is a process of automatically or manually selecting the subset of the most appropriate and relevant features to be used in model building. Feature

Do Not Copy, Lead City University, Nigeria

selection is performed by either including the essential features or excluding the irrelevant features in the dataset without changing them.

8. **Feature Extraction** is transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

1.9 Organization of the Thesis

This section gives a brief overview of this thesis report. The following section, chapter two, sheds light on the works of literature reviewed during the course of the study for its applications and methodologies and the directly related works. Chapter three delves into the methods proposed for this study starting from the dataset to the methods used to create the models, the algorithms, and the classification; the evaluation parameters to be considered; and the tools to illustrate the results generated from the evaluations. Chapter four is the result and the discussion proper, handling the report section by section. First, the models developed, which deals with the first and second objectives; then the classifications, which is the result of the third objective; next, the evaluations tackling the fourth objective; then the comparison of this study with others based on accuracy dealing with the last objective of this study—also, a tabular summary relating the research question, objectives, methods, and results achieved. Chapter Five gives a summary of the study, recommendations, and conclusion.

Do Not Copy, Lead City University, Nigeria

Endnotes

[1] D. Mu, W Sun, G. Xu, & W. Li. "Random Blur Data Augmentation for Scene Text Recognition." ***IEEE Access*** **9** 2021: 136636–136646.

[2]D. Yang & S. Qin, "*Restoration of Partial Blurred Image Based on Blur Detection and Classification*," **Journal of Electrical and Computer Engineering** 2016: 1–12, http://www.hindawi.com/journals/jece/2016/2374926/.

[3]R. Huang, M. Fan, Y. Zing &Y. Zou, "*Image Blur Classification and Unintentional Blur Removal*," **IEEE Access 7** 2019: 106327–106335.

[4]K. Pogorelov, O. Ostroukhova, M. Jeppsson & H. Espeland "*Deep Learning and Hand-Crafted Feature Based Approaches for Polyp Detection in Medical Videos*," in *Proceedings* - **IEEE Symposium on Computer-Based Medical Systems**, vol. 2018-June IEEE, 2018; 381–386.

[5]A. S. Lundervold & A. Lundervold, "*An Overview of Deep Learning in Medical Imaging Focusing on MRI*," **Zeitschrift Fur Medizinische Physik**, May 2019.

[6]M. Shorfuzzaman, M. S. Hossain, & M. F. Alhamid, "T*owards the Sustainable Development of Smart Cities through Mass Video Surveillance: A Response to the COVID-19 Pandemic*," **Sustainable Cities and Society 64** , January 2021: 102582.

[7]K. Hamad & M. Kaya, "*A Detailed Analysis of Optical Character Recognition Technology*," **International Journal of Applied Mathematics, Electronics and Computers** 4, no. Special Issue-1 December 2016: 244–244.

[8]I. R. I. Haque & J. Neubert, "Deep Learning Approaches to Biomedical Image Segmentation," **Informatics in Medicine Unlocked**, 2020.

[9] R. C. Chen, C. Dewi, SW Huang & R. E Caraka "*Selecting Critical Features for Data Classification Based on Machine Learning Methods,"* **Journal of Big Data 7**, no. 1, December 2020: 52

[10]I. Vasiljevic, A. Chakrabarti& G. Shakhnarovich, "*Examining the Impact of Blur on Recognition by Convolutional Networks*" November 2016, http://arxiv.org/abs/1611.05760.

[11]J. H. Alkhateeb, A. A. Turani& A. A. Alsewari, "*Performance of Machine Learning and Deep Learning on Arabic Handwritten Text Recognition*," in **ETCCE 2020 - International Conference on Emerging Technology in Computing, Comm*unication and Electronics* IEEE,** 2020, 1–7.

[12]I. H. Sarker, "*Machine Learning: Algorithms, Real-World Applications and Research Directions,*" **SN Computer Science 2**, no. 3, May 2021: 160.

[13]U. Yasmeen, J. H Shah, M. A Khan, G. J. Ansari, Su Rehman, M. Sharif, S. Kadry & Y. Nam "*Text Detection and Classification from Low Quality Natural Images,*" **Intelligent Automation and Soft Computing** 26, no. 6 2020: 1251–1266.

Do Not Copy, Lead City University, Nigeria

[14]X. Sun, Q. Wang, X. Zhang, C. Xu & W. Zhang "*Deep Blur Detection Network with Boundary-Aware Multi-Scale Features,*" ***Connection Science*** **34, no. 1**, doi:10.1080/09540091.2021.1933906; June 2022: 766–784.

[15]L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu & M. Pietikanen "*Deep Learning for Generic Object Detection: A Survey*," **International Journal of Computer Vision 128**, no. 2 ,2019: 261–318.

[16]S. Ayesha, M. K. Hanif & R. Talib, "*Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data*," **Information Fusion 59** July 2020: 44–58.

[17]L. J. Marcos-Zambrano, K. Karaduzovic-Hadziabdic, T. L Turukalo & V. Trajkovic "Applications of Machine Learning in Human Microbiome Studies: A Review on Feature Selection, Biomarker Identification, Disease Prediction and Treatment," ***Frontiers in Microbiology*** **12,** February 2021.

[18]G. J. Ansari, J. H Shah, M. C. Q Farias, M. Sharif, N. Qadeer & H. U. Khan, "*An Optimized Feature Selection Technique in Diversified Natural Scene Text for Classification Using Genetic Algorithm,*" **IEEE Access 9**, no. April 2021: 54923–54937.

Do Not Copy, Lead City University, Nigeria

Chapter Two

Literature Review

2.1 Overview of Blurred Text Detection in the Wild Scene

Obscurity in pictures makes it harder to see crucial points like edges, shapes, territories, and items, making image processing in the feature space challenging. In the waveform, actuation low contrast and geometric distortion differ slightly, and these structures can be employed to tell the difference between the two types of haze conveniently. When we move the diffraction pattern into the spatial domain, researchers can deduct from the movement item's resonance frequency that its dominant line segments with approaching beliefs are not in the same direction as the sequence. In a low-resolution haze, some geometric negligible structures can be seen. Even if both distorts are present simultaneously, the consequences of both dissolves can be seen[1].

There are various kinds of blurs, and in most the sampling frequency of any image is excellent for figuring out what kind of blur it is. Motion blur and geometric distortion look different in the waveform, and it is easy to tell the difference between them by looking at all these structures. Here, it is pertinent to note that an unclear object's bandwidth is dominated by foggy lines with principles close to zero perpendicular to the direction of movements. When geometric distortion is fuzzy, a few rotating non-existent characteristics can be seen. Even if both distorts are present simultaneously, the consequences of obscurity can be seen. Blur can affect static image data and video (motion) image data. All of these must undergo image noise classification in order to achieve deblurization.

Do Not Copy, Lead City University, Nigeria

The question everyone would ask at this point becomes, '*How then can one remove these obscurities*?' Primarily by noise separation - the procedures in the image noise classification techniques are.

- image data preparation,
- figuring out the sigmoid available bandwidth,
- extracting features from image noise structures,
- formulating a neural net to ensure validity and
- analyzing the findings.

The very first phase is to process and analyze pictures that need to be clarified. Then, to get the low contrast correlations, using log-linear spectrum analyzer of the unclear and unretouched pictures. After getting the low-contrast shapes, the training and evaluation databases are usually set up by calculating the contourlet power characteristics. Finally, training and testing of this feature database are carried out using a feed-forward backpropagation neural network[2].

Furthermore, in image noise categorization, one needs to do a few things first. Initially, the image data from the video camera is turned into eight-bit image pixels by calculating the average pathways for each appearance. Usually, orientations at the edges of a photo often cause significant wavelengths, resulting in them being turned into lines both vertically and horizontally that can be seen in the frequency distribution. As these boundaries might not be the same as or overlap the streaks precipitated by the jumble, they must all be split up using only a synchronization feature before getting changed[3].

At this point, figuring out what document is in an image using either a slider detector window (SDW) or a connected component analysis (CCA) is critical. These methodologies proceed by attaining the message segmentation method, then using a classification algorithm to affirm the

Do Not Copy, Lead City University, Nigeria

segmentation method and method for controlling content's current location[4]. Sliding detection window (SDW) methods, for example, detect text from the top down. This method uses a sliding window to scan the entire scene image, extract entrant transcript sections, and utilize a pre-trained classifier to determine if the text is confined in the sub-window[5].

SDW and CCA are mainly used in scene text recognition (STR) and detection to discover text in composite section pictures. Text detection in several frameworks, for example, records, ID cards, coupons, intelligent road traffic situations, highway symbols, authorized plate recognition, and so on, are examples of scene text recognition (STRS) and detection[6].

Because of the variety of text forms in natural scene images, text discovery in normal scene imageries is more problematic than text detection in scanned paper imageries. In Sceneric text, it is highly possible to mix multiple languages. Characters can come in various sizes, types, ensigns, illumination, contrast, and so on. Text lines can be parallel, perpendicular, coiled, interchanged, warped, or arranged in any other way. The image's text area may also be partial (perception, affinal alteration), have imperfections, distorting, or other effects[7].

Text can exist on a plane, floor, or layered area. The letter can be close to complicated intervention patterns, or non-text regions can all have styles that look like documents, like aggregates, vegetation, guardrails, concrete floors, and others. Different kinds of writing in images Manuscripts in natural scene images can have distinct font sizes, hues, sizes, and perspectives, even within one image, because plot lines in image features usually contain the same font, shape, color, and configuration[8].

Perspectives in photos and videos of natural scene images can have much going on beneath them. Signposts, barricades, concrete blocks, and vegetation are hard to tell apart from actual writing and can conveniently result in misunderstandings and mistakes[9]. Hence, there is a

Do Not Copy, Lead City University, Nigeria

need to evaluate some methods mostly considered in STR and detection, which are discussed in the following sub-sections.

2.1.1 Variables of Interaction: Contour techniques

Loud sound, defocus, disturbance, image resolution, uneven lighting, and cluttered background are all things that can make it hard to find and read the text in an image. In ST, texts are treated as a unique texture by contour techniques, which use regional amplitudes, filtration feedback, and fractal correlation coefficient to tell the difference between text and other parts of an image. Since all locations and scales must be inspected, these techniques are usually costly to operate on a computer. Furthermore, these approaches primarily transact with parallel transcripts and penetrate to revolution and measure change[10].

2.1.2 Document Detectors

The whole programmed to control will find messages in pictures and create boxes around the words. Personality, message, and particular phrase methodologies are the three main types of old ways of doing things[11]. A movable panel method is used to find textual data. This methodology includes trying to move a faceted sub-window through every different location in a photo and then employing an expert classification algorithm to figure out if the writing is located within the navigation pane. Authors have suggested a full-pipeline system in that they use a sliding window (SW) classification model to find characters on different scales[12].

2.1.3 Technique Based on Connected Components (CC)

These methods that fall into this category find and complement minor parts into one significant component, then use a classification model to search out parts that are not documented or transparent, as the case may be, and finally break the textual aspect from a picture and puts it inside a candidate region. These methods are suitable because they are easy

Do Not Copy, Lead City University, Nigeria

to figure out and work well. Individually, they have some problems, such as needing to be more capable of handling twisting, geometric transformations, occlusions, and other tricky situations. The most common CC-based methodologies are suitably sustainable intense sectors (MSERs) and stroke width transform (SWT), which uses the edge detector from Canny to find edges and figure out the spacing per digital image from the most probable sensor that contains a brain aneurysm. MSERs (maximally secure elliptic regions) technique is used to find text by making a segmentation method and then characterize them using identity and non-character categorization that has been learned. Last, the system makes the line of text. An AdaBoost classification model is then employed to find the message[13].

We also have Connected Component Labelling (CCL) algorithms for remote sensing image classification. This algorithm searches line-by-line, top to bottom, to assign a splotch label to each current pixel connected to a splotch, done by assigning a label to a new object. Most labeling algorithms use a scanning step that examines some of its neighbors. The first strategy deeds the dependencies among the neighbors to reduce the number of neighbors examined. The second strategy uses an array to store the equivalence information among the labels, replacing the pointer-based deep-rooted trees used to store the same equivalence information. It reduces the memory required and produces consecutive final labels. The connected component labeling assigns labels to a pixel such that adjacent pixels of the same features are assigned the same label[14].

Do Not Copy, Lead City University, Nigeria

2.1.4 Texture-Based Technique

This method looks at the text as multiple courses of composition. It utilizes image patches like regional amplitudes, filtration system responses, and softmax multipliers to tell the difference between texts and non-textual portions of the image. Although all positions and proportions need to be processed, these methods are usually cost-prohibitive in computing power. Also,

these methods work best with lateral messages and are adaptable to changes in turning and size. A scholar made a filtration that can also be found in the document in the field of discrete cosine transformation (DCT). Although his automated system is fast, it could be better at finding things. A multistage character recognition analyzer with a signal that utilizes a neural network so that machine translation rules can be understood instantaneously is also developed. This method can be used to find different styles and dimensions of the manuscript in a controller[15].

2.2 Texture Image Challenges

Image data evaluation and categorization face two critical problems with undesirable effects. Transformational style includes a rotational and noisy picture. If the approaches always seem to discriminate against certain typical occurrences and are not persistent, the precision of the outcomes can be drastically lowered; therefore, the approaches used to investigate and describe the imagery shall be as durable and steady as conceivable, eliminating their detrimental impacts. Magnitude, orientation, and brightness may also vary between photos, which is a complex texture categorization challenge. Potential treatments to overcome these difficulties were offered[16,17].

2.3 Application of Texture Analysis

The picture structure tells us about the item's structure, components, backing setting, and more. Information is said to be effectively passed through image texts. Edge detection is used in several domains of preprocessing, including Facial Recognition, Media Object Recognition, Quality And product Diagnostic tools, Diagnostic Machine Vision, Satellite Imagery, and Field Interpretation[18].

Do Not Copy, Lead City University, Nigeria

2.4 Machine Learning (ML)

Machine learning drives Artificial Intelligence (AI). It comprises probability distribution, statistical data, approximating principles, geometric analysis, and algorithms cognitive science. ML studies how technologies replicate or provide human behavioral responses to acquire new information or abilities and restructure relevant knowledge to enhance productivity. Its algorithmic knowledge comes from concepts from previous data using methodologies, generating projections or evaluations on new data collected, and then developing like people. Most methods of machine learning use deep knowledge[19].

ML teaches algorithms to analyze information more effectively, though it is important to note that frequently analyzing data while examining it is challenging, which ML tries to resolve. The more data/samples to analyze, the more these ML algorithms become necessary. ML extracts crucial details in numerous sectors via automation, which learns from data[20]. Several academics and developers utilize different ways to solve this complex challenge. ML makes predictions to tackle data challenges. Intelligence scientists say there must be currently no single technique for fixing problems. The type of issue, several variables, the optimal approach, etc., decide the approach employed[21].

Here are some popular machine learning algorithms. Machine learning entails understanding a function that assigns an intake to an outcome from the reference group created. It creates a purpose with the identified training dataset. Guided machine learning systems need outside help; training and testing datasets are inputted to enable an efficient ML system. Different machine-learning techniques and their implementations[22]:

Supervised Learning has a predictable arbitrary function. All procedures use the learning database to identify or classify the test collection[23]. At all times, the database schema instance

Do Not Copy, Lead City University, Nigeria

has a recognition accuracy. It formalizes the principle of training from input and outcome samples[24]. **Unsupervised Learning** is harder than supervised Learning. We tell the machine to learn whatever we do not teach it. This strategy does not classify but maximizes benefits. Self-organized neural networks employ machine learning techniques to recognize clues in data samples. Unsupervised Learning improves text annotations. Unsupervised Learning identifies system stages and transformations. **Semi-supervised Learning** combines supervision with unsupervised intelligence. Unsupervised Learning is accessible; however, results of data analysis must be found[25]. In **Reinforcement Learning**, selection strategies rely on activities made. The student only knows what to carry out once a circumstance arises. Learning behaviors significantly affect scenarios[26].

2.5 Dimensionality Reduction (DR)

This is an essential approach in numerous disciplines, particularly information processing, advanced analytics, reinforcement learning, object recognition, and knowledge discovery. In a variety of everyday data analyses and visualization, high-dimensional information is often present and in order to effectively control the knowledge, its redundancy must also be decreased. This is an aspect of data processing which increases the classifier's durability and decreases its use. High-dimensional information is challenging for predictive models to process owing to its high computational complexity and cache consumption. Extraction of features (FE) and feature selection (FS) are two-dimensionality reduction strategies. FE is also referred to as image compression expressly or information modification[27].

The benefit of FS is that any knowledge regarding the significance of a particular application is lost; nevertheless, if a restricted collection of attributes is demanded and the key features are quite complex, data will be lost since some characteristics must have been discarded even during feature ranking operation. In general, extracting the features is commonly capable of

Do Not Copy, Lead City University, Nigeria

reducing the dimensionality of the feature set despite compromising a significant portion of data from the high-dimensional feature space. The decision combining feature detection and feature selection processes is dependent upon the device's data structure and context[28].

2.5.1 Feature Extraction (FE)

Data feature lessening is the modification of high-dimensional collected information into a lower dimension. With the tremendous expansion in resistance to high, the usage of methods to reduce dimensionality in a variety of settings has become widespread. Moreover, current technological ways emerge continuously. All the processes are called data preprocessing strategies which in essence turns an increased collection into a small information while preserving as much true context as feasible. The scalability scourge is mitigated by the primary system's low-dimensional rendition. The low-dimensional knowledge is straightforward to examine, manipulate, and examine. Certain advantages can be realized when feature extraction approaches are used for information[29].

i. Information loading space can be reduced as the number of proportions decreases.
ii. It only takes a short amount of time to compute.
iii. Data that is redundant, inappropriate, or noisy can be removed.
iv. Data quality can be enhanced.
v. Some procedures do not execute well when the number of dimensions is increased. As a result, reducing these dimensions allows a procedure to work more proficiently and accurately.
vi. Visualising data in higher dimensions is difficult. As a result, dropping the dimension may permit us to project and scrutinise shapes more evidently.
vii. It facilitates and improves classification. Computational complexity reduction occurs using the selection of features and feature reduction. Some variables should be omitted throughout the selection of features, which reduces knowledge. Feature

Do Not Copy, Lead City University, Nigeria

extraction can minimise dimensionality without diminishing the basic feature collection.

2.5.2 Feature Selection (FS)

This is a dimensional reduction approach used within data mining and knowledge extraction to remove irrelevant or redundant features whilst maintaining separability, reduced information transmission and better data mining. It reduces packet losses, maximum throughput, and memory. Feature selection is a significant subject in learning algorithms and appropriate resources since it removes redundant or disruptive characteristics, enhancing information quality and learning processing capacity. The rapid growth of digital image databases spurred Content-Based Image processing techniques, which require effective search algorithms. Typically acquired low-level visual aspects include coloration, pattern, and structure. The selection of features uses a smaller portion of the original variables. The selection of features removes features that provide scant or no predictive information whilst preserving classification performance. High-importance features cannot be eliminated without lowering categorization accuracy. Poor realization of a characteristic can boost the accuracy of classification[30].

Feature Selection Issue

Choosing the most essential features is a significant component of resolving classification and regression problems, especially when it comes to identifying cursive. That's because it takes time to search for all feasible subsets of parameters that can be made from either the existing batch, each feature is crucial for the least selection of the inequalities, and there aren't too many differences respectively cross-functional and cross and inter-class. After a specified point, adding more features makes efficiency progressively worse instead of effectively[31].

Do Not Copy, Lead City University, Nigeria

The filtering method (FM) and the wrapper method are the two main ways to get rid of different factors in an image (WM). FM is mostly a way to find the best attributes before they are processed. In this method, vastly graded attributes are used as predictive variables. In WM, predictors are wrapped in an optimization technique that selects a subgroup and gives it the maximum prediction models possible. Advanced optimization procedures have trouble finding information within large sets of data. So, optimization procedures like genetic algorithms (GA) and particle swarm optimization (PSO) or progressive exploration schemes have remained developed. They work well and are easy to use. Wrapper methods have two kinds of algorithms: those that use heuristics to find things and those that pick things one at a time. Based on how methodologies are used in the classification algorithm, FS can be characterized into three distinct parts: The Filter, Wrapper method, and the Embedded approaches[32].

Filtering Methods (FM)

FM is employed to figure out the information's criterion appositeness ranking (AS). This process occurs just before the trained model is used in any way. The AS is used to put all of the features in order, and attributes with poor ratings are thrown out. Univariate (UV) and Multivariate (MV) are two types of AS methodologies or strategies that can be used (MV). The UV research evaluates AS for each feature separately, while the MV technique builds information exchange among features slowly and is less flexible than the other two. UV procedures here include -Test, the 2, Mutual Information, Fisher's Discriminant Ratio (FDR) (distinctive Proposition testing), and Similarity, among others. MV techniques, on the other hand, are including mutual information methodologies or variation analysis (ANOVA)[33].

Wrappers Method (WM)

Do Not Copy, Lead City University, Nigeria

Each attribute gets a certain number of points. This rating helps decide on the most efficient and effective attribute, but it is usually just a reasonable estimate of the attributes in the approach that was recommended (PM). PMs are the people who try to guess what will happen. Some examples of PM are SVM, Boosted Trees, GLM, Random Forest, MARS, Multilayer Perceptron, CART, AVAS, Ordinary least square, and Linear Regression[34].

Embedded Method (EM)

The EM method takes the best parts of both FM and WM methods. EM takes personal qualities out of or adds them to the feature subset as the model is built and interpolated. When the probabilistic model is made, EM selects the attributes, while wrappers are using space for all groupings of features. Because of this, EM makes better use of data. It also lets the model find the optimum solution subsegment extra fast. Random forests, decision trees, and SVM include some of the most prevalent empirical models[35].

2.6 Optimization Techniques

This section deals with all techniques that have been used and could be used as algorithms for Dimensionality Reduction. Several scholars have used one or a mixture of these to achieve feature extraction or/and feature selection.

2.6.1 Genetic Algorithm (GA)

Genetic algorithms (GA) are an adjustable metaheuristic optimization method used to recognize estimated best solution to optimization difficulties with a vast search process and can be advantageously employed in the identification of optimized attributes[36]. In GA, a chromosome-like structure called a "person" is used to incorporate data about a possible answer to a problem. Algorithmically, a huge number of people are grouped to form a population, and then the GA optimization procedure is applied to that population. A

Do Not Copy, Lead City University, Nigeria

chromosome is the principal provider of genetic information, consisting of a grouping of chromosomes. The visible representation of a person's shape is dictated by the combinations of specific chromosomes, but its interior development is regulated by the environment[37, 38].

The concept of the natural collection serves as the conceptual basis for GA, a search strategy. A GA optimization technique has four main components namely: a population of participants (or chromosomes) that each symbolize a possible answer, an adaptive threshold feature, an efficiency increased to determine which participants will go on to generate the next stage of evolution, and a genetic integrator like crossover or mutation to probe the uncharted territory of the new search process[39].

To indicate an alternate, a set of qualities (chromosomes or genomes) is employed, which would ordinarily be defined by binary character string 0's and 1's. To generate a brand-new generation, the chromosomes of all individuals are transformed by genetic mutations and crossovers, and the most adaptable members of the present population are picked for each cohort. The method ends when the resulting number is at or over the maximum value supported.

Here are some advantages of genetic algorithms: increased resilience and performance in optimization algorithms; streamlined complexity; easy implementation[40].

GA is commonly employed in real-world issues because of its many advantages. An improved verification strategy is guaranteed by using GA in the dynamical airflow arrangement to identify an extremely dependable BES model (enhancing energy imitation environment) that perfectly encompass the current properties of structures[41]. GA can sometimes be utilized to tackle big issues because of its faster computation time and enhanced degree of integration. However, GA can sometimes be utilized for substantial equations and

Do Not Copy, Lead City University, Nigeria

must instead be applied to solving simple problems. Genetic algorithm refinement and improvement are key to finding a long-term approach to this problem[42].

Workable alternative encryption, activation, efficiency measurement, closure constraint verification, selections, crossovers, and mutations are the phases of GA sequencing. The features shown by {α^1, α^2, …, α^n} are the original ones. To begin, it generates a numeric encoded for each chromosome that stands in for extracted feature combinations that could help solve the problem. During the introductory stage, an arbitrary beginning population {β^1, β^2, …, β^n} is formed, and the population size is established. Finally, the suitability per each chromosomal is determined by applying the fitness function that has been established. The optimization process is a metric for measuring how well a set of chromosomes performs. One of the most important determinants of GA functionality is how fitness functions are defined[43]. In summary, what genetic algorithm does is generate a population of candidates for a problem's solution and then put that population through evolutionary pressure.

Analyzing the population to discover the most likely answers requires a evaluation of the capability of solutions (individuals) (best solution to the problem) to reproduce, certain individuals are favored over others through the process of selection. How suitable a solution is directly correlates with how likely it is to be picked. - Crossing: recombining some of the features of the selected solutions to create new persons. - Mutation: the process by which the features of offspring are changed, hence increasing genetic diversity. Individuals of the current generation have been integrated into the population; - Completion: it is determined if the prerequisites for the conclusion of development have been met, and either the process is restarted, or the evolution is halted if they have not. Genetic algorithms are depicted in Figure 2.1. Further, the pseudocode for a traditional genetic algorithm is included in Algorithm 2.1.

Do Not Copy, Lead City University, Nigeria

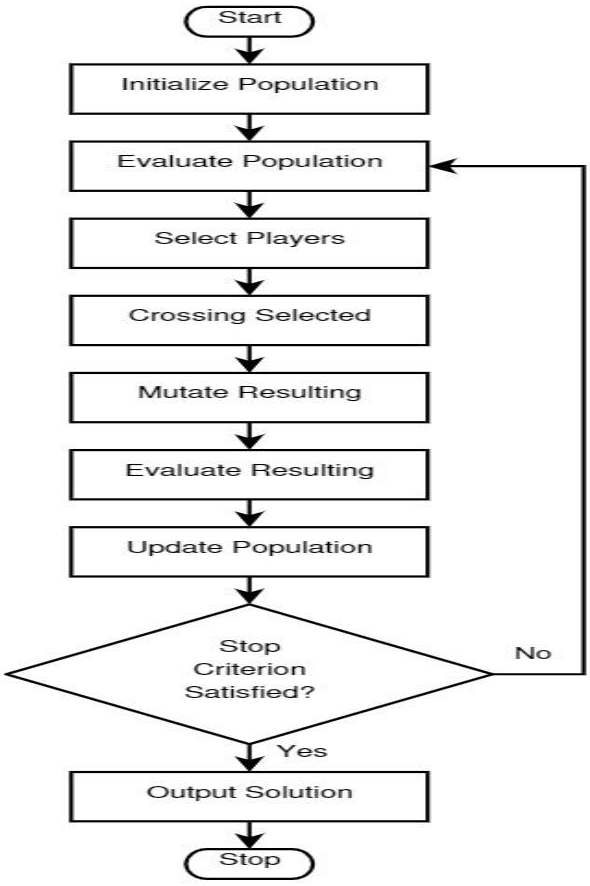


Figure 2.1: Genetic Algorithm Structure [44]

Algorithm 2.1: Genetic Algorithm

Input:

Population Size, n

Maximum number of iterations, MAX

Output

Global best solution, Y_{bt}

begin

Generate initial population of n chromosomes Y_i (i = 1,2,……,n)

Set iteration counter t = 0

Compute the fitness value of each chromosomes

while *(t< MAX)*

Do Not Copy, Lead City

Select a pair of chromosomes for initial population based on fitness

Apply crossover operation on selected pair with crossover probability

Apply mutation on the offspring with mutation probability

Replace old population with newly generated population

Increment the current iteration t by 1.

end while

Return the best solution Y_{bt}

end

Genetic Algorithm[45]

2.6.2 Ant Colony Optimization Algorithm (ACO)

In swarm-based searching, ACO serves as a solution algorithm and is therefore a discrete heuristic algorithm. It functions similarly to how real ant colonies act when searching. The algorithm steps are outlined below[46].

The ant searches randomly for food in the region. Artificial pheromones were left behind after the ants brought the materials back to the cavern. There was a drastic improvement in signal concentration from higher-to-higher sample sizes. The pheromone trail is used by other ants to locate the source of the sampling. First, pheromone trails need to be set up. Each ant then comes up with a solution that depends on the pheromone's value utilizing the deterministic conditional probability rules. Moreover, the proportion of pheromones varies with time, going through an "evapotranspiration" step in which some of the pheromones are lost and an "intense" phase where every ant has a big stockpile of hormones; this calculation indicates the system provides versatility. Repeated until the conditions are no matter how many years, the procedure is continuous[47]. The ACO algorithm is a reiterative development, as shown by the flow chart in Figure 2.2.

Do Not Copy, Lead City University, Nigeria

Algorithm 2.2: Ant Colony

Input: Original pheromone paths values.

Output: Finest result initiated or a conventional result.

Replicate for each ant do

Solution construction using the pheromone trail;

Update the pheromone trails:

Evaporation;

Reinforcement;

end

until Ending conditions

Ant Colony[48]

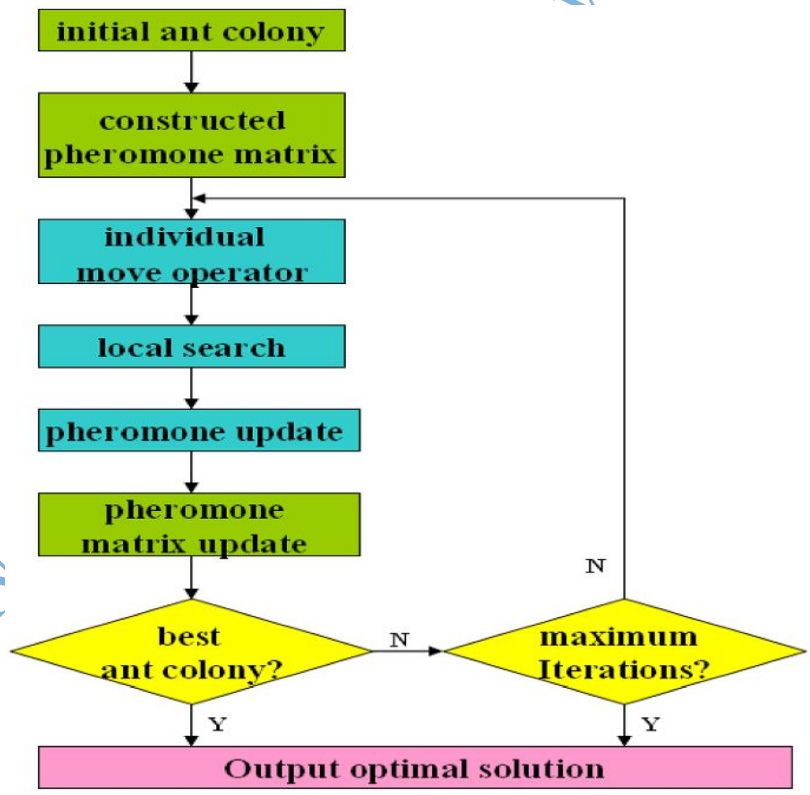


Figure 2.2: Flow diagram for ACO [49]

versity, Nigeria
Do N

2.6.3 Factor Analysis (FA)

The primary idea of FA is to categorize the observations information, grouping those that are highly correlated together into the same group. This reduces the degree of correlation seen between individual groups of factors, and hence the degree of irrelevance among them. The outputs of the multidimensional platform, as well as the impact of the many elements on the framework, may be described, and the optimal group for each of the characteristics can be determined using the knowledge included in the subset[50].

The Principal Component Analysis (PCA) technique and the FA methodology differ in that the former makes the assumption of the perceptible arbitrary vector (aI = a1, a2, ldots, an), while the latter assumes the unmeasured vector (Vj = V1, V2, ldots, Vm) in the scheme aI = math op summation limits j = 1m (n > m), where (a ij) is the factor Public factors (V) are theoretical variables that occur in the representation of all unique participants to different but cannot be directly observed by researchers. The variable (var epsilon I stand in for the weight of the singular factor (c j), and the variable (aI influences the singular factor (c j) (var epsilon i). The factor loading is found by computing the linear relationship between the parameters, and this is the central issue in FA. Let's pretend (A) is the composite reliability matrix, defined as A = (a ij) n * m [51].

To perform dimensionality reduction, first, identify the number of latent variables using the value of (alpha), and then compute the factors' synthesis score. Dehak recommended a language that is extremely depictive for use in supervised methods. In this approach, a low point space, reliant on both the device and the network, is generated. This region is dubbed the "may choose region" since it accounts for variations in both the speaker and the medium [52].

Do Not Copy, Lead City University, Nigeria

2.6.4 Artificial Bee Colony Algorithm (ABC)

This is an advanced procedure for minimizing or optimizing a function through communication between individuals in a community. Bees can be categorized as workers, observers, or scouts. Half of the bees are worker bees who are dedicated to a specific flower or crop (a current solution). The other half consists of bystander bees, which will approach the source of food in ratio to the nectar associated directly by the worker bees, so boosting the predation operation. After a fixed number (m) of iteration, if the bees haven't found a way to improve their lives, the worker bee is replaced by a scout bee, which flies off in seek out a new prey species. Therefore, the ABC relies on three adjustable parameters: swarm size (NP), the maximum number of failed visits (limit), and sampling interval (maxCycle). The steps in the method are as follows [53]:

i. Spread the scout bees out randomly among the first food sources. A scout is hired at the i*th* point, denoted by the D-dimensional vector Xi.

ii. Employed bees can be sent in search of food, with the quality of their solution being measured against a cost metric.

iii. Estimate how likely it is that bees passing by will go check out the source of food. i. SN is the whole sum of worker bees and G(i) is the fitness purpose derived from g, Pi is distinct as G(i)/ SNj=1 G(j). Let the curious bees loose in a fair selection, and then rate their answer quality.

iv. Deploy foraging bees to discover undiscovered food sources.

v. If the conditions are not met, think back to the best food source you've located this distance and go back to phase 2.

vi. This formula describes the function G(i): G(i) = 11 + gIgI01 + abs (gI gI 0).

vii. At each subsequent call (phases 2 and 4), efforts are made to refine the precise position of the food supply.

Do Not Copy, Lead City University, Nigeria

If j is a measurement selected at arbitrary, then is a consistently circulated arbitrary sum in the series [1, +1], and k = I then the i*th* exploited bee positioning is maintained using the input of a selected at the random employed bee with index k. A greedy approach occurs when the answer is improved, indicating X′i as the new food basis; alternatively, the solution is rejected, and the position remains the same. When Xi,j and Xk,j are vectors, then X′i,j = Xi,j Plus (Xi,j Xk,j) (2) After m iterations of probing the same food source without success, the worker bee linked with that source turns out to be a spy bee and flies to a place chosen at random within the search area. As a result, with a limit of one scout every iteration, the worker bees will carry out the investigation, leaving behind depleted sources and introducing undiscovered sources [54].

Algorithm 2.3: Artificial Bee Colony

Begin

InitPopulation()

As the rest of the iterations

choose local search sites

To assess fitness and populate the desired locations,

bees must be recruited.

Find the healthiest bee and pick it.

Put the rest of the bees out

look for whatever you want

Assess the viability of the surviving bees.

Until UpdateOptimum() Is Called,

Return the best solution

End

Artificial Bee Colony[55]

Do Not Copy, Lead City University, Nigeria

2.6.5 Bee Colony Optimization (BCO) Algorithm

An instance of a demographic approach is the BCO. It was first suggested, and later implementations have further refined and improved upon the original concept. The original iterations of the algorithm were more faithful recreations of natural bee activity. Scout bees played a significant part in these variants, hive locations were prioritized, and the recruiting process was more analogous to the actual one than in the present algorithm. Here, we'll give a thorough rundown of the current iteration, pointing out key distinctions while also demonstrating some practical uses. The optimal solution is found by a cooperative effort from a swarm of agents (robotic bees) that are all B bees. Only one answer to the problem is produced by each synthetic bee. The respective step of the BCO procedure consists of a frontward and a regressive pass, which occur at regular intervals. In every forward pass, the artificial bees investigate new areas of the search space. It uses a predetermined amount of steps to build and/or enhance the solution, yielding a different answer. Bee 1, Bee 2, ..., and Bee B should all be able to vote on n different entities. A bee should only forward pass to a single target at a time[56].

The second stage, characterized as the backpropagation algorithm, begins once the bees have returned to the hive with their newly acquired optimization algorithms. During the reverse phase, all of the artificial ants discuss the efficacy of their suggestions. In the wild, after bees find a good food patch, they will return to the hive to alert the other bees of the abundance of the patch and its accessibility to the hive through a spinning routine. The search algorithm uses bees to broadcast the optimizer desirability or the number of the optimization problem. After considering all the options, each bee makes a probabilistic decision as to whether or not it will stick with its current answer. The bees who come up with the best solutions are the ones most likely to keep them and spread the word. Artificial bees that are faithful to their incomplete answers are recruiters, in the sense that other artificial bees take their solutions

Do Not Copy, Lead City University, Nigeria

into account, in comparison to their natural counterparts. Once a bee gives up on a solution, it is no longer committed to it and must instead pick one of the promoted alternatives. This determination is likewise based on frequency, so that more promising promoted solutions are more likely to be pursued. At the beginning of each regressive pass, the bees are split into clusters “R recruiters, and residual B-R indifferent bees”. The principles of R and B-Rare shift from single round of retrogression to the next[57].

Forward and backward pass phases of the search process alternated to produce all possible solutions (one for each bee). One iteration of the BCO is accomplished when the finest solution is found and utilized to apprise the total best result. Here, we get rid of all the B solutions and start a new iteration. Until a termination condition is reached, the BCO will continue to iteratively loop. Conceivable stopping measures contain reaching the maximum allowed CPU time, reaching the maximum allowed the sum of frontward/regressive passes without enhancing the target function, etc. Once all other solutions have been exhausted, the best one (the "global best") is reported. The following are the required settings for an algorithm before it can be run: The number of bees in the beehive is denoted by the letter B. NC - The total number of forwarding passes in which all positive actions were taken. At the outset of the search, all of the bees are safely within the hive. Rendering to the central concept of the most up-to-date implementation of the BCO algorithm, the hive is an abstract entity without a fixed position that has no bearing on the algorithm's performance. Its sole function is to designate the checkpoints at which the bees meet to discuss the progress of the search in real-time [58].

Do Not Copy, Lead City University, Nigeria

Algorithm 2.4: Bee Colony Optimization

1. Begin an empty solution provided to each bee.
2. For each bee: / the preliminary round
    i. counter positive forward passes by setting C = 1.

ii. consider all potential positive actions, and

iii. make the decision based on an appraisal by spinning the roulette wheel;

It might be written as follows:

iv. c = c + 1; If c = NC,

continue to ii.

3. every bee has flown back to the hive, and the reverse pass has begun.

4. compute the (partial) value of the bee's objective function.

5. Every bee makes a haphazard choice between continuing its independent research and becoming a recruiter and becoming a follower.

6. Choose a different answer from the recruiters using the roulette wheel for each of your followers.

7. Step 2 is taken if all solutions have not been implemented.

8. choose which option is the most viable, then implement it

9. If the requirement for stopping is not met, continue with step 2;

10. Show the best answer that was discovered

2.6.6 Firefly Algorithm

The firefly algorithm (FA), advanced in 2008, is stimulated by the emotional instability and dazzling light displays of tropical fireflies. FA is easy to understand and execute due to its flexibility and simplicity[59]. At zero range, the desirability of two fireflies is 0, therefore the motion of one firefly, I, is driven to the motion of another, brighter firefly, j. Third, we have diversification, where x_i is a factor for how much to shuffle the deck, and ti is a vector of pseudo-random taken at time t from a Gaussian kernel.

In other research, like Lévy flights, randomized is defined in terms of t I, which may be easily generalized to cover a wider range of distributions. We have extensively analyzed the

Do Not Copy, Lead City University, Nigeria

firefly method and its many variations. From this, we may deduce that mutation is employed in both local and global searches. Larger scale mutation occurs when t I is selected from a Distribution function and Lévy flights. Nonetheless, if it is set to an extremely small value, the resulting transformation will be tiny and confined to a narrow region of space. Interestingly, the selection in the algorithm is implicit rather than apparent because g is not employed in FA. Updates in FA's two loops make use of ranking and selection logic, nevertheless. Using attraction is a completely new concept, and it has never been used before in any SI-based algorithm until FA. FA's population can naturally divide into subgroups, each of which can swarm towards a local mode due to the greater strength of long-distance attraction compared to a local attraction. The real optimality of the problem is always the global best approach from all the local modes. FA is a natural and economical solution to multimodal challenges[60].

Do Not Copy, Lead City University, Nigeria

Algorithm 2.5: Firefly (FA)

Objective function f(x), $x= (x_1, \ldots\ldots\ldots, x_d)^T$

Generate an initial population of fireflies x_{ik}, i = 1, 2, ..., n and k = 1, 2, ……., d

Where d = number of dimensions

Maxgen: Maximum no of generations

Evaluate the light intensity of the population I_{ik} which is directly proportional to $f(x_{ik})$

Initialise algorithm's parameters

While (i<=n)

While (j <= n)

If ($I_j < I_j$)

Move firefly i toward j in d-dimension using Eq. (4)

End if

Attractiveness varies with distance r via

exp[-r^2]

Evaluate new solutions and update light intensity using Eq. (1)

j= j+1

End while

Rank the fireflies and find the current best

if stopping criteria is satisfied then stop

else i = i + 1

End while

Post process results and visualization

Firefly (FA)[61]

2.6.7 Particle Swarm Optimization (PSO)

It employs a swarm of probes to help find what you're looking for. In PSO, particles remember their positions in the search space and how they relate to the optimal answer. My favorite place in the world is right here (pbest). However, the best global position for a particle is the one where its fitness is maximized (gbest). Based on its own and its neighbors' flight histories, a particle will adjust its position in the search domain and its velocity to head in the direction of its pbest and gbest locations[62]. The following basic elements must be considered when implementing the PSO [63]:

In mathematics, a particle is a candidate solution denoted by a vector.

Swarm: A group of moving objects where individual particles appear to be traveling in different directions.

Location optimal for a particle: At each position in the search space, a particle's fitness is compared to the highest value it has ever achieved.

Do Not Copy, Lead City University, Nigeria

Position at the top of the global leaderboard, as determined by the aggregate of all previous first places.

Particle velocity refers to the rate of motion of individual particles. After calculating the best local and global placements, the particle's speed is adjusted accordingly. When a new generation's updated velocity is calculated, the positions of all particles are reset [64]. The PSO algorithm's search procedure is as follows:

Algorithm 2.6: Particle Swarm Optimization

The first step is to define the parameters of the PSO and the constraints on the choice variables of the optimization problem.

The second step is to generate a population of particles moving at random speeds and locations.

Third, evaluate the health of each particle.

Fourth phase: Iterate until the endpoint is reached.

(a) Evaluate each particle's fitness with its pbest. Keep the greatest as your pbest.

(B) Evaluate how the current gbest rank stacks up against the previous gbest rank.

Each particle's velocity should be updated using Eq (1).

(d) Modify the location of each particle with Eq (2).

Fifth, have the world's best particle coordinates and fitness level printed out.

End

Particle Swarm Optimisation[24]

2.7 Methods of Hybridization

The standard method for creating hybrid algorithms is through iterative trial and error. Thus, hybridization can be seen as a metaheuristic strategy for evolutionary change. Selecting two or more optimization algorithms or any algorithm at all at random from a collection of algorithms (both standard and novel) yields a basic method of hybridization. Hybrid

Do Not Copy, Lead City University, Nigeria

procedures, for instance, ABC-HS, SA-PSO, DE-PSO, and numerous more, can be created from a collection of existing algorithms such as PSO, BA, DE, ABC, ACO, CS, FA, HS, FPA, SA, then hill climbing. However, a hybrid's performance is often inconsistent; certain aspects may improve while others may worsen if such a simplistic method is employed [65].

To develop superior hybrids, familiarity with algorithm fundamentals is required. Nevertheless, this heavily depends on the skill and knowledge of the algorithm developer. Important algorithm operators include crossover, mutation, random walks, elitism, Lévy flights, gradients, and chaos. An algorithm can be improved upon by including these aforementioned operators, Also, an algorithm can be fine-tuned by including one or more new parts. Consequently, some scientists have developed PSOs, genetic algorithms[66], and so on, that incorporate elements of chaos. Look more attentively at these underlying elements. They can be broken down into four distinct classes[67]:

When talking about biological controllers here, Elitism can take the form of crossover or recombination, mutation, or selection.

The use of randomization. Some examples of probability distributions are the random walk, the Lévy flight, and the Gaussian distribution.

Characterized by an absence of predictability. Maps that are iterated over, and utter anarchy.

Both attraction and repulsion. Light intensity, gravity, electromagnetism, attractiveness, and other forces based on distance or resemblance attract one another.

Reasons for aversion include unfamiliarity, threat, antagonism, and variety.

There is a wide range of effectiveness amongst categories, and each can serve a unique purpose in terms of exploration and exploitation potential. Theoretically, a hybrid can be created by selecting one component from two or more distinct groups based on their

Do Not Copy, Lead City University, Nigeria

characters or qualities. This strategy has the potential to yield superior algorithms to those obtained from a simple mix and match of any two algorithms[68].

2.8 Feature Extraction

After the preprocessing stage of a character recognition system, features are extracted. Accurately classifying input patterns into one of several target classes is the major focus of pattern recognition. Feature extraction is crucial in any pattern classification process since it helps identify the most distinguishing characteristics between classes. In this process, feature vectors are constructed by extracting salient features from objects and alphabets. Classifiers use these feature vectors to effectively match input and desired output units[69]. It is thanks to this data that the classifier can make such fine distinctions.

The term "feature extraction" is used to describe how important data is picked out of a vast dataset. Feature extraction is the development of identifying and removing the unique characteristics that give a character its physical appearance. Each character is given a distinct fingerprint in the arrangement of a feature vector extracted during the feature extraction procedure. Extracting a set of features that exploits recognition amount with the fewest number of components is a primary goal of feature extraction, as is developing feature sets that are identical for different instances of the same symbol. Template matching, deformable templates, and unitary image modifications are some of the most frequently used methods of feature extraction[70]. Descriptors such as the Fourier transform, spline curve calculation, gradient, and Gabor features, Zernike moments, silhouette contours, partitioning, symmetrical instant invariants, and zernike instants. There needs to be a representation of the data that makes it easy to perform some sort of analysis on it later, be it pattern recognition, denoising, information density, imagining, or others. Finding a positive variation is now possible using one of several proven primary procedures. Numerous feature extraction algorithms exist and

Do Not Copy, Lead City University, Nigeria

are used routinely, including principal component analysis, partial least square, independent component analysis, and linear discriminant analysis[71].

Significance of Feature Extraction (FE)

To get features, FE approach is useful to the sectors after pre-processing and the necessary subdivision level (word, mark, letter, or sign) obtained. This is surveyed by the classification and post-processing application procedures. Due to its obvious effect on the identification system's efficacy, the feature extraction phase warrants the utmost attention. "Extracting from raw data information that is most suited for classification purposes," even though "minimizing within class pattern variability and increasing between class pattern variability," is the description of feature extraction. Consequently, attention must be reserved while indicating the right FEn method based on the input to be used. Because of this, it is important to investigate the numerous feature extraction methods applicable to a specific domain and their applicability to a wide range of use cases [72].

2.8.1 Principal Component Analysis (PCA)

Reducing the independent variable numbers in a set of data is one goal of the data mining approach known as dimensionality reduction. Common definitions of dimensionality reduction include an assortment of features constructed from a subset of all features with removal function constructed from the combination of preexisting features to generate a new subclass of the groupings. Feature extraction using PCA is a common practice[73]. The following principal component is the orthogonal linear grouping with the next highest alteration to the initial PC. The number of determinants is comparable to the number of constraints. Many datasets can have their remaining PCs discarded with little loss of information since the initial few PCs describe so much of the modification. Since variance grows in proportion to the size of a variable, it is usual practice to first normalize each

Do Not Copy, Lead City University, Nigeria

variable to have a zero mean and one standard deviation. Since becoming institutionalized, any differences in initial variable measurement have been standardized. We can anticipate reliable information from the empirical covariance matrix[74].

PCA is the most effective direct dimension reduction approach in terms of mean-square error. Since it utilizes a grid based on the variables' co-variances, we classify it as a second-order technique. In different contexts, this technique may be referred to by several names, including the Karhunen-Loeve transmute, the Hoteling transmute, and the experiential orthogonal function (EOF) technique. By locating a small number of impertinent linear combinations (the PCs) of the first variables with the highest alteration, PCA aims to lessen the data dimensionality. The first principal component (PC), denoted by s1, is the linear grouping with the greatest degree of variability. where u1 = (u1,1..u1,q) S denotes a p-dimensional coefficient vector[75].

To summarize, PCA is a statistical analytic procedure that exploits feature validity to reduce a huge number of feature indicators to a trivial number of comprehensive indicators. The goal of principal component analysis (PCA) is simplification, and PCA accomplishes this by allowing the innovative intricate attribute to be recognized by a few unified factors that indicate the relevant data evidence in the innovative variable as significantly as conceivable but have no relationship with one another. There are (np) observations since each sample measures a different number of indicators, but the indicators often interact with one another, hence the purpose of PCA is to investigate how to extract the principal components from the indicators[76].

Using PCA, we may reduce a massive dataset (containing n correlated variables) to a manageable size (containing m highly uncorrelated components or factors; mon), all while

Do Not Copy, Lead City University, Nigeria

keeping as much of the original dataset's variation as possible. Respective component is a linear grouping of the original variables, and PCA determines their eigenvalues and eigenvectors. The first factor is largely responsible for the total variation, the second for the remaining fraction of the total, and so on. There is a continuous scale from 0 to 1 that represents the variance of each factor. If you give it a number, say 0.9, you can pick a subset of components whose cumulative variance is equal to or larger than 0.9, which will explain 90% of the modification in the full dataset. A subset of the original variables might be more relevant to investors than the components that principal components analysis (PCA) often yields [77].

Algorithm 2.7: Principal Component Analysis

1: Train PCA

2: Calculate point creation matrix: YS Y = QN i=1(yi − μ) S (xi − μ)

3: Eigen-analysis: YS Y = UΛUS

4: Calculate eigenvectors: V = YUΛ− 1 2

5: Retain precise amount of first components: Ve = [v1, . . ., ve]

6: Calculate d features: Y = Ud T X

Principal Component Analysis[78]

Analyze the Principal Components (PCA)

An unsupervised analysis is performed using PCA. The information it generates is standard, and the covariance matrix is diagonalized. Orthogonal variation is used to convert the standard inherent correlation coefficient characteristic into linear predictor variables. One challenge with linear dimensionality minimization approaches is the concentration of noise in a smaller subset of the dimension. To filter examples and boost the possibility of illustrating,

Do Not Copy, Lead City University, Nigeria

PCA is used. Popular applications of principal component analysis (PCA) include dimensionality reduction, attribute extraction, data compression, and visual analytics[79].

PCA defines the principal dimensions of the subspace by capitalizing on the uncertainty of static results. An illustration of the experimental value produces a function vector of the observed values in this major subspace. An eigenvector ui, of matrix S, is produced by the vector that maximizes the change in the predictable information, while an eigenvalue I, is produced by the vector that produces the largest variance size along the route of the eigenvector. Eigenvectors with the best eigenvalues in the M-bit representation of matrix S make up the primary subspace obtained via principal component analysis[80].

Concerning recognition systems, feature extraction is the most crucial step. Recognizability typically improves with the deployment of effective feature extraction methods. Ultimately, the purpose of feature extraction is to yield an effective illustration of the full image based on the extracted features.

A linear transform is used in statistics; Principal Component Analysis (PCA) extracts key patterns from large data sets. Initially developed by Pearson, it quickly gained popularity. Character recognition, data compression, and facial recognition are just a few of the many patterns' recognition uses for this algorithm. In this study, PCA was utilized as a global statistical text feature extraction technique to excerpt and choose the imperative features of images before classification with SVM classifiers. To work with smaller images, feature extraction techniques like PCA are frequently employed. The data mean matrix is the first thing that PCA does. The covariance of the information is then calculated. Next, estimates are made for the Eigenvalues and Eigenvectors. The PCA looks for the direction in space that top arrests the largest alteration in the information. To transform data from high-dimensional space (A = a1, a2, ..., an, where n is the number of models and ai is the i*th* comment, sample,

Do Not Copy, Lead City University, Nigeria

or pattern) to low-dimensional space (PCA space, W), this definition provides a PCA space (W)[81].

Feature extraction and dimension reduction using Principal Component Analysis have proven useful in several standalone character recognition systems. The PCA is employed in the subsequent six phases to extract and select characteristics from blurred images[82]. In the first step, the two-dimensional blurred image is transmuted into a one-dimensional vector by joining all the columns and rows of the corresponding two-dimensional matrix. A vector, px, is the rate of picture element Xi in the sample image, and T is the transpose of the vector set. The next segment is to compute the average of the image. Third, locate the middle of the picture by calculating the wi. Four, locate S, the covariance matrix. The connections between two or more dimensions are evaluated by this matrix. Fifth: Find the Eigenvalue and Eigenvector of S. Afterward, the eigenvectors are ranked according to the eigenvalues with which they are associated. Sixth, pick the largest eigenvectors, W = v1... vk, based on their eigenvalues. The values for W were picked such that they would fall within the PCA's projection domain. Then map the W values onto the PCA's low-dimensional space.

2.8.1.1 Kernel Principal Component Analysis (KPCA)

Several methods and kernel functions have been explored as possible PCA extensions to deal with non-linearity. Before executing PCA on the data, a kPCA first maps the samples into a high-dimensional kernel space, where the data can be transformed from a nonlinear to a linear distribution. KPCA is based on the premise that the original input vectors can be transformed into a high-dimensional feature space F using a non-linear function, and then the linear PCA can be calculated in feature space. The covariance medium in F is given by CF = 1/m i=1m (xi)(xi) T, where xi is one of the input vectors (1, 2, 3, ...) in Rn[83].

Do Not Copy, Lead City University, Nigeria

The magnitude of the kernel matrix in kernel PCA is proportionate to the square of the number of instances in the input set of data, which is just one of the numerous major limitations. In addition, keeping large pairwise distances are a top priority for Kernel PCA[84]. By contrast, the covariance matrix is not used in kernel PCA; rather, the primary eigenvector of the kernel matrix is calculated. The kernel matrix K is a representation of the information points xi. Where k is the kernel purpose, the entries of the kernel matrix are defined as cij = c(ai,aj). Because the kernel matrix is most analogous to the process by which data points are generated in high-dimensional space when a kernel function is used, PCA can be reformulated directly in kernel space. Applications of kernel PCA in areas as diverse as face recognition, speech recognition, novelty detection, and more have been met with great success[85].

Algorithm 2.8: KPCA

Initialization $m=1$, $w_1 = 1$, $K_1 = {}_k(x_1,x_1)$, $\beta_1 = 1$

At each instant $t \geq 2$, upon acquisition of x_i

1. Compute k (x_t) $k(x_t) = [k(x_{w1}, x_t) \ldots. k(x_{w1}, x_t)]^T$
2. Subspace representation of $k(x_t,.)$ $\beta_t = K_m^{-1} k(x_t)$
3. Compute (square) distance to subspace $\epsilon^2_t = k(x_t, x_t) - (x_t)^T \beta_t$
4. IF the distance criterion is satisfying: $\epsilon^2_t \geq v$

   Increment the model order $m = m + 1$, $w_m = t$, ${}_t = [\alpha_t^T \ 0]^T$

   Update the inverse of the Gram matrix, $K^{-1}{}_m = \begin{bmatrix} k_{m-1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\varepsilon_t^2} \begin{bmatrix} -\beta_t \\ 1 \end{bmatrix} [-\beta_t^T 1]$

   and the empirical kernel map $k(x_t)$ $k(x_t) = [k(x_t)^T k(x_1, x_t)]^T$

   Update subspace representation of $k(x_{ti}.)$ $\beta_t = [O_{m-1}^T \ 1]^T$

5. Output $\psi t(xt)$ $y_t = {}^T_t k(x_t)$
6. Update the coefficients ${}_{t+1} = {}_t + n_t y_t (\beta_t - y_t \ {}_t)$

Principal coordinate of any x $\psi(x) = {}^T k(x)$

KPCA[86]

Do Not Copy, Lead City University, Nigeria

2.8.1.2 PCA in Two-Dimension (2DPCA)

After projecting PCA over a larger space in two-dimension, we get 2DPCA. Its core premise is still the same as PCA, though: to project as much of the state-of-the-art data as imaginable onto a small set of primary components while maximizing the variance sum of that set. Image data training set A = AiRmn,I = 1, 2, M, where M is the number of training models, m and n are the rows and column pixel dimensions, and W = [w1, w2, wd] is the feature matrix. If we let Rnd be the feature matrix and d be the number of major projection vectors following transformation, then the unbiased function of 2DPCA can be written as follows. Matrix traces can be represented as[87]:

Maximum Work Time Worked = Iktr(i=1MWT(Ai)TAiWT) = Iki = 1MAiW2F tr(); if there is an n-by-n matrix A, then A's trace is equivalent to the sum of its eigenvalues, or the sum of its principal transverse elements[88].

Algorithm 2.9: 2DPCA

Input: $\mathbf{A}_i \in \mathbf{R}^{m \times n}$ (i = 1, ……, N), k, where **A** is centralized, ϒ=0.00001. Initialize $\mathbf{V}^{(t)} \in \mathbf{R}^{m \times n}$ which satisfies $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, t =1

while not converge **do**

1. For all training samples, calculate $d^{(t)}$ (i = 1, ….., N) by Eq. (8)
2. Calculate $H^{(t)}$ according to Eq. (9), i.e.., $H^{(t)} = \sum_{i=1}^{N} \boldsymbol{A}_i^T \mathbf{d}_i^{(t)} \mathbf{A}_i$
3. Solve $V^{(t+1)}$ = argmax tr($\mathbf{V}^T\mathbf{H}^{(t)}\mathbf{V}$): the column of the optimal solution $\mathbf{V}^{(t+1)}$ are the eigenvectors of $\mathbf{H}^{(t)}$ corresponding to the k largest eigenvalues.
4. Update t ← t + 1

end while

Output: $\mathbf{V}^{(t+1)} \in \mathbf{R}^{m \times k}$

2DPCA[88]

2.8.2 Independent Component Analysis (ICA)

A modern statistical technique, ICA has emerged recently. The determination of this technique is to linearly partition observational information into statistically indiscernible

Do Not Copy, Lead City University, Nigeria

subsets. The core concept of ICA is to employ a model of statistical variables that are left implicit, x = As[89]. ICA is a computational method for decomposing a sign into its constituent parts. As an example of a popular ICA solution, consider the "cocktail party problem," in which test data from people speaking in a room are reduced to basic voice signals. One popular method for making sense of this intricacy is to look for the lack of delays or echoes. If there are N sources, then you must also account for the fact that you will need N approximations (e.g., microphones) to mine the primary signals[90, 91].

This particular statistical model is known by its acronym ICA. What this means is that the visible data is the result of a combination of several distinct components. To further complicate matters, only the random vector (x) may be explicitly observed, the mixing matrix (A) is presumed to be indefinite, and both (A) and (s) must be approximated under certain constraints[92]. The ICA presumes the components are non-Gaussian and statistically independent, and that the unidentified mixed matrix is square. Calculating s = Wx's independent components requires first determining the inverse W of A. For this reason, the ICA model can't rank the variance of components or establish their order of importance[93].

The sparseness of the basis function is employed in an enhanced version of the independent component analysis algorithm for feature extraction in images. Since it does not necessitate the optimization of high-order nonlinear assessment functions, this technique possesses respectable sparsity and a quick convergence speed. Studies in the works that employed ICA for facial recognition showed its vast future potential. Since ICA is a relatively new concept, its theory and algorithm are still in their infancy and several aspects of the field could use further refinement or expansion. Wang introduced the ICA-DR method of dimensionality reduction, which employs mutual information as a criterion for measuring data statistical independence above and beyond second-order statistics. Therefore, the ICA-DR can maintain

Do Not Copy, Lead City University, Nigeria

and retain information that is lost when using dimensionality reduction approaches based on second-order information. Numerous research has employed independent component analysis (ICA) to extract features from microarray gene expression data. A unique (artificial bee colony) ABC-based feature selection strategy was developed consisting of two phases: an ICA-based and an ABC-based wrapper extraction method. One benefit of ICA is the total features removed is proportional to the input examples [94].

Nigeria

Algorithm 2.10: Independent Component Analysis

Step 1: Take the detected signals and center them to get rid of the mean;

Step 2: Normalize the data;

Step 3: Pick an initial value and let it stand for now; Step 4:

Step 4: Select a starting vector at random with norm 1;

Step 5: Revise the Lagrange multiplier;

Step 6: Revise the demixing vector by, where is the rate of learning;

Step 7: Normalize by;

Step 8: When the second increment of or a minus is found, the algorithm is restarted with a new beginning value via the deflationary orthogonalisation method.

Independent Component Analysis[95]

2.8.3 Linear Discriminant Analysis (LDA)

To extract classification information and reduce dimension, LDA's primary goal is to project the unique example onto the finest discriminant vector space, so that the projected data samples have the broadest covariance separation and the shortest inter-class range (determined inter-class scatter matrix and minimum intraclass scatter matrix)[96].

As an extension of PCA, IPCA and Incremental Discriminant Analysis (IDA) are also available. Also, incremental weighted average sample analysis is offered as a novel incremental facial feature extraction method for immediate face identification. It is proposed to employ semi-supervised linear discriminant analysis (SLDA), which allows LDA to accommodate the circumstance of only a small amount of categorized data by training on both the small amount of categorized data and the vast number of unlabeled information. Statistics-free identification analysis and other approaches successfully compensate for the weaknesses of more conventional mathematical techniques[97].

The statistical analysis techniques only look at the data's features or classify data subsets statistically; they don't look at the data's features, examine the data satisfied of the data subclass, or estimate the consistency and efficacy of the model from the information's perspective. Furthermore, there are numerous examples of high-dimensional data where the statistical investigation technique's assumption that the data set statistics are unrelated is not [98].

Algorithm 2.11: Linear Discriminant Analysis (LDA)

LDA (e = {(ySi , zi)}pi=1):

Ei = yUk | zk = di, k = 1, ⋯, n- 1, i = 1, 2 // class-precise subgroups

μi = mean (Ei), i = 1, 2 // class means

C = (μ1 − μ2)(μ1 − μ2)U // among class scatter medium

Zi = Ei − 1niμSi , i = 1, 2 // midpoint class mediums

Ti = ZUi Zi , i = 1, 2 // class scatter matrices

T = T1 + T2 // within-class scatter medium

λ1, x = eigen(T−1C) // calculate main eigenvector

Linear Discriminant Analysis (LDA)[99]

Do Not Copy, Lead City University, Nigeria

2.8.4 Partial Least Squares (PLS) Algorithms

By establishing hyper-plane disparity in the response and independent variable quantity, PLS is a precise method that permits some association to the key regression of the components. Applying both estimated and perceptual variables, PLS can identify a linear regression model in a new space.

Existing dimension reduction approaches, such as PCA, FA, NMF, etc., account for sample variables directly. Their size-reduction ability is great. The typical dimension reduction strategy struggles with small sample sizes since the feature dimension is bigger than the number of examples and the covariance matrix is unique. The PLS technique is introduced and then used on a small sample of dependent variables to reduce dimension. Partial least squares differ from OLS in significant ways (OLS). Its main principle is to consider the correlation of matrix Y as matrix X is compressed.

Let's say n independent variables "x1, x2, ..., xn" and p dependent variables "y1, y2, ..., yn" X is decomposed into Y = UQU + F after some processing. U, Q, and F are the scoring matrix, load matrix, and residual error matrix, respectively. JU is the score vector, u is the weight vector, J is the i*th* column of matrix U, and q is the i*th* column of matrix R. Z VRu + G is the dependent matrix. V = score matrix, R = load matrix, and G = residual error matrix. The previous formula may be written as qk Z = vkrkU + G k= 1, 2, ..., s.

Independent variable (X) and dependent variable (Y) scores (t, u) are extracted independently using the partial least squares technique. In addition, the covariance between the two scores is the highest possible, meaning that they account for the determined amount of variation data for both independent and dependent variables. The formula for the regression equation is as follows. The matrix version of the expression is Y = BX, and the regression coefficient is

Do Not Copy, Lead City University, Nigeria

denoted by k b. The formula's coefficient matrix, B, is denoted by the letter b. B=W(PTW)-1QT. In the formula, the weight matrix is indicated by W.

In PLS, each dimension's computations are performed iteratively utilizing the other dimensions' data, with I u j t being adjusted for the second round of extraction based on the X, Y residual information at each iteration. Once the complete rate of the remaining matrix component approaches zero, the process stops since the desired precision has been achieved. During iteration, I u j t can exploit both the X and Y variance expressions.

Algorithm 2.12: Partial Least Square

1: procedure PLS

2: Discover eigenvectors of Sw that agree to a non-zero eigenvalue of a matrix (usually N − C), i.e. U = [u1, . . ., uN−C] by performing eigen analysis to (I − M)XT X(I − M) = VwΛwVw T and computing U = X(I − M)VwΛw −1 (execution whitening on Sw).

3: Plan the data as X˜ b = UT XM.

4: Complete PCA on X˜ b to find Q (i.e., calculate the eigen analysis of X˜ bX˜ T b = QΛbQT). 5: The total transform is W = UQ

Partial Least Square[100]

2.8.5 Locally Linear Embedding (LLE)

Some non-linear dimensionality reduction techniques translate high space to low space or vice versa; others merely offer a depiction of low-dimensional graphs or grids. In the context of machine learning, plotting strategies can be seen as an original extraction stage, surveyed by pattern recognition procedures[101]. There are two main categories for non-linear dimensionality reduction techniques: those that map from a high-dimensional space to an embedding in a lower-dimensional space, and those that just deliver a conception in the form of charts or plans at lesser dimensionalities. In the context of machine learning, plotting procedures can be

Do Not Copy, Lead City University, Nigeria

thought of as an elementary feature extraction stage that is then trailed by pattern recognition procedures[102].

There are a few different supervised and unsupervised methods for reducing the dimensions of data, respectively. As a rapid and unsupervised feature extraction methodology for microarray data analysis, a reviewed LLE method was suggested here. LLE performance is restrained in relationships of the classification precision element of a support vector machine classifier. After collecting the expression data, LLE is used to reduce the dimensionality from the thousands down to a more manageable range[103]. The effectiveness of the feature reduction procedure is then assessed using the SVM classifier and the Leave-one-out classifier measure. It is not necessary to precisely grid the embedding space for LLE, which corresponds to the intrinsic plentiful dimensionality constant d. To compute high-dimensional embeddings, LLE only needs to be applied once because the dominant dimensions do not change when new dimensions are added to the embedding space. LLE can be used on manifolds with any dimension, unlike other approaches like principal curves and sides or improver component models. By applying reinstating weights computed from the embedding vectors YWi to the data points XWi, an independent projection of d's intrinsic value can be made[104].

A popular non-linear dimension-reduction technique, local linear embedding (LLE) is appealing for its computational ease and quick execution. With LLE, the input points are linearly restructured from their neighbors, and then the neighborhood connections are preserved in a lower dimensional space [105]. There is no need for class labels in the LLE feature extraction procedure because it is completely unsupervised. This design for feature extraction is a straightforward linear algebra problem that requires no training or iteration, making the process really fast. Feature reduction may imply that LLE feature extraction is appropriate for binary-class feature lessening strategies[106].

Do Not Copy, Lead City University, Nigeria

Algorithm 2.13: Locally Linear Embedding

1. Select Kmax, λ, and E.

2. Discover Kmax adjacent neighbors to respective argument xi.

3. Discover the scarce masses utilized to inscribe individual point as a linear grouping of its nearest neighbors xj by resolving Equation (3) for respective point xi. If wij < E, set wij = 0.

4. Regulate lower-dimensional entrenching vectors by resolving the scarce eigenvector delinquent given by Equation (1) for Y.

Locally Linear Embedding[107]

2.8.6 Canonical Correlation Analysis (CCA)

Harold Hoteling introduced a statistical technique called canonical correlation analysis for interpreting cross-covariance matrices. CCA is a statistical method for determining the linear combinations of two sets of data and the variables that correlate most strongly with one another. Model equations connecting two sets of variables can be constructed with CCA, for example, a set of performance measurements and a set of expressive variables, or a set of outputs and inputs. CCA was able to make accurate class predictions utilizing only a small, carefully selected subset of samples from known classes. To give statistical data significance, Harold Hoteling recommended employing cross-covariance matrices and canonical correlation analysis. Considering the two types of variables and the interactions between them, we can perform a canonical analysis of the associations to find the linear clusters of variables that have the strongest connections to one another. In addition, CCA can be utilized to construct a classical equation with two sets of variables, such as a set of performance measurements and a set of expressive variables, or a set of inputs and outcomes[108].

Two methodologies that can be used to merge different modalities are CCA and its normalized counterpart, RCCA. Linear correlations between pixel values in photographs and

Do Not Copy, Lead City University, Nigeria

textual commitment between images have been discovered using CCA. Overfitting occurs in CCA, despite its simplicity when the modalities have many dimensions. The incorrect inverses of the conforming covariance matrices are due to CCA's lack of regularization. Through the use of canonical correlation analysis (CCA), we may ascertain the linear relationship between two-dimensional factors. It picks two bases, one for respective variable, that work well for establishing correlations, and does so simultaneously. The method essentially identifies the two points of support where the maximum diagonal correlations exist in the correlation matrix between the variables. CCA differs significantly from classical correlation analysis, which relies primarily on the principle by which the variables are distinct, in this crucial respect[109].

Algorithm 2.14: Canonical Correlation Analysis

Input: Information medium X 2 n ↠ p1, Y 2 n ↠ p2. A target measurement CCA. Amount of impertinent reiterations t1

Output: Xkcca 2 n ↠ kcca, Ykcca 2 n ↠ kcca contain upper kcca canonical variables of X and Y.

1. Produce a p1 ↠ kcca dimensional random matrix G with i.i.d standard regular entrances.

2.Let X0 = XG

3. for t = 1 to t1 do Yt = HYXt1 where HY = Y(Y>Y)1Y> Xt = HXYt where HX = X(X>X)1X>

end for

4. Xkcca = QR(Xt1), Ykcca = QR(Yt1) Function QR(Xt) excerpt an ortho-normal base of the post space of Xt with QR disintegration

Canonical Correlation Analysis[110]

2.9 Classification

Parameters are typically user-set in machine-learning classifiers. Classification-optimal parameter estimation, or "parameter tweaking." The risk of overfitting is of particular

Do Not Copy, Lead City University, Nigeria

importance when employing very effective classifiers, such as machine-learning techniques. This happens when the classifier's mapping of the training data is too precise, preventing it from generalizing well. Consequently, it is even more crucial for machine learning to adhere to the universal remote-sensing law that one assesses the classification accuracy utilizing innovative data not utilized in training the classifier[111].

Choosing a machine-learning classifier for a given job can be problematic since there is such a large variety of such approaches, and because the works seem conflicting, making it hard to simplify the qualified classification accuracy of specific machine-learning algorithms. Some examples of more accurate models are artificial neural networks (ANNs), support vector machines (SVMs), ensembles (E), k-nearest neighbors (KNNs), and decision trees. Classification comparison studies often yield contrasting findings, which may be attributable, in part, to methodological differences across the studies. Assuming there are no studies that can shed light on selecting a machine-learning classifier, however, would be a mistake[112].

Following the selection of an algorithm, the ensuing classification performance may be affected by a wide variety of issues, like the nature of data training, the values selected for the procedure's parameters, and the feature space of the predictor variables. We will now delve into these core concerns, starting with problems with training data [113].

Do Not Copy, Lead City University, Nigeria

2.9.1 Support Vector Machine (SVM) Classification Algorithm

Support Vector Machine first proposes the idea of a vector machine. When it comes to classification and regression, SVM is a supervised learning procedure that is extensively employed. Finding the hyperplane that has the largest margin between the classes so that the classes can be divided linearly is the purpose of SVM. Vector machine learning is used to

help find datasets with insufficient training data for standard statistical methods to guarantee an optimal answer[114].

In the context of challenges with categorization. The SVM algorithm is employed; it is also applicable to regression analysis. Points representing individual data points are then plotted in an N-dimensional feature space, with each feature's value serving as a coordinate. Then, the hyper-plane that most separates the two groups is identified and used to make the categorization. When determining support vectors, the coordinates of an observation nearest to the boundary are considered. For SVM training, data is partitioned into support vectors, which in turn define the decision function[115].

The training data are predictable to a higher dimensional feature space as the linear separation in the kernel functionality gets easier in the input space. SVM uses a variety of kernel functions, including the radial basis function (RBF) and the polynomial kernel, to discover a hyperplane that effectively classifies data with small training sets. To accurately evaluate hyperplanes and lessen classification errors, the SVM algorithm needs a suitable kernel function. In the SVM method, the kernel shape is the most critical component. The performance of the SVM is largely dependent on the size of the kernel, while smooth surface similarity is largely dependent on the kernel density. And unlike other machine learning techniques, picking the kernel function is a delicate and laborious process. It will be computationally expensive to increase the dimensionality of the problem to support SVM segregation because the power of the SVM approach is its ability to optimize itself by modifying its kernel feature. Many researchers have investigated SVM and used it in a wide range of applications. Many real-world challenges are out of their league because of the exponential growth in the volume of training vectors that necessitates computational and storage resources[116,117.]

Do Not Copy, Lead City University, Nigeria

An example of a supervised machine learning method, the Support Vector Machine (SVM) uses a distance measure, such as a margin, to classify input as positive or negative [118]. The bias in SVM is denoted by b, while x and w are vectors. The hyperplane is defined as wT x = 0. The optimum hyperplane for SVM is the one with the largest margin separation. An increase in the gap between the hyperplanes can be achieved by decreasing the || w || and is formulated. In a non-linear setting, the data may not be accurately categorized. To account for the possibility of error, we employed a slack variable I, where I = 1, 2, 3, ..., n, and it is linked to a constant C. The error rate increases with increasing values of C, and decreases with decreasing values of C. D = a1, a2, a3, ..., aN Y = (7) xi.w + b +1 for all yi = +1 xi.w + b 1 for all yi = 1 minw,b = ||w||.

The width of the hyperplane's margin is established by the SVM's support vectors. The prediction accuracy of the SVM algorithm can be enhanced in several ways. It is possible to alter the kernel function. It is possible to transform data that is not linearly separable into data that is linearly separable using a mathematical function called the kernel trick, which involves projecting the data into a higher-dimensional space. The notation for a kernel function is K(m,n). We had two dimensions, m, and n, as inputs, and a function, f, that mapped those dimensions into some other spatial dimension. Based on the mathematical operations performed during the kernel technique, it is separated into the following classes[119]; We'll be looking at the Linear Kernel, a radially biased Kernel, and a Polynomial Kernel.

The following definition explains how to work with linear kernels, which are conceptually comparable to the usual dot product of two vectors. To differentiate itself from the linear kernel function, the polynomial kernel function uses a degree of polynomials. The kernel is also known as the Radial Bias Function and may be written as ||x1 x2||, where ||x1 x2|| is a

Do Not Copy, Lead City University, Nigeria

Euclidean distance characterizing the decision region. In SVM, the Margin (M) and the Misclassification Rate (MCR) are directly related to one another. This means that the rate of misclassification rises as the margin of the hyperplane gets larger. The SVM can better calculate the hyperplane margin with the help of the cost parameter "C." There should be no misclassifications, hence we should use a smaller margin to classify all vectors into the correct space.

Algorithm 2.15: Support Vector Machine[120]

***Input*:** D =[X,Y]; X(array of input with m features), Y (array of class labels)

Y= array(C)// Class label

Output: Find the performance of the system

function train_svm(X,Y, number_of_runs)

initialise: learning_rate **in**number_of runs

error = 0;

for i **in** X

if (Y[i]*(X[i]*w) < 1 **then**

update: w = w + learning_rate *((X[i]*Y[i]) * (-2 *(1/number_of_runs)*w)

else

Update: w = w + learning_rate * (-2 *(1/number_of_runs)*w)

end if

end

end

Support Vector Machine[120]

2.9.2 Ensemble Classification Learning

The goal of ensemble classification, a generic meta-approach to machine learning, is to improve predicted performance by integrating methods. Combining many machine learning

Do Not Copy, Lead City University, Nigeria

algorithms into a single model will maximize its accuracy because it will capitalize on the advantages of each unique method. The use of an ensemble of learners has proven superior to that of a single classifier when it comes to picture prediction and classification[121].

Specifically, bagging, stacking, and boosting are the three subcategories of ensemble learning. As opposed to the stacking technique, which involves fitting multiple separate models onto the same data and then using a third model to learn the combined predictions, the bagging method focuses on making many decisions on different samples of the same dataset and determining the average forecast[122].

By combining the strengths of numerous models, the ensemble system of machine learning applications can generate results that surpass those of a single model alone. By combining multiple models into one, an ensemble can be used to increase prediction accuracy, decrease prediction error (through bagging), or counteract bias (by boosting) (stacking). Technical classifications of ensemble methods including bagging, boosting, stacking, and random forest exist, as do sequential, parallel, homogeneous, and heterogeneous ensemble machine-learning techniques[123,124].

By contrast, the continuous approach sequentially initiates the basic learners (where the data reliance persists). Moreover, all subsequent data in the foundational level depends on the data in the foundational level before it, and the incorrectly labelled must be weight-adjusted to produce a performance analysis of the system. One technique that fits this description is called "boosting." The parallel approach guarantees that the foundational learner is started simultaneously, there is no data dependency, and all data are generated separately[125].

Do Not Copy, Lead City University, Nigeria

The homogeneous ensemble approach is useful for a wide variety of datasets because it employs a mixture of the same kinds of classifiers. Each classifier uses a unique dataset, and after compiling its results, a robust model emerges. No matter what sort of data you're looking to train on, you can use the same feature selection approach. The biggest issue with this kind of model is that it is highly costly to compute. The most well-known implementation of this paradigm is the bagging and boosting strategy. On the other hand, tiny datasets can benefit from the heterogeneous ensemble approach, which mixes many classifiers that were all trained on the same data but with distinct feature selection processes. A typical classifier of this sort is stacking[126].

2.9.2.1 Ensemble Approach Technical Classification

I. Bagging

The bootstrapping and aggregation of two models into a single ensemble model is the essence of ensemble learning with the bagging approach. Every element in bagging has an equal chance of appearing in the new dataset, as random sampling is used to reduce model variance by generating extra data during the model's training stage. This technique helps lessen uncertainty and zero in on a more precise prediction. When it comes to increasing the precision of an algorithm, nothing beats the tried-and-true technique of bagging. The method of estimate utilizing several classifiers is conceptually similar to that of bagging. The results of each course are evaluated and chosen.

II. Boosting

The boosting technique of ensemble classification makes use of a constant approach of classifying depending on the features that the subsequent model will use. When applied to a poor learner model, boosting techniques can improve its performance by increasing its weighted average. The considerably more robust trained model largely relies on the many, much more limited trained models. The least correlated true classification is found in the

Do Not Copy, Lead City University, Nigeria

weak learner, and incremental improvements lead to somewhat higher correlation in the next weak learner and so on until the weak learners are aggregated into the strong learner, which is significantly correlated with accurate classification[127].

III. Stacking

The stacking strategy utilizes regression methods or several model combinations, as well as a separate classifier. Training the lower-level model using the complete dataset is required, and then the results are fed back into the training of the collective model. This is distinct from boosting because the lower level is in line with parallel training. The next model is built as a stack based on the forecast from the previous level. The stack operation is structured so that the most trained and most accurate prediction is at the top of the stack, while the least trained and least accurate prediction is at the bottom of the stack. One prediction is made after another until the one with the fewest mistakes stands out as the winner[128].

2.9.2.2 Ensemble Framework

When it comes to machine learning, ensemble methods and/or models combine many learning algorithms to produce superior predicted performance compared to that of using a single learning model alone. Since the individual models in an ensemble are more distinct, the resulting experimental results are more reliable.

2.9.3 Random Forest

As its name suggests, the Random Forest (rf) classifier is an ensemble of numerous individual decision trees. In this technique, each tree in the random forest predicts a class, and the class with the most votes determine the model's forecast[129]. The concept and driving force behind rf are "the wisdom of crowds." The reasoning behind this is that a group of models (trees) with low correlation will perform better than a single model. To rephrase, the rf classifier, like the DT classifier, does not need feature scaling and is instead built from a collection of "weak

Do Not Copy, Lead City University, Nigeria

learners" (trees). The rf classifier outperforms the DT classifier in terms of robustness to both training sample selection and noise in the training dataset[130]. However, the rf classifier presents additional challenges in terms of comprehension. The data set is sampled in the random forest models by tree operations. The tree is aggregated after being fitted on bootstrap samples to reduce error. It employs the features to randomly select the dataset to construct the tree, which helps to lower the outputs' correlation.

Because each tree in the model has its unique structure, the random forest model is ideal for identifying missing data because it selects a random subset of the sample to lower the likelihood of sharing prediction values. When less accurate predictions from many trees are averaged, the resulting variation leads to improved results[131]. This algorithm, which is a forest of decision trees loaded with arbitrary and unique algorithms, provides a set of procedures that function together. Based on the results of numerous decision trees, the majority vote decides. It's one of the most powerful algorithms out there. It's effective at classification and regression, but overfitting is a major problem[132].

2.9.4 Decision trees Algorithm

Defining the connections and weights between attributes is the focus of decision trees, a potent classification system. The benefits of these algorithms include a lack of the need for complex data preparation and the display of rules that are easy to grasp and interpret. Both numerical and qualitative data show improved results when using these techniques[133].

Data mining routinely makes use of systems that build classifiers. Data mining classification techniques may scale to enormous datasets. It can generalize class labels, categorize information learned from previous datasets, and classify unlabeled data[134].

Do Not Copy, Lead City University, Nigeria

Decision trees are a robust technique that has found widespread application in various areas, such as machine learning, image processing, and pattern recognition. DT is a sequential model that effectively and consistently binds together a battery of fundamental tests, each of which compares a numerical property to a threshold value. The numerical weights in a neural network of node connections are more difficult to construct than the conceptual rules. The primary application of DT is categorisation[135]. In addition, DT is widely employed as a classifier in the field of data mining. Branches and nodes make up each tree. Each node stands in for a different attribute of the classifiable entity, and its possible values are defined by the subsets. Given their ease of use and ability to accurately analyze a wide variety of data, decision trees have been widely used in a variety of settings[136].

2.9.4.1 Different Types of Decision Tree Algorithms

Different DT algorithms exist, for instance, Iterative Dichotomies 3 (ID3), Its Successor (C4.5/5.0), Chi-squared Automatic Interaction Detector (CHAID), Classification and Regression Tree (CART), Multivariate Adaptive Regression Splines (MARS), Comprehensive, Conditional Inference Trees (CTREE), Unbiased Interaction Detection and Estimation (GUIDE), and Classification Rule with Unbiased Interaction Selection (C (QUEST)[137].

2.9.4.2 Information Gain and Entropy

As a measure of unpredictability or impurity, entropy is useful in determining the quality of a dataset. The value of entropy is continuous between 0 and 1. Being closer to 0 is preferable, as its value improves when it is equal to 0 and worsens when it is equal to 0. If the target set G has varying attribute values, then the entropy of the classification of set S concerning those states is c. Information gain (or mutual information) is a statistic that can be used for

Do Not Copy, Lead City University, Nigeria

classifying groups of data. This provides a natural measure of how much is known about a random variable's value. The higher its value, the better; it is the opposite of entropy [138].

2.9.4.3 Decision Tree Advantage

As a supervised learning algorithm, the DT method's primary purpose is to construct a training model for predicting the class or value of target variables based on learning decision rules derived from training data[139].

2.9.5 K-Nearest Neighbors classifier (kNN)

Both statistical prediction techniques and pattern recognition make use of the supervised k-Nearest Neighbor algorithm. This algorithm's purpose is to assign labels to objects based on the labels assigned to their nearest neighbors. For this model space, the number of neighbors, denoted by the positive integer k, must be less than or equal to the size of the data set. When k is equal to 1, we use the training sample's class that is most similar to the unknown sample in the model space. The kNN algorithm's performance is affected by the size of k, with larger values of k reducing the impact of the noise variable in the classification and blurring the distinction between the classes[140].

There are several applications for the K-Nearest Neighbor (KNN) optimization method, including production optimization, pattern recognition, image processing, and many others. A substantial amount of training data is necessary for algorithms to benefit from the KNN approach. The KNN method seeks to classify new objects based on qualities and training examples, and it yields reliable optimization outcomes[141]. When designing a classification method, KNN optimization is preferable since it does not presume anything about the form of the "smooth" function f that connects y (the response variable) to x. (predictor variables). If there is no need to estimate any parameters for the function f, then we say that it is non-

Do Not Copy, Lead City University, Nigeria

parametric. With KNN, we iteratively find k observations from the training data set that are most similar to a given point p = (p1, p2, ..., pn) (the k nearest neighbors). A measure of how far apart (x1, x2, ..., xn) and (p1, p2, ..., pn). For each n-dimensional object (i.e., an object with n characteristics) in the training data, the Euclidean distances between the requested object and all the training data objects are computed, and the queried object is given the class label that most of the k closest training data objects have[142].

The K-NN technique relies largely on the furthermost theorem theoretically. When selecting a choice, think about how close your options are. Thus, the illustrative disproportion problem can be settled by employing this technique. K-NN considers only a small set of nearest neighbors and does not take any kind of decision boundary into account. Thus, it is remarkable to state that K-NN is suitable for classifying the case of boundary intercrossing and in that case overlapping examples. The formula for the Euclidean distance is as follows. Assuming we have two vectors, xi and xj, where xi = (xi1, xi2, xi3, xi4, xi5, ..., xin), and xj = (xj1, xj2, xj3, xj4, xj5, ..., xjn)[143].

The K-nearest neighbor (KNN) algorithm is a popular choice since it has a high positive predictive value and a low false positive rate. The general notion is as follows: fresh data entered for classification may be of an unknown category but can be assigned a category by comparing it to existing samples. At the outset, it's necessary to extract features from the data to be classified and compare them to those of each known category in the test set. The K-nearest neighbor data was taken from the evaluation set, and the proportion of data in each category was tallied. At last, the classified information is placed here[144].

With N training samples A = x1, x2, ..., xn, the KNN classification method was applied, and the data was partitioned into S classes W1, W2, ..., and WS. The categories are represented by a total of Ni I = 1, 2, ..., S) samples used for training. Determine K such that K12 ... KS12 are

Do Not Copy, Lead City University, Nigeria

the nearest samples. Classifying sample X into one of several possible buckets is accomplished by using the discriminant function, defined as = ki, where I = 1, 2, ..., S, and the maximum likelihood estimate, Max, are given (Ki). Details of how the K-Nearest Neighbors (KNN) algorithm is put into practice are outlined below[145]:

First, we split the data into a training set and a testing set. A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, U, V, W, X, Y, Z are the training, evaluation, and evaluation sample sets, respectively.

Second, set X's nearest neighbor to the initial k value.

Third, calculate the gap between the test sample points and the rest of the training sample points.

The fourth step is to put the resulting distance in ascending order before deciding on a value for k.

In the fifth stage, you will select the k nearest known samples.

Sixth, we tallied how many instances there were of each of the categories with the highest probability among the k samples we had.

Seventh, Based on the statistical results from Step 6, classify the points in the test sample.

The KNN classification method has numerous benefits, including being easy to learn and producing accurate classification results, but it also has many downsides. That it takes a lot of extra time and space to run the algorithm is one of them. KNN is a lazy algorithm, hence it prefers to just be given raw data to classify. This necessitates considering every possible set of sample data, which might take quite some time[146].

2.9.6 Artificial Neural Network

Artificial neural networks (ANN), often known as deep learning, are another supervised learning approach (depending on the number of hidden layers). Studies of the human nervous

Do Not Copy, Lead City University, Nigeria

system served as inspiration for ANN. Artificial neural networks (ANN) are an efficient method of mining large datasets. It takes its cue from biological systems with the ability to recognise patterns and forecast future outcomes. It is in the realm of solving actual world problems that neural networks have made the most progress in recent years[147]. This ranges from concerns like detecting fraud to how businesses should react to their customers in the real world. Due to its unique ability to learn, DL may extract crucial relationships or patterns from fragmentary, complex, and possibly imprecise data that cannot be recognized by other computational methods. Each layer in a DL architecture—the input layer, the hidden layer, and the output layer—serves a different purpose.

Applications demonstrate that DL's superior predictive accuracy compared to other approaches or human experts is beneficial for data challenges. In addition, DL is capable of self-organization, reasoning, parallel processing, and storing large amounts of data. In addition, it can quickly fit nonlinear data, a feature that helps it address a wide range of issues that would otherwise be intractable. Despite its many benefits, DL is not entirely open to outside scrutiny. It is generally agreed that the DL algorithm is opaque. This implies that it makes absurd claims. This is because of the intricate design of the building. Even though a single hidden-layer ANN may seem straightforward, it is impossible to explain why a given data point is assigned to one class and another. The inability to classify data using symbolic rules is another drawback of DL. It is therefore not appropriate for human expert verification and interpretation in an explicit sense[148].

2.9.7 Naive Bayes Classifier

Bayesian classifiers are widely used because they are effective, efficient, easy to train, applicable to real-world issues, and accurate. Classes in the Naive Bayes algorithm take into account the hypotheses to which they are most likely to belong. This classifier uses frequency

Do Not Copy, Lead City University, Nigeria

calculations with a tree of hypotheses to sort the input data. Naive Bayes is a classification technique that is based on the total probability and Bayes theorem. In theory, multiple Naive Bayes algorithms have been devised[149,150.]

Students of the regulations: It's important to note that this technique takes into account not one, but two classifiers. OneR is a collection of rules for evaluating a single property in the form of a single-level decision tree. It's a quick and cheap way to get accurate rules for defining data structures. As with previous classifiers, this one is based on the comparison. As a corollary, it shows how accurate predictions may be made using certain traits. The Jrip classifier employs the RIPPER (Repeated Incremental Pruning to Produce Error Reduction) method[151].

2.10 Related Works

Deep Networks were proposed for scene text recognition in the wild with motion deblurring[152]. Two primary challenges that make information extraction from video shot by a moving vehicle challenging have been addressed in a recent solution to the problem of video text detection and identification. Moving vehicles' videos have a lot of blurring because of the motion, which is one of the main problems prohibiting precise text detection. The orientation of the identified text, which may or may not be on the same plane, is the second main challenge. They suggest a new whole pipeline that makes use of deep neural networks. For text detection, they use a fully convolutional network; for motion blur removal, they use a Generative Adversarial Network; for curved and perspective text, they use a rectification network that uses Thin Spline Transformations; and for recognition, they use a recognition network that also uses Thin Spline Transformations and a Spatial Transform network. Rather than deblurring the entire image, the study only deblurs the area surrounding text boxes and keeps track of them from frame to frame to prevent having to recognize the text again. Better

Do Not Copy, Lead City University, Nigeria

categorization scores are evidence that this greatly boosts the system's performance compared to the state of the art.

Scene text detection and recognition in the deep learning era with the rise and development of deep learning, computer vision has been dramatically transformed and reshaped[153]. Important in the field of computer vision, scene text identification and recognition have been profoundly impacted by the advent of deep learning. Over the past few years, the community has made great strides in terms of outlook, strategy, and results. Through the presentation of novel concepts and insights, the elaboration on recent approaches and benchmarks, and the anticipation of future trends, this review attempts to synthesize and assess the major changes and notable advancements in scene text detection and recognition in the deep learning age. The study emphasizes both the enormous progress that has been made thanks to deep learning and the enormous difficulties that still need to be overcome. They hope that this review article would serve as a bible for future scholars in this area. Their GitHub repository also includes a collection and compilation of relevant materials: https://github.com/Jyouhou/SceneTextPapers.

Scene text detection via extremal region-based double threshold convolutional network classification was demonstrated in natural images using a robust text detection approach based on a region proposal mechanism[154]. A robust low-level detector called saliency enhanced-MSER is proposed, which guarantees a high recall rate by combining saliency detection techniques. Given a genuine image, the saliency-improved MSER algorithm selects potential characters from three channels in an illumination-invariant color space based on human perception. A discriminative convolutional neural network (CNN) is jointly trained with multi-level information, including pixel-level and character-level information, to serve as a classifier for potential characters. Double threshold filtering and confidence ratings derived from a convolutional neural network (CNN) are used to categorize each image patch as either

Do Not Copy, Lead City University, Nigeria

strong text, weak text, or non-text, as opposed to the more common approach of performing this task all at once. To better prune non-text regions, this research created a recursive neighborhood search algorithm to follow trustworthy texts from a weak text set. The last step is to employ heuristic features to organize the characters into lines of text based on their proximity to one another, size, color, and stroke width. Experiments reveal that their solution outperforms various state-of-the-art algorithms on the public datasets ICDAR 2011 and ICDAR 2013.

It was proposed to use two-stage Deep Belief Networks for image blur classification and parameter identification[155]. Blind picture deblurring relies heavily on the ability to classify blur kernels and estimate their parameters. When the Point Spread Function (PSF) of the blur is unknown, the current state-of-the-art methods for blind deconvolution fail because they rely on manually constructed blur features tuned for a certain type of blur. This research proposes a two-step technique utilizing Deep Belief Networks (TDBN) for determining the blur type and associated parameters. This is the first time a Deep Belief Network (DBN) has been employed to address a blur analysis issue, as far as the authors are aware. Instead of estimating blur on the assumption of a single blur type, as is done with current methods, they attempt to identify the blur type from a mixed input of distinct blurs with varied characteristics. To do this, a semi-supervised DBN is trained to classify features obtained from projecting input samples into a discriminative feature space. In addition, the proposed edge detection on the logarithm spectrum supports DBN in accurately determining blur settings. The proposed methods are superior to the state-of-the-art, as demonstrated by experiments on the Pascal VOC 2007 dataset and the Berkeley segmentation dataset.

A Blur Classification Approach Based on Pattern Classification was proposed [156]. Finding the cause of the blur is a crucial first step in fixing a blurred image. It is commonly assumed that

Do Not Copy, Lead City University, Nigeria

the blur type is known before attempting to restore such photos when doing so blindly. However, this approach is not viable in actual use cases. Therefore, it is highly desirable to determine the type of blur before employing the blind restoration method to recover a blurred image. In this study, we provide a system for distinguishing between motion blur, defocus blur, and combined blur. The blur pattern features are energy features based on the curvelet transform, and the classification is performed by a neural network. The accuracy of the proposed method is shown through simulation.

With ensemble convolution neural networks, you can identify blurry images. In image processing, blur image classification is an important step toward image[157]. In this piece, we utilise an ensemble convolutional neural network (CNN) to identify and label four distinct kinds of blurred images (defocus blur, Gaussian blur, hazy blur, and motion blur). For this purpose, we propose a two-stage pipeline consisting of deep compression and the ensemble technique to boost model discriminability without significantly increasing the computational load. In particular, before applying their strategy, popular networks like Alexnet and GoogleNet are pruned with an optimal compression ratio. The two trimmed networks are Simplified-Fast-Alexnet (SFA) and Simplified-Fast-GoogleNet (SFGN) (SFGN). The study then used an ensemble approach to merge the SFA and SFGN, renaming it SFA+SFGN by giving each component a predetermined weight based on a voting method. In addition, a standard set of blur images for use in training and testing classification algorithms was also provided.

This research resulted in the release of a new public blur image dataset (accessible at http://doip.buaa.edu.cn/info/1092/1073.htm) that includes over 200,000 artificially blurred images and over 80,000 patch-level, naturally blurred photographs created using an enhanced super-pixel segmentation technique. Compared to the baseline Alexnet and GoogleNet, as

Do Not Copy, Lead City University, Nigeria

well as other state-of-the-art methods, the suggested method performs better in numerical experiments.

A Class-Specified Topic Model was proposed for Bayesian Text Classification and Summarisation[158]. Class-specific text summarizing and text categorization jobs are handled by the class-specified topic model (CSTM). The model presumes that there exists, for each class, a collection of latent topics that are distinct from the set of latent topics shared by all classes. Each document has a random blend of content from both classes. For any given lexicon, the frequency with which each topic is discussed in each class or within classes varies. In the semi-supervised setting, they develop a Bayesian inference of CSTM, while the supervised setting remains an outlier. Using the 20 Newsgroups dataset as a text classification benchmark, they show that CSTM is superior to a two-stage strategy based on latent Dirichlet allocation (LDA), as well as many current supervised extensions of LDA and an L1 penalized logistic regression. CSTM's improved performance is also demonstrated through Monte Carlo simulations and analysis of the Reuters dataset.

Deep Learning-based Blur Image Classification was proposed in this next article[159]. Defocus blur, Gaussian blur, hazy blur, and motion blur can all be identified with the use of a reliable classification system based on a Convolution Neural Network (CNN). A supervised learning model of Simplified-Fast-Alexnet (SFA), an abridged and modified version of Alexnet, is built to map the input images into a higher dimensional feature space where the blurs may be classified reliably. By eliminating Alexnet's first two Full Connected layers (FCs) and halving the output number of each convolution layer, the SFA simplifies Alexnet and gets around the fatal problem of parameter redundancy. In addition, the dropout approach is replaced by adding batch normalization layers to the selected classifier; these layers speed up deep network convergence during training by decreasing internal covariate shifts and relieving the

Do Not Copy, Lead City University, Nigeria

overfitting issue. Experimental results on the popular Berkeley dataset and the Pascal VOC 2007 dataset show that the suggested method beats the original Alexnet and the state-of-the-art.

The detection and classification of blur images were performed using a Multi-Class Support Vector Machine[160]. In modern technologies, blind picture restoration is essential. Here, a Multi-class Support Vector Machine (MSVM) framework is applied to the problem of blur categorization in digital images. The focus of this effort is on using MSVM to categorize blurry photos. The goal of the MSVM classifier is to identify crisp, defocused, and motion-blurred images. We conduct several tests on data from the Beihang University Blur Image Database (BHBID). Sobel, Laplacian, and Roberts cross-edge detections are used to determine the average, standard deviation, and maximum of the feature matrix describing the edges detected in each image. Each component of the MSVM classifier is trained using a different set of features selected using a sampling method. It compares the results of optimizing SVM parameters with several kernels, including Linear, Polynomial, Radial Basis Function (RBF), and Gaussian. They concluded that their proposed approach successfully located the specified scenarios 95.7% of the time.

The detection and classification of blurred images were performed here too. While digital cameras are becoming more popular, digital photos are being produced in large quantities; however, not every photo is of high quality[161]. Multiple circumstances can lead to the blurring of an image, which can significantly lower the image's quality. A technique was developed to identify blurred photos and sort them into distinct groups. With the use of support vector machines, the blur detector can determine how blurred an image is. Images with blurring can be further categorized as either locally blurred or globally blurred. Point spread functions of globally blurred photos are estimated and sorted into categories such as camera shake and out-

Do Not Copy, Lead City University, Nigeria

of-focus. They employed a segmentation technique to identify the blurred areas in locally blurred photos and found that estimating the point spread function in that area helped distinguish between still and moving pictures. The blur detection and classification procedures are fully automated and can help users eliminate blurry photos from digital photo albums.

A Line-segment Feature Analysis Algorithm for Handwritten Text Recognition with Input Dimensionality Reduction was presented[162]. Handwriting recognition has seen a meteoric rise in popularity in recent years and is already being used in places as diverse as automated mail sorting, vehicle license plate readers, and digital notepads. Moreover, in the area of image recognition, convolutional neural network-based algorithms have been employed for handwriting detection with great success. However, as the variety of recognising use cases increases, so does the number of moving parts in learning and reasoning processes. When this occurs, a principal component analysis (PCA) method is used for dimensionality reduction. However, PCA is likely to make the accuracy loss from data compression much worse. Therefore, the study offered a line-segment feature analysis (LFA) strategy for lowering input dimensionality in handwritten text recognition.

This proposed approach utilizes 3X3 and 5X5 filters to separate the line segments that make up the input data image and assign a separate value to each one. The singular values are used to determine the total number of lines, and the sum of these values yields a 1-dimensional vector of size 512. For machine learning, this vector is essential. Effectiveness was measured by applying it to data from the Extending Modified National Institute of Standards and Technology (EMNIST) database. PCA achieved 96.6 and 93.86 percent accuracy with k-nearest neighbors (KNN) and support vector machine (SVM), whereas LFA achieved 97.5 and 98.9 percent accuracy with KNN and SVM, respectively.

Do Not Copy, Lead City University, Nigeria

The problem of detecting blurred image regions was proposed in Features Extraction for Detection of Blurred Image Regions[163]. Twenty of the proposed features are based on the discrete wavelet transform, while the other is a ratio calculated from the co-occurrence matrix of grayscale values. Background blur, motion blur, and out-of-focus blur are all types of blurs that can be detected using the features mentioned in this section. The proposed characteristics are very easy to compute and parallelize. A backpropagation-based multilayer perceptron is trained on the above-mentioned properties. Nonoverlapping fixed-size windows, nonoverlapping recursively divided windows, and the introduction of a morphological closure operator are described and evaluated.

For image deconvolution, a Deep Convolutional Neural Network was proposed[164]. As such, deconvolution operators find widespread applications in a wide variety of fundamental image processing applications. Real-world blur degradation rarely coincides with a perfect linear convolution model for reasons including camera noise, saturation, and image compression. Instead of trying to accurately depict outliers, which is challenging for a generative model, they developed a deep convolutional neural network to capture the properties of degradation. Results from the study indicate that it is not a good idea to rely on preexisting deep neural networks for reliable output. The approach taken in the research is to combine traditional optimization-based approaches with a neural network architecture that relies on a distinct, decoupled structure as a solid basis for artifact-resistant robust deconvolution. Each of the two modules that make up their network has been carefully monitored and correctly initialized. They excel at non-blind photo deconvolution, where they outperform earlier generative-model-based algorithms.

Do Not Copy, Lead City University, Nigeria

It discussed a new dataset size reduction approach for PCA-based classification in an OCR application[165]. One major challenge for pattern recognition algorithms is dealing with the massive size of training datasets, which often include duplicate and similar training samples. Several methods for decreasing the overall volume and dimensionality of datasets have been developed to deal with this matter. Existing methods for reducing the size of datasets often get rid of data from the edges and centers of classes as well as support vector samples that fall between classes. Conversely, the support vector is essential for evaluating the efficacy of a system, and the samples close to a class center contain valuable information on the class characteristics. In this paper, we explore the potential of the Modified Frequency Diagram method for reducing the overall size of a dataset. In this innovative method, a training dataset is first restructured, and then sieved, to remove irrelevant data. Principal Component Analysis is used to combine the filtered training dataset with automatic feature extraction/selection for an OCR application. Hoda is one of the largest handwritten Farsi/Arabic number standard OCR datasets, and experimental results produced using the proposed approach demonstrate a recognition rate of roughly 97 percent. The recognition speed improved by a factor of 2.28 while the accuracy dropped by only 0.7% when using a sieved version of the dataset that is only half the size of the original training dataset.

There was a proposal for a review of dimensionality reduction techniques for efficient computation[166]. To improve learning feature accuracy and decrease training time, dimensionality reduction (DR) is a pre-processing approach that eliminates redundant features, noisy data, and irrelevant data. Dimensionality reduction strategies have been developed and implemented using feature selection and extraction approaches. To speed up the learning process, principal component analysis (PCA) is used as one of the Dimensions reduction methods. In this research, we analyzed the efficiency and precision of several popular feature extraction methods, including principal component analysis (PCA) and empirical mode

Do Not Copy, Lead City University, Nigeria

decomposition (EMD), and feature selection strategies, including correlation, linear discriminant analysis (LDA), and forward selection. These methods are frequently used in Deep Neural Networks to enhance the accuracy with which medical images are diagnosed and categorized. The authors discussed the role of dimension reduction in deep learning.

A model was given for designing a hybrid dimension reduction for increasing the performance of Amharic news document categorisation[167]. The availability of online resources written in Amharic has grown tremendously during the past few years. Therefore, automated document classification is essential. Their unique dimension reduction technique for improving classification accuracy is achieved through the combination of feature selection and feature extraction. Key features are selected with the use of the Information Gain (IG), Chi-square test (Chi), and Document Frequency (DF) methods, and then refined with the help of Principal Component Analysis (PCA), all within the context of the new dimension reduction method. This research evaluates the proposed dimension reduction method using a dataset comprised of 9 different types of news articles. The proposed strategy for reducing dimensions outperforms competing approaches. When compared to IG, CHI, and DF, the new dimension reduction's classification accuracy is 92.60 percent, 13.48 percent, 16.51 percent, and 10.19 percent higher, respectively. More effort is required since minimizing processing time necessitates shrinking feature size, which in turn reduces classification accuracy.

The proposal included a detailed analysis of optical character recognition technology[68]. There is a significant need for transferring data from printed or handwritten documents or photos onto a computer storage disk for subsequent use by computers in a wide range of fields. It may be simple to transfer data from various print sources to a computer system by scanning the documents and saving them as image files. However, it would be very difficult to interpret the text from these image files or query other information included therein to reuse this data.

Do Not Copy, Lead City University, Nigeria

Consequently, there needs to be a system for extracting and storing data, especially text, from digital images. The goal of optical character recognition research is to build an algorithm that will enable computers to automatically identify and interpret the text in images. The purpose of optical character recognition (OCR) is to digitize and alter non-typable text such as handwritten text, printed text, or scanned text images. Therefore, OCR makes it possible for a machine to automatically recognize text within such documents. Some fundamental issues must be identified and resolved before automation can be implemented successfully. Some relatively recent problems have involved the quality of printed text and images. Because of these problems, computers may have trouble correctly recognizing characters. In this study, four distinct approaches to OCR are examined. First, they lay out in detail all the problems that could occur during the OCR process. After that, they discussed the many stages of an OCR system, such as pre-processing, segmentation, normalization, feature extraction, classification, and post-processing. Then, we discussed the most recent developments in OCR and its widespread applications. The research paper provided a concise background of OCR. Therefore, this discussion offers a reasonably comprehensive summary of the current situation in the region.

A Comprehensive Review of Deep Convolutional Neural Networks for Image Classification was conducted[19]. Convolutional neural networks (CNNs) have been applied to visual difficulties since the late 1980s. Despite some limited uses, neural networks didn't receive widespread adoption until the mid-2000s, when improvements in processing power, the availability of massive amounts of labelled data, and improved algorithms catapulted them to the forefront of a neural network renaissance. This overview, which focuses on CNNs' usage in picture classification tasks, traces their development from their earliest ancestors to the most advanced deep learning systems available today. They looked back at their early successes, their participation in the deep learning renaissance, the symbolic works that have

Do Not Copy, Lead City University, Nigeria

contributed to their recent prominence, and numerous enhancement initiatives, analyzing the contributions and challenges from over 300 articles. The trends and problems now confronting them are also discussed.

A proposal was made for Deep Learning: A Comprehensive Overview of Techniques, Taxonomy, Applications, and Research Directions[70]. The subset of ML and AI, deep learning (DL) is now generally acknowledged as an essential part of the Fourth Industrial Revolution (4IR or Industry 4.0). For its data-learning capabilities and widespread use in fields as diverse as healthcare, visual recognition, text analytics, and cybersecurity, DL technology has emerged as a hot topic in the computing world. The challenge in creating a reliable DL model comes from the fact that real-world circumstances and data are inherently dynamic and subject to change. Furthermore, DL techniques become black-box devices that impede regular advancement due to a lack of fundamental understanding. This article presents a well-structured introduction to deep learning methods, including a taxonomy that accounts for the variety of real-world tasks that can be either supervised or unsupervised. They have a taxonomy that includes deep networks for supervised/discriminative learning, unsupervised/generative learning, hybrid learning, and relevant others. Many examples of how deep learning algorithms can be used in practice were discussed. Finally, they covered 10 potential components of next-generation DL modelling and future research directions. The overarching objective of this study is to serve as a resource for both researchers and practitioners in the field of DL modelling by providing such a comprehensive overview.

A Survey of Object Detection in 20 Years[166]. One of the oldest and most challenging problems in computer vision, object detection has recently received a lot of research and development attention. Its development during the past twenty years represents a high point in the field of computer vision. Looking back 20 years, they can appreciate the wisdom of the cold weapon age if they view modern object detection as an example of technological

Do Not Copy, Lead City University, Nigeria

aesthetics fueled by deep learning. In this study, almost 400 papers are analyzed to determine how object detection has changed over the past quarter of a century (from the 1990s to 2019). This paper covers a wide range of topics, from historical landmark detectors and detection datasets through metrics, fundamental components of the detection system, acceleration methods, and cutting-edge detection techniques. The limitations and current technological advances of several essential detection applications are also investigated in this paper. These applications include pedestrian identification, face detection, text detection, and so on.

Image Data Augmentation for Deep Learning is the subject of a survey[167]. When it comes to Computer Vision tasks, deep convolutional neural networks have proven to be superior. To prevent overfitting, however, these networks require massive amounts of information. If a network attempts to correctly mimic the training data, it will overfit if it learns a function with a very large variance. There is a lack of large data sets in many sectors of application, including medical image analysis. This study examines the data-space solution known as data augmentation, which was developed to address the problem of insufficient data. Training datasets can be improved in quantity and quality using a variety of methods collectively referred to as "data augmentation," which in turn allows for the development of more robust Deep Learning models. This survey delves into a variety of image augmentation techniques, including geometric transformations, color space enhancements, kernel filters, blending images, random erasing, feature space enhancements, adversarial training, generative adversarial networks, neural style transfer, and meta-learning. GAN-based augmentation methods receive extensive coverage in this overview. In addition to augmentation methods, this study will briefly examine additional facets of Data Augmentation, including augmentation during testing, the impact of resolution, the final dataset size, and curriculum-based learning. This survey will describe current approaches to Data Augmentation, as well as promising developments and meta-level considerations for using Data Augmentation. This

Do Not Copy, Lead City University, Nigeria

book will teach you how to utilize data augmentation to improve the accuracy of your models and make better use of limited data sets by leveraging the power of big data.

Low-Quality Natural Image Text Detection and Classification The detection of textual data from scene text images is a challenging problem in computer graphics and visualisation[168]. The challenge gets much more challenging when intelligent devices at the periphery are involved. As a result of difficulties including blur, low resolution, and low contrast, text detection, and categorization are more challenging in this low-quality image. Therefore, such an important concern is factored into the study. The proposed technology is comprised of three main contributions. (a) Following synthetic blurring, the blurred image undergoes preprocessing before being restored by the deblurring method. (B) The tried-and-true maximal stable extreme regions (MSER) method is then used to recognize and locate text. The query image is then processed by K-Means to isolate three distinct clusters: one each for the foreground, background, and character levels. A unique convolutional neural network (CNN) structure is then used to classify the segmented text into textual and non-textual regions (c). For this project, you will be tasked with reducing the number of false positives. The effectiveness of the proposed method was determined by analyzing results from the widely used datasets SVT, IIIT5K, and ICDAR 2003. The classification rate for the SVT dataset was 90.3%, the classification rate for the IIIT5K dataset was 95.8%, and the classification rate for the ICDAR 2003 dataset was 94%. Proof that the proposed method is well-suited for model learning indicates the method's superiority. Finally, the proposed approach is compared to established baseline text-detection systems to verify the validity of the results obtained.

Two Decades of Progress in Texture Representation and Classification, from B&W to CNN As a fundamental property of several image formats, texture representation has long been the

Do Not Copy, Lead City University, Nigeria

subject of intense scholarly investigation [164]. Since the year 2000, researchers have looked at texture representations based on Bag of Words and Convolutional Neural Networks, and their findings have been encouraging. It is the goal of this work to provide a comprehensive evaluation of the advancements in texture representation during the past two decades. Nearly 250 articles representing the state-of-the-art in the subject are referenced in this survey, covering topics as diverse as benchmark datasets and state-of-the-art results. This review takes stock of past efforts and looks ahead to upcoming challenges and research opportunities.

Overfitting in text recognition algorithms is a problem caused by a lack of data, and this paper explores and recommends several ways to address this issue. Multiple machine learning strategies rely on large datasets to reduce the likelihood of overfitting. By artificially inflating datasets using the techniques discussed in this survey, the benefits of big data can be realized in the limited data environment. The quality of datasets can be greatly enhanced by using data augmentation. There are numerous proposed enhancements, most of which can be categorized as data warping or oversampling methods.

Do Not Copy, Lead City University, Nigeria

2.11 Summary of Literature Reviewed

From all the research made in the course of this study, it is observed that machine learning and deep learning so far remain the best solutions for computer vision, Image and data fusion, Image and data augmentation, and pattern recognition problems such as this. Again, Dimensionality Reduction (DR) is one ML technique that is said to be reliable for recognition and detection of text in the wild. With its feature extraction and selection methods, it allows scholars to come up with hybrid methods of DR that fine-tune the dataset that is being worked on to achieve the expected result. This is one of the reasons that birthed this study, where a hybrid of DR is used to decipher what the text on unclear scenery pictures is saying. A

combination of already existing DR methods (ICA), ML techniques (GA), and hand-crafted feature discovery methods will be novel for this study.

Do Not Copy, Lead City University, Nigeria

Endnotes

[1] X. Sun, Q. Wang, X. Zhang, X. Chen, & W. Zhang. "*Deep Blur Detection Network with Boundary-Aware Multi-Scale Features*." **Connection Science 34**, no. 1 doi:10.1080/09540091.2021.1933906. June 2022: 766–784.

[2]S. Misra & Y. Wu, "*Machine Learning Assisted Segmentation of Scanning Electron Microscopy Images of Organic-Rich Shales with Feature Extraction and Feature Ranking,*" in *Machine Learning for Subsurface Characterization* **Elsevier**, 2019, 289–314.

[3]S. Tiwari, V. Shukla, S. Biradar & A. Singh "*Blur Parameters Identification for Simultaneous Defocus and Motion Blur*," **CSI Transactions on ICT** 2, no. 1, March 2014: 11–22.

[4]D. Cao, Y. Zhong, L. Wang, Y. He & J. Dang "*Scene Text Detection in Natural Images: A Review,*" **Symmetry 12**, no. 12 https://www.mdpi.com/2073-8994/12/12/1956. November 2020: 1–26.

[5]J. Fabrizio, B. Marcotegui & M. Cord, "*Text Detection in Street Level Images*," ***Pattern Analysis and Applications* 16**, no. 4 November 2013: 519–533.

[6]S. Uchida, "*Text Localization and Recognition in Images and Video*," in **Handbook of Document Image Processing and Recognition** London: Springer London, 2014, 843–883.

[7]Q. Ye & D. Doermann, "*Text Detection and Recognition in Imagery: A Survey*," **IEEE Transactions on Pattern Analysis and Machine Intelligence 37**, no. 7, July 2015: 1480–1500.

[8]F. C. Monteiro & A. C. Campilho, "*Performance Evaluation of Image Segmentation,*" *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4141 LNCS 2006: 248–259.

[9]Y. Zhu, C. Yao & X. Bai, "*Scene Text Detection and Recognition: Recent Advances and Future Trends,*" **Frontiers of Computer Science 10**, no. 1 February 2016: 19–36.

[10]R. Sarkar, S. Malakar, N. Das, S. Basu & M. Kundus "*Word Extraction and Character Segmentation from Text Lines of Unconstrained Handwritten Bangla Document Images*," **Journal of Intelligent Systems 20,** no. 3 January 2011: 227–260.

[11]Y. Wei, Z. Zhang, W. Shen, D. Zheng, M. Fang & S. Zhou "*Text Detection in Scene Images Based on Exhaustive Segmentation*," **Signal Processing: Image Communication 50**, https://linkinghub.elsevier.com/retrieve/pii/S0923596516301540. February 1, 2017: 1–8.

[12]H. Il Koo, "*Text-Line Detection in Camera-Captured Document Images Using the State Estimation of Connected Components*," **IEEE Transactions on Image Processing 25**, no. 11 November 2016: 5358–5368.

[13]M. Sumathi & T. Balaji, "*Improved Connected Component Labeling Algorithm for Remote Sensing Image Classification*," **International Journal of Scientific Research in Computer Science, Engineering and Information Technology 3307,** 2021: 294–303.

Do Not Copy, Lead City University, Nigeria

[14]L. Armi & S. Fekri-Ershad, "*Texture Image Analysis and Texture Classification Methods - A Review*" 2019, http://arxiv.org/abs/1904.06554.

[15] H. Koo. "*Text-Line Detection in Camera-Captured Document Images Using the State Estimation of Connected Components*." **IEEE Transactions on Image Processing 25**, no. 11 November 2016: 5358–5368.

[16]H. Wu, "*Texture Image Classification Method of Porcelain Fragments Based on Convolutional Neural Network*," ed. Syed Hassan Ahmed, **Computational Intelligence and Neuroscience 2021,** June 2021: 1–10.

[17]J. Wang, R. Chen, M. Liu & P-C Liao *"Research Trends of Human–Computer Interaction Studies in Construction Hazard Recognition: A Bibliometric Review*," ***Sensors* 21**, no. 18, September 2021: 6172.

[18]A. Subasi, "*Machine Learning Techniques*," in **Practical Machine Learning for Data Analysis Using Python Elsevier**, 2020, 91–202.

[19]M. Algren, W. Fisher, & A. E. Landis, "*Machine Learning in Life Cycle Assessment*," in **Data Science Applied to Sustainability Analysis Elsevier**, 2021, 167–190.

[20]J. J. Ranjani & C. Jeyamala, "*Machine Learning Algorithms for Medical Image Security,*" in **Intelligent Data Security Solutions for E-Health Applications Elsevier**, 2020, 169–183.

[21]J. E. van Engelen & H. H. Hoos, "*A Survey on Semi-Supervised Learning*," **Machine Learning 109**, no. 2, February 2020: 373–440.

[25]I. H. Witten, "*Beyond Supervised and Unsupervised Learning*," in **Data Mining Elsevier,** 2017, 467–478.

[26]J. M. Lodge, G. Kennedy, L. Lockyer, A. Arguel & M. Pachman "*Understanding Difficulties and Resulting Confusion in Learning: An Integrative Review*," **Frontiers in Education 3** June 2018.

[27]S. Ayesha, Mk Hanif, & R. Talib, "*Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data*." **Information Fusion 59** July 2020: 44–58.

[28]S. Khalid, T. Khalil, & S. Nasreen, "*A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning*," in **Proceedings of 2014 Science and Information Conference, SAI** *2014* IEEE, 2014, 372–378.

[29]S. V. Patil & D. B. Kulkarni, "*A Review of Dimensionality Reduction in High-Dimensional Data Using Multi-Core and Many-Core Architecture*," in **Communications in Computer and Information Science, vol. 964**, 2019, 54–63.

[31]S Suganya, "*Analysis of Feature Extraction of Optical Character Detection in Image Processing Systems*" 2015: 1–8.

Do Not Copy, Lead City University, Nigeria

[32]S. Jain & A. O. Salau, "*An Image Feature Selection Approach for Dimensionality Reduction Based on KNN and SVM for AkT Proteins*," ed. Wei Meng, **Cogent Engineering 6,** no. 1 January 2019.

[34] A. Slowik & H. Kwasnicka, "*Evolutionary Algorithms and Their Applications to Engineering Problems*," **Neural Computing and Applications 32**, no. 16, August 2020: 12363–12379.

[35]Y. Liu, J - M Wu, M. Avdeev, & S- Q Shi "*Multi-Layer Feature Selection Incorporating Weighted Score-Based Expert Knowledge toward Modeling Materials with Targeted Properties,*" **Advanced Theory and Simulations 3**, no. 2, February 2020: 1900215.

[36] P. Bansal, R. Lamba,V. Jain, T Jain, S. Shokeen, S. Kumar, Singh P. K. & B. Khan. "*GGA-MLP: A Greedy Genetic Algorithm to Optimize Weights and Biases in Multilayer Perceptron*." Edited by Yuvaraja Teekaraman. **Contrast Media & Molecular Imaging** 2022. February 2022: 1–14.

[37] V. R Messias, J. C. Estrella, R. Ehlers, M. J Santana, R. C Santana, & S. Reiff-Marganiec. "*Combining Time Series Prediction Models Using Genetic Algorithm to Autoscaling Web Applications Hosted in the Cloud Infrastructure.*" **Neural Computing and Applications 27**, no. 8, November 2016: 2383–2406.

[38] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri & V. B. S Prasath. "*Choosing Mutation and Crossover Ratios for Genetic Algorithms-a Review with a New Dynamic Approach*." **Information (Switzerland) 10**, no. 12, 2019.

[39]A. Slowik & H. Kwasnicka, "*Evolutionary Algorithms and Their Applications to Engineering Problems*," **Neural Computing and Applications 32**, no. 16, August 2020: 12363–12379.

[40]A. Alajmi and J. Wright, "*Selecting the Most Efficient Genetic Algorithm Sets in Solving Unconstrained Building Optimization Problem*," **International Journal of Sustainable Built Environment 3,** no. 1 June 2014: 18–26.

[41]M. M. Mijwel, "*Genetic Algorithm Optimization by Natural Selection,*" **Computer Science,College of Science 1**, no. 1 2016: 1–6.

[42]E. Sevinc, "A Novel Evolutionary Algorithm for Data Classification Problem with Extreme Learning Machines," *IEEE Access* 7 2019: 122419–122427.

[43]S. Katoch, S. S. Chauhan, & V. Kumar, *A Review on Genetic Algorithm: Past, Present, and Future*, *Multimedia Tools and Applications*, vol. 80 **Multimedia Tools and Applications**, 2021, http://link.springer.com/10.1007/s11042-020-10139-6.

[44]H. Ahmad, A. Khalid A. Esra'a, A. Eman, H. Awni & P. Surya "*Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach,*" **Information MDPI** 2019.

[45]F. U. Ogban & R. Nentui, "P*heromone Deposition/Updating Strategy in a Network: Using Ant Colony Optimization (ACO) Approach*," **Global Journal of Pure and Applied Sciences 24**, no. 2 December 2019: 215–222.

Do Not Copy, Lead City University, Nigeria

[46]C. E. Braun, L. D. Chiwiacowsky, A. T. Gómez, "*Variations of Ant Colony Optimization for the Solution of the Structural Damage Identification Problem*," **Procedia Computer Science 51**, no. 1 2015: 875–884.

[47] W. Jia, M. Sun, J. Lian, & S. Hou. "*Feature Dimensionality Reduction: A Review.*" **Complex and Intelligent Systems 8**, no. 3 January 2022: 2663–2693.

[48]E. Knekta, C. Runyon & S. Eddy, "*One Size Doesn't Fit All: Using Factor Analysis to Gather Validity Evidence When Using Surveys in Your Research*," ed. Peggy Brickman, *CBE* **Life Sciences Education 18**, no. 1, March 2019: rm1.

[49] M. A Abu, N. H. Indra, A. Abd Rahman, N. Sapiee, & I. Ahmad. *A study on Image Classification based on Deep Learning and Tensorflow.* 12. 2019: 563-569.

[50]W. Jia, M. Sun, J. Lian, & S. Hou. "*Feature Dimensionality Reduction: A Review.*" **Complex and Intelligent Systems 8**, no. 3 January 2022: 2663–2693.

[51]D. Ustun, A. Toktas, U. Erkan & A. Akdagli. '*Modified artificial bee colony algorithm with differential evolution to enhance precision and convergence performance'*, **Expert Systems with Applications,** Volume 198, 116930, ISSN 0957-4174, doi:10.1016/j.eswa.2022.116930, 2022.

[52]H. Sharma, J. C. Bansal, K. V. Arya & Xin-She Yang. *Lévy flight artificial bee colony algorithm*, **International Journal of Systems Science,** 47:11, 2652-2670, DOI: 10.1080/00207721.2015.1010748, 2016.

[53] B. Crawford, R. Soto, R Cuesta & F. Paredes "*Application of the Artificial Bee Colony Algorithm for Solving the Set Covering Problem,"* **The Scientific World Journal** 2014 2014: 1–8.

[54]T. Davidovic, D. Teodorovic, & M. Selmic, "*Bee Colony Optimization - Part I: The Algorithm Overview,*" **Yugoslav Journal of Operations Research 25**, no. 1, 2015: 33–56.

[55]B. Yuce, M. S Packianather, E. Mastrocinque, DT Pham & A. Lambiase. '*Honey Bees Inspired Optimization Method: The Bees Algorithm. Insects'.* 2013 Nov 6;4(4):646-62. doi: 10.3390/insects4040646. PMID: 26462528; PMCID: PMC4553508. November 2013.

[56]D. Teodorovic, M. Selmic, & T. Davidovic, "*Bee Colony Optimization - Part II: The Application Survey*," **Yugoslav Journal of Operations Research 25,** no. 2 2015: 185–219.

[57]X S Yang, "Firefly Algorithms," in *Nature-Inspired Optimization Algorithms* **Elsevier**, 2021, 123–139.

[58]K. Chaudhari & A. Thakk. 2019. *Travelling Salesman Problem: An Empirical Comparison Between ACO, PSO, ABC, FA and GA*. In: Shetty, N., Patnaik, L., Nagaraj, H., Hamsavath, P., Nalini, N. (eds) **Emerging Research in Computing, Information, Communication and Applications. Advances in Intelligent Systems and Computing, vol 906. Springer**, Singapore. doi:10.1007/978-981-13-6001-5_32. 2019.

Do Not Copy, Lead City University, Nigeria

[59]M. K Sahoo, J. Nayak, S. Mohapatra, B. K. Nayak &H. S Behera. '*Character Recognition Using Firefly Based Back Propagation Neural Network'*. In: Jain, L., Behera, H., Mandal, J., Mohapatra, D. (eds) **Computational Intelligence in Data Mining - Volume 2. Smart Innovation, Systems and Technologies**, vol 32. Springer, New Delhi. doi:10.1007/978-81-322-2208-8_15. 2015.

[60]W. Lin, Z. Lian, X. Gu & B. Jiao. '*A local and global search combined particle swarm optimization algorithm and its convergence analysis'*. **Mathematical Problems in Engineering**. doi:10.1155/2014/905712. 2014.

[61]A. S. Ashour & Y. Guo, "*Optimization-Based Neutrosophic Set in Computer-Aided Diagnosis*," in **Optimization Theory Based on Neutrosophic and Plithogenic Sets. Elsevier**, 2020, 405–421.

[62]D. Freitas, L. G. Lopes, & F. Morgado-Dias, "*Particle Swarm Optimisation: A Historical Review up to the Current Developments,*" **Entropy 22,** no. 3 March 2020: 362.

[63] R. Abu Khurma, I. Aljarah, A. A. Sharieh, M. A. Abd Elaziz, R. Damaševičius, & T. Krilavičius. 2022. '*A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem'*. ***Mathematics*.10**, no. 3 January 2022: 464.

[64] H. Sharma, J. C. Bansal, K. V. Arya & Xin-She Yang. *Lévy flight artificial bee colony algorithm*, **International Journal of Systems Science,** 47:11, 2652-2670, DOI: 10.1080/00207721.2015.1010748. 2016.

[65] W. A. Khan, N. N. Hamadneh, S. L. Tilahun & J. M. T. Ngnotchouye. *"A Review and Comparative Study of Firefly Algorithm and Its Modified Versions*," in **Optimization Algorithms - Methods and Application***s* inTech, 2016.

[66]S. Sayah & A. Hamouda, "*A Hybrid Differential Evolution Algorithm Based on Particle Swarm Optimization for Nonconvex Economic Dispatch Problems*," **Applied Soft Computing Journal 13,** no. 4 April 2013: 1608–1619.

[67]K K. Kumar, K. Chaduvula, & B. Rao Markapudi, "*A Detailed Survey On Feature Extraction Techniques In Image Processing For Medical Image Analysis,*" **European Journal of Molecular & Clinical Medicine 7**, no. 10 2021: 2275–2284.

[68]R. R. Zebari, A. M. Abdulazeez, D. Q. Zeebaree, D. A. Zebari & J. N. Saeed. *"A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction,*" **Journal of Applied Science and Technology Trends 1**, no. 2, doi:10.38094/jastt1224. 2020: 56–70

[69]E. Fantin . Raj & M. Balaji, "*Application of Deep Learning and Machine Learning in Pattern Recognition*," 2022, 63–89.

[70] V. Prasad & Y. Jayanta, "(PDF) *A Study on Method of Feature Extraction for Handwritten Character Recognition,"* **Indian Journal of Science and Technology 6**, no. S3 https://www.researchgate.net/publication/258029618_A_study_on_method_of_feature_extraction_for_Handwritten_Character_Recognition. 2013: 174–178.

Do Not Copy, Lead City University, Nigeria

[71]P. Soto-Quiros & A. Torokhti, "*Extended Principal Component Analysis*" 2021, http://arxiv.org/abs/2111.03040.

[72] W. Jia, M. Sun, J. Lian, & S. Hou. "*Feature Dimensionality Reduction: A Review.*" **Complex and Intelligent Systems 8**, no. 3 January 2022: 2663–2693.

[73]L. BruntonJ. Steven & Kutz Nathan, "*Singular Value Decomposition (SVD),*" in ***Data-Driven Science and Engineering*** **Cambridge University Press**, 2019, 3–46.

[74]I. T. Jollife & J. Cadima, "*Principal Component Analysis: A Review and Recent Developments,*" ***Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*** 374, no. 2065: 2065, doi:10.1098/rsta.2015.0202. April 2016

[75]Soto-Quiros, Pablo, & A. Torokhti. "*Extended Principal Component Analysis*" 2021. http://arxiv.org/abs/2111.03040.

[76]S. Zafeiriou, "*Notes on Implementation of Component Analysis Techniques,*" no. January 2015: 1–5.

[77]I T. Jollife, & C. Jorge. "*Principal Component Analysis: A Review and Recent Developments*." **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374**, no. 2065 April 2016: 2065. https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202.

[78]I. M. Johnstone & A. Yu Lu, "*On Consistency and Sparsity for Principal Components Analysis in High Dimensions,*" ***Journal*** **of the American Statistical Association 104**, no. 486 June 2009: 682–693.

[79]S. P. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. P. Swain, R. Saikhom, S. Panda M. Laishram. "Principal Component Analysis," *International Journal of Livestock Research :* 1 2017, http://www.ejmanager.com/fulltextpdf.php?mno=261590.

[80]**Y. Chen,** J. Tao, Q. Zhang, K. Yang, X. Chen, J. Xiong, R. Xia, & J. Xie. "*Saliency Detection via the Improved Hierarchical Principal Component Analysis Method*," **Wireless Communications and Mobile Computing** May 2020: 1–12.

[81]Z. Ge, C. Yang, & Z. Song, "I*mproved Kernel PCA-Based Monitoring Approach for Nonlinear Processes,*" **Chemical Engineering Science 64,** no. 9 May 2009: 2245–2255.

[82]E. Postma, "*Dimensionality Reduction : A Comparative Review Dimensionality Reduction : A Comparative Review*," **Journal of Machine Learning Research 10**, no. October 2016 2007: 1–35.

[83]R. Rosipal, L. J. Trejo, & A. Cichocki, "*Kernel Principal Component Regression with EM Approach to Nonlinear Principal Components Extraction*," **Computing and Information Systems Technical Reports 12,** no. December 2000: 1–42, http://cis.paisley.ac.uk/research/reports.

Do Not Copy, Lead City University, Nigeria

[84]P. Honeine, "*Online Kernel Principal Component Analysis: A Reduced-Order Model,*" *IEEE* **Transactions on Pattern Analysis and Machine Intelligence 34**, no. 9 September 2012: 1814–1826.

[85]C. Chen & K. Xie, "*Face Recognition Based on Two-Dimensional Principal Component Analysis and Kernel Principal Component Analysis*," **Information Technology Journal 11,** no. 12 2012: 1781–1785.

[86]F. Zhao, I. Rekik, S-W Lee, J.Liu, J. Zhang, & **D. Shen** "*Two-Phase Incremental Kernel PCA for Learning Massive or Online Datasets,*" **Complexity 2019**, February 2019: 1–17.

[87]A. Hyvärinen, "*Independent Component Analysis: Recent Advances,*" **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371**, no. 1984 February 2013: 20110534, https://royalsocietypublishing.org/doi/10.1098/rsta.2011.0534.

[88]K. Nordhausen & H. Oja, "*Independent Component Analysis: A Statistical Perspective*," ***Wiley Interdisciplinary Reviews: Computational Statistics*** **10,** no. 5 September 2018.

[89]F. Anowar, S. Sadaoui, & B. Selim, "*Conceptual and Empirical Comparison of Dimensionality Reduction Algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE),"* **Computer Science Review, 2021,** accessed November 4, 2022, doi:10.1016/j.cosrev.2021.100378. 2022

[90]A. Hyvärinen. "*Independent Component Analysis: Recent Advances*." **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371**, no. 1984 February 2013: 20110534. https://royalsocietypublishing.org/doi/10.1098/rsta.2011.0534.

[91]D. K Naik, G. R & Kumar, "*An Oversview of Independent Component Analysis and Its Applications,*" **Informatica (Ljubljana) 35,** no. 1 2011: 63–81.

[92]Murinto & A. Harjoko, "*Dataset Feature Reduction Using Independent Component Analysis with Contrast Function of Particle Swarm Optimization on Hyperspectral Image Classification*," in **Proceeding - 2016 2nd International Conference on Science in Information Technology, ICSITech 2016: Information Science for Green Society and Environment IEEE**, 2017, 285–290.

[93]Jian-Xun Xun Mi, "*A Novel Algorithm for Independent Component Analysis with Reference and Methods for Its Applications*," ed. Hans A. Kestler, **PLoS ONE 9,** no. 5 May 2014: e93984.

[94]K.Raju, Y.Srinivasa Rao, & M.Narsing Yadav, "*Performance Analysis of PCA and* LDA," **International Journal of Innovative Research in Electronics and Communications 2**, no. 2 2015: 17–22, www.arcjournals.org.

[95]S. Madhavan & N. Kumar, "*Incremental Methods in Face Recognition: A Survey,"* **Artificial Intelligence Review 54,** no. 1 January 2021: 253–303.

Do Not Copy, Lead City University, Nigeria

[96]P. Mishra, C, Pandey, U. Singh, A. Keshri & M. P Sabaretnam. "*Selection of Appropriate Statistical Methods for Data Analysis,*" **Annals of Cardiac Anaesthesia 22**, no. 3 2019: 297–301.

[97]D. Zhang, Xiao-Yuan Jing, & J. Yang, "*Linear Discriminant Analysis*" 2011: 41–64.

[99] S. Zafeiriou. "*Notes on Implementation of Component Analysis Techniques,"* no. January 2015: 1–5.

[100]V.S. Sumithra & S. Surendran, "*A Review of Various Linear and Non Linear Dimensionality Reduction Techniques,"* **International Journal of Computer Science and Information Technologies 6, no. 3 2015:** 2354–2360.

[101]F. S. Tsai, "*Comparative Study of Dimensionality Reduction Techniques for Data Visualization*," **Journal of Artificial Intelligence 3**, no. 3 2010: 119–134.

[102]F. Anowar, S. Samira, & S. Bassant. "*Conceptual and Empirical Comparison of Dimensionality Reduction Algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE).*" **Computer Science Review**, 2021. Accessed November 4, 2022. https://sci-hub.st/https://doi.org/10.1016/j.cosrev.2021.100378.

[103]C. Bartenhagen, H-U Klein, C. Rucket, X. Jiang & M. Dugas. "*Comparative Study of Unsupervised Dimension Reduction Techniques for the Visualization of Microarray Gene Expression Data*," **BMC Bioinformatics 11**, no. 1 December 2010: 567.

[104]N. Varghese, "*A Survey Of Dimensionality Reduction And Classification Methods,*" **International Journal of Computer Science & Engineering Survey 3,** no. 3 2012: 45–54.

[105]J. Clark & F. Provost, "*Unsupervised Dimensionality Reduction versus Supervised Regularization for Classification from Sparse Data*," **Data Mining and Knowledge Discovery 33**, no. 4 July 2019: 871–916.

[106]L. Ziegelmeier, M. Kirby, & C. Peterson, "*Sparse Locally Linear Embedding,*" **Procedia Computer Science 108** 2017: 635–644.

[107]W. K. Härdle & L. Simar, "*Canonical Correlation Analysis,*" in **Applied Multivariate Statistical Analysis** Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, 385–395.

[108]A. Golugula "*Supervised Regularized Canonical Correlation Analysis: Integrating Histologic and Proteomic Measurements for Predicting Biochemical Recurrence Following Prostate Surgery,*" **BMC Bioinformatics 12,** no. 1 December 2011: 483.

[109]Y. Lu & D. P. Foster, "*Large Scale Canonical Correlation Analysis with Iterative Least Squares,*" **Advances in Neural Information Processing Systems 1,** no. January 2014: 91–99.

[110]A. E. Maxwell, T. A. Warner, & F, Fang, "*Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review,*" **International Journal of Remote Sensing 39**, no. 9 May 2018: 2784–2817.

Do Not Copy Lead City University, Nigeria

[111]S. B. Kotsiantis, I. D. Zaharakis, & P. E. Pintelas, "*Machine Learning: A Review of Classification and Combining Techniques*," **Artificial Intelligence Review 26**, no. 3 November 2006: 159–190.

[112]F. Hutter, L. Xu, H. H. Hoos, & K. Leyton-Brown. "*Algorithm Runtime Prediction: Methods & Evaluation*," **Artificial Intelligence 206** January 2014: 79–111.

[113]I. Zoppis, G. Mauri, & R. Dondi, "*Kernel Methods: Support Vector Machines,*" in **Encyclopedia of Bioinformatics and Computational Biology, vol. 1–3 Elsevier**, 2018, 503–510.

[114]J. Schmidt, M. R. G. Marques, S. Botti & M. A. L. MArques. "*Recent Advances and Applications of Machine Learning in Solid-State Materials Science,"* **npj Computational Materials 5**, no. 1 December 2019: 83.

[115]T. Kavzoglu & I. Colkesen, "*A Kernel Functions Analysis for Support Vector Machines for Land Cover Classification*," **International Journal of Applied Earth Observation and Geoinformation 11**, no. 5 October 2009: 352–359.

[116]I. H. Sarker, A. S.M. Kayes, & P. Watters, "*Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-Aware Smartphone Usage*," **Journal of Big Data 6**, no. 1 December 2019: 57.

[117]M. Awad & R. Khanna, "*Support Vector Machines for Classification,*" in **Efficient Learning Machines Berkeley**, CA: Apress, 2015, 39–66.

[118]V. Viitaniemi, M. Sjöberg, M. Koskela, S. Ishikawa & Jorma Laaksonen, "*Advances in Visual Concept Detection: Ten years of TRECVID*',. Editor(s): Ella Bingham, Samuel Kaski, Jorma Laaksonen, Jouko Lampinen in **Advances in Independent Component Analysis and Learning Machines Elsevier,** 2015, 249–278.

[119]K. Harimoorthy & M. Thangavelu, "*Multi-Disease Prediction Model Using Improved SVM-Radial Bias Technique in Healthcare Monitoring System*," **Journal of Ambient Intelligence and Humanized Computing 12**, no. 3 March 2021: 3715–3723.

[120]Yun Yang, "*Ensemble Learning*," in **Temporal Data Mining Via Unsupervised Ensemble Learning** Elsevier, 2017, 35–56.

[121]D. Cha, C. Pae, S-B. Seong, J. Y. Choi & H-J. Park,, "*Automated Diagnosis of Ear Disease Using Ensemble Deep Learning with a Big Otoendoscopy Image Database*," **EBioMedicine 45** 2019: 606–614.

[122]T. G. Dietterich, "*Ensemble Methods in Machine Learning*," **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)** 1857 LNCS 2000: 1–15.

[123]P Verma, A. Dumka, R. Singh, A. Ashok, A. Gehlot, P. K. Malik, G. S. Gaba, & M. Hedabou, "*A Novel Intrusion Detection Approach Using Machine Learning Ensemble for IoT Environments,*" **Applied Sciences 11, no. 21** November 2021: 10268.

Do Not Copy, Lead City University, Nigeria

[124]L Alzubaidi, J. Zhang, AJ Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, MA Fadhel, M. Al-Amidie, & L. Farhan, "*Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*," **Journal of Big Data 8, no**. 1 December 2021: 53.

[125]S. Tewari & U. D. Dwivedi, "*A Comparative Study of Heterogeneous Ensemble Methods for the Identification of Geological Lithofacies*," **Journal of Petroleum Exploration and Production Technology 10,** no. 5 June 2020: 1849–1868.

[126]Yun Yang, "*Ensemble Learning*," in **Temporal Data Mining Via Unsupervised Ensemble Learning Elsevier**, 2017, 35–56.

[127]M. Ayaz, F. Shaukat, & G. Raja, "*Ensemble Learning Based Automatic Detection of Tuberculosis in Chest X-Ray Images Using Hybrid Feature Descriptors*," **Physical and Engineering Sciences in Medicine 44, no. 1** March 2021: 183–194.

[128]R. Abdulhammed, "*Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection*," **Electronics 8, no. 3** March 2019: 322.

[129]C. Vens, "*Random Forest*," in **Encyclopedia of Systems Biology New York, NY: Springer New York**, 2013, 1812–1813.

[130]Y. Mansour & M. Schain, "*Learning with Maximum-Entropy Distributions*," **Machine Learning 45**, no. 2 2001: 123–145.

[131]T. Sanlı, Ç. Sıcakyüz, & O.H. Yüregir, "*Comparison of the Accuracy of Classification Algorithms on Three Data-Sets in Data Mining: Example of 20 Classes*," **International Journal of Engineering, Science and Technology 12**, no. 3 2020: 81–89.

[132]S. B. Kotsiantis, "*Decision Trees: A Recent Overview*," **Artificial Intelligence Review 39, no. 4** April 2013: 261–283.

[133]A. S. Sadiq, H. Faris, A M. Al-Zoubi, S. Mirjalili & K. Z. Ghafoor, "*Fraud Detection Model Based on Multi-Verse Features Extraction Approach for Smart City Applications*," in **Smart Cities Cybersecurity and Privacy Elsevier,** 2018, 241–251.

[134]B. Charbuty & A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," **Journal of Applied Science and Technology Trends 2, no. 01** 2021: 20–28.

[135]A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "*Overview of use of decision tree algorithms in machine learning*," ***2011 IEEE Control and System Graduate Research Colloquium***, Shah Alam, Malaysia, doi: 10.1109/ICSGRC.2011.5991826, 2011, pp. 37-42.

[136]Y. Y. Song & Y. Lu, "*Decision Tree Methods: Applications for Classification and Prediction*," **Shanghai Archives of Psychiatry 27, no. 2** 2015: 130–135.

[137]A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh & A. A. Alhasanat, "*Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach*" **International Journal of Computer Science and Information Security, International**

Do Not Copy, Lead City University, Nigeria

Journal of Computer Science and Information Security, Vol. 12, No. 8, August 2014, http://arxiv.org/abs/1409.0919.

[138]C. Zhang, P. Zhong, M. Liu, Q. Song, Z. Liang & Xiao Wang, "*Hybrid Metric K-Nearest Neighbor Algorithm and Applications,"* ed. Luis Payá, **Mathematical Problems in Engineering** January 2022: 1–15.

[139]H Novitasari, N. Hadianto, S. Sfenrianto, A. Rahmawati, R. Prasetyo, J. Miharja, & W. Gata, "*K-Nearest Neighbor Analysis to Predict the Accuracy of Product Delivery Using Administration of Raw Material Model in the Cosmetic Industry (PT Cedefindo)*," **Journal of Physics: Conference Series 1367, no. 1** November 2019: 012008.

[140]M. Zhao & J. Chen, "*Improvement and Comparison of Weighted k Nearest Neighbors Classifiers for Model Selection*," **Journal of Software Engineering 10, no. 1** 2016: 109–118.

[141]D. Chanal, N. Steiner, P. Raffaele, D. Chamagne, & M-C. Marion-Péra, "*Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering*," in **Reference Module in Earth Systems and Environmental Sciences Elsevier,** 2021.

[142]H. Wang, P. Xu, & J. Zhao, "I*mproved KNN Algorithm Based on Preprocessing of Center in Smart Cities*," **Complexity 2021**.

[143]L. Wang, "*Research and Implementation of Machine Learning Classifier Based on KNN*," **IOP Conference Series: Materials Science and Engineering 677, no. 5** December 2019: 052038.

[144]I. H. Sarker, "Deep Learning: *A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions*," **SN Computer Science 2**, no. 6 November 2021: 420.

[145]N. L.W. Keijsers, "*Neural Networks,*" in **Encyclopedia of Movement Disorders Elsevier**, 2010, 257–259.

[146]Y. Huang & L. Li, "*Naive Bayes Classification Algorithm Based on Small Sample Set,*" in **CCIS2011 - Proceedings: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems IEEE**, 2011, 34–39.

[147]T. Sanlı,, Ç. Sıcakyüz, & O.H. Yüregir. "*Comparison of the Accuracy of Classification Algorithms on Three Data-Sets in Data Mining: Example of 20 Classes*." **International Journal of Engineering, Science and Technology 12**, no. 3 2020: 81–89.

[148]S. Anand, S. Susan, S. Aggarwal, S. Aggarwal & R. Singla, "*Scene Text Recognition in the Wild with Motion Deblurring Using Deep Networks*," in **Communications in Computer and Information Science, vol. 1378 CCIS**, 2021, 93–103.

[149]S. Long, X. He, & C. Yao, "*Scene Text Detection and Recognition: The Deep Learning Era*," **International Journal of Computer Vision 129,** no. 1 January 2021: 161–184, https://doi.org/10.1007/s11263-020-01369-0.

Do Not Copy, Lead City University, Nigeria

[150]W. Zhu, J. Lou, L. Chen, Q. Xia, M. Ren, "*Scene Text Detection via Extremal Region Based Double Threshold Convolutional Network Classification*," ed. Yuanquan Wang, ***PLoS ONE* 12,** no. 8, August 2017: e0182227.

[151]R. Yan & L. Shao, "I*mage Blur Classification and Parameter Identification Using Two-Stage Deep Belief Networks*," in **BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference 2013** British Machine Vision Association, 2013, 70.1-70.11.

[152]S. Tiwari, "*A Pattern Classification Based Approach for Blur Classification*," **Indonesian Journal of Electrical Engineering and Informatics** *(IJEEI)* 5, no. 2 June 2017.

[153]R. Wang, W. Li, & L. Zhang, "*Blur Image Identification with Ensemble Convolution Neural Networks,*" **Signal Processing 155** February 2019: 73–82.

[154]F. Wang, J. L. Zhang, Y. Li, K. Deng, & J. S. Liu, "*Bayesian Text Classification and Summarization via a Class-Specified Topic Model*," **Journal of Machine Learning Research 22** 2021.

[155]R. Wang, W. Li, R. Qin & J. Wu, "*Blur Image Classification Based on Deep Learning*," in **IST 2017 - IEEE International Conference on Imaging Systems and Techniques, Proceedings,** vol. 2018-January IEEE, 2017, 1–6.

[156] M. Bansal, A. Goyal & A. Choudhary. '*A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning,'* **Decision Analytics Journal**, Volume 3, 100071, ISSN 2772-6622, 2022, doi:10.1016/j.dajour.2022.100071.

[157]P. Hsu & Bing-Yu Yu Chen, "*Blurred Image Detection and Classification*," in **Lecture Notes in Computer Science** *(Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **vol. 4903 LNCS**, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 277–286.

[158]C. M. Kim, J. H, Ellen, C. Kyungyong & C. P. Roy "*Line-Segment Feature Analysis Algorithm Using Input Dimensionality Reduction for Handwritten Text Recognition*," **Applied Sciences (Switzerland) 10,** no. 19, October 2020: 1–17.

[159]A. Bera & D. Sychel, "*Features Extraction for Detection of Blurred Image Regions*," **Applied Artificial Intelligence 30**, no. 3 March 2016: 201–215.

[160] J. Pang, W. Sun, J. S Ren, C. Yang , & Q. Yan. "*Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching."* In **Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW** *2017*, 2018-January. IEEE, 2017:878–886.

[161]M. A. Shayegan & S.Aghabozorgi, "*A New Dataset Size Reduction Approach for PCA-Based Classification in OCR Application,*" **Mathematical Problems in Engineering** 2014: 1–14.

[162]S. Velliangiri, S. Alagumuthukrishnan, & S. Iwin Thankumar Joseph, "*A Review of Dimensionality Reduction Techniques for Efficient Computation,*" in **Procedia Computer Science**, vol. 165, 2019, 104–111.

Do Not Copy, Lead City University, Nigeria

[163]D. Endalie & T. Tegegne, "*Designing a Hybrid Dimension Reduction for Improving the Performance of Amharic News Document Classification*," ed. Thippa Reddy Gadekallu, ***PLoS ONE* 16,** no. 5 May 2021: e0251902.

[164]K. Hamad & M. Kaya, "*A Detailed Analysis of Optical Character Recognition Technology,*" **International Journal of Applied Mathematics, Electronics and Computers 4, no. Special Issue-1** December 2016: 244–244.

[165]W. Rawat & Z. Wang, "*Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,*" **Neural Computation**, September 2017.

[166] Z. Zou, Z. Shi, Y. Guo, & J Ye. "*Object Detection in 20 Years: A Survey*" May 2019. http://arxiv.org/abs/1905.05055

[167]C. Shorten & T. M. Khoshgoftaar, "*A Survey on Image Data Augmentation for Deep Learning*," **Journal of Big Data 6,** no. 1 December 2019: 60.

[168]L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, & M. Pietikäinen. "*From BoW to CNN: Two Decades of Texture Representation for Texture Classification.*" **International Journal of Computer Vision 127**, no. 1, January 2019: 74–109.

Do Not Copy, Lead City University, Nigeria

Chapter Three

Materials and Methods

3.1 Introduction

This chapter discusses the method to solve the problem established in previous chapters.

In solving this issue, a hybrid dimensionality reduction technique will extract relevant data from a dataset while minimizing the quality loss inherent in data compression. Due to the high dimensionality of text recognition input, this research proposes enhancing/improving the Genetic Algorithm (BA- GA) with an Independent Component Analysis (ICA). **(BA-GA – ICA).** The development and design of the models for this study emanate from the dataset to be used, which Is the ICDAR 2019 SLVT.

3.1.1 Large-Scale Street View 2019 ICDAR Text (LSVT19)

The LSVT collection includes 450,000 photos with text from the streets, such as stores and landmarks. 30,000 annotated images are used for training, and 20,000 test photos are used for testing out of 50,000 annotated examples. The remaining 400,000 photos comprise the training set and are only minimally labeled. Each photo was taken with a separate mobile device and by a different person. Algorithms can detect and obscure sensitive information, such as faces and license plates, to protect users' anonymity[1,2,3]. Written and printed materials were scanned and included in the ICDAR 2019 dataset.

This dataset, like others like it, came about through a challenge. This challenge focuses on scene text reading in natural photos, which can be broken down into scene text detection and spotting challenges, using the proposed Large-scale Street View Text with Partial Labelling (LSVT) dataset as a basis. LSVT dataset is at least 14x larger than the most advanced reading benchmarks. Moreover, it is the first scene text dataset with partial annotations for use in text

Do Not Copy, Lead City University, Nigeria

detection and identification competitions. Additionally, a larger percentage of the data set is wholly annotated than in past robust reading benchmarks. All of the photographs were captured on public streets, which provide a diverse set of complicated real-world scenarios like stores and landmarks, making the challenge exceptionally challenging by closing the gap between academic study and practical application.

The data used in ICDAR2019 can be accessed by anyone interested in the study. You can find it in the repository at https://rrc.cvc.uab.es/?ch=16&com=introduction.

3.1.2 Python Programming Language

Python programming language is to be used for this study for several reasons, such as it is easy to learn and a flexible language that has simple syntax, platform-agnosticism, and an abundance of libraries and frameworks. These characteristics make it a popular choice for machine learning engineers, mainly due to the prevalence of machine learning-related Python libraries. Because of its ubiquity, it is also easy to find support and help with Python questions. Some of the most popular tools for completing machine learning tasks with Python include Tensorflow, Cy-Kit Learn, Pytorch, OpenCV, Theano, and ML Pack. Each library helps achieve a specific machine learning-related task. For example, OpenCV is an image-manipulating library used for image recognition. Many libraries that enable machine-learning tasks were written for Python.

Python would be used to achieve all the aspects of this study - preprocess dataset; develop models using already existing ICA and GA; infusion of bird approach into the GA; classification using SVM, K-NN, and Ensemble; and the evaluation parameters calculation and generation of results in the form of Confusion matrixes and ROCs. Python will be used

Do Not Copy, Lead City University, Nigeria

for all training, testing, modeling, and development. All these will be done on the Jupiter development environment and the Google Collab Environment.

3.2 Research Design

This research work is to be done experimentally using a quantitative approach. It is to be carried out in five phases. The first phase comprises the design of two dimensionality reduction models, one using Independent Component Analysis (ICA) and another using a novel (improved) Genetic Algorithm (BA-GA). These will be further explained in the following sections: the second phase is the design of a hybrid dimensionality reduction model, combining the already separately designed model (ICA and BA-GA); thirdly, these models are to be tested by classification using support vector machine, k-nearest neighbor and ensemble separately; the fourth phase is to evaluate the classifications done the third phase with some evaluation metrics such as accuracy, precision, and f1-score; finally, the accuracy parameters of these models to be developed from this study would be compared with already existing works.

The section below explains in detail the research methodology: Section 3.3 addresses objective 1, which is the development of DR models for optimization. Section 3.4 investigates the classification methods for evaluating the models developed, which addresses the methods for objective 2. Section 3.5 deals with the evaluation metrics to be used to tackle part of objectives 3 and objective 4.

3.3 Dimensionality Reduction Model Design

This research proposes a hybrid dimensionality reduction strategy for identifying blurry text in natural scene databases. This section gives the method to address objective 1.

Do Not Copy, Lead City University, Nigeria

This Dimensionality reduction (DR) is to be achieved by both feature extraction and feature selection methods. Separately, the feature extraction algorithm uses the ICA learning approach to extract latent components from reduced data to compare the effectiveness of linearity and non-linearity in learning. In contrast, the feature selection algorithm uses an Improved Genetic Algorithm to extract relevant information from high-dimensional data1. The data from these investigations are then used by three distinct classifiers—SVM, K-NN, and Ensemble—to assign labels to previously unknown individuals (test individuals) – which will be explained in the classification section of this chapter.

Using machine learning to improve blurred text primarily aims to anticipate the correct class labels for supplied samples based on the expression profile of those samples. As a result, machine learning works quite well with the hazy text. The dataset is a database of images available online. To facilitate analysis, the raw data is refined. Researchers can utilize these to evaluate or mimic detection methods or develop new ones.

3.3.1 Independent Component Analysis (ICA) for Feature Extraction Phase

To reduce the negative effects of the curse of dimensionality, fresh variables of selected features can be generated by function extraction. Here, the use of Independent Component Analysis (ICA) is proposed, being the first preprocessing stage of this dataset; this would produce the image dataset for the STR images and attempt to extract the blurred images from them. At this stage, it is pertinent to know the justification for using ICA.

Here, using independent components analysis (ICA), this data set with many variables can be condensed into a smaller set of dimensions that can be understood as autonomous functional networks. In addition, the ICA algorithm presupposes a linear mixing mechanism. Scholars have postulated various ICA algorithms, but for this study, Algorithm 3.1 will be used to

Do Not Copy, Lead City University, Nigeria

develop the first model. This algorithm would be inputted into the development environment using the Python language syntax and semantics.

Algorithm 3.1: ICA

1. Centre the data X to make its mean zero, and whiten it to give z.
2. Choose m, the number of independent components to estimate;
3. Choose initial values for the wi, i=1, …, m, and each of the unit norms. Orthogonalize the matrix W (=A−1) as in step 5.
4. For every i=1, …, m, wi←E{zg(wTiz)}−E{g'(wTiz)}w, where g(y) = tanh(ay)(l≤ a ≤2).
5. Do a symmetric orthogonalization of the matrix W = (w1, …, wm), T, by W← (W WT) W.
6. If not converged, go back to step 4.

Independent Component Analysis[4]

3.3.2 Genetic Algorithm for Feature Selection

Feature selection is the process of picking the most relevant features from a pool of potential features that have no known correlations between them. It helps with dimensionality reduction and boosts the reliability and precision of categorization. When anticipating a specific outcome, it pinpoints which factors are most crucial. GA is one of the most recommended and used algorithms for FS because it tends to mimic humans' logic through the scenario of genetics.

In this work, a modified Genetic Algorithm is used to choose features of interest from hazy image datasets for training, skipping any qualities that are not relevant to the task at hand to sharpen latent image features by removing attributes that stretch the image, hence making it blurred out Using a genetic algorithm as a selector function helps get rid of irrelevant features, lowers data dimensionality, and boosts classification precision. Feature selection often

Do Not Copy, Lead City University, Nigeria

involves narrowing a pool of potential candidates to the optimal m attributes out of a possible n. First, the dataset is cleaned up by removing irrelevant features; then, it is adapted to the classifier framework and split into two categories: training and testing.

This research uses an enhanced Genetic Algorithm to locate essential elements for predicting blurred image data but to understand the effects of the improvement done on GA in this study, we look at GA.

For this conventional GA algorithm (Algorithm 3.2) - Python language allows it to be keyed into the development environment, executed, and modified as one deems fit while able to see the effect the modification brings. This GA would be modified as it is being executed so that differences can be noted and monitored.

Algorithm 3.2: GA

Initiate: Set nPop = m, tmax, t = 0;

Confirm the Optimal feature subset with the highest suitable rate.

1: where (t = tmax) do

2: Make pop a, tmax;

3: Do for k = 1 to a

4: [a1, a2] parents = system selection (a, nPop)

5: Xor[a1, a2] = Child

6: M u = [Child] mutation

7: Finish for

8: Replace a with Child1, Child2, ..., Childm. t = t + 1;

10: Remember to save the highest fitness value chromium = certain or non-certain feature through the threshold, set value = 0.5, and = the number of selected features where a = population size, r = random number between 0 and 1 and = the number of features chosen.

Genetic Algorithm[5]

Do Not Copy, Lead City University, Nigeria

3.3.2.1 Improved Genetic Algorithm (Bird Approach – Genetic Algorithm)

The primary challenge of the GA method is picking optimal features from inherently predictable data. In order to ensure that the best feature needed to predict the blurred texts in the dataset, a Bird Approach was incorporated into the GA.

To further improve the GA's already impressive structure, scholars have always recommended fusing it with the smooth behavior typical of GAs. For instance, in the case of this study, If a bird needs to find food, it sees it clearly at a distance. It goes for it irrespective of how minute it might be to the human eyes, or if it discovers an object for any other reason, it can do so easily and quickly because of its keen sense of sight. Once the birds have arrived at the feeding field, they will continue their search. Hunters snip and venture forth to discover new food sources. Following are examples of standard behavior for both producers and freeloaders:

$$xi, jt + 1 = xi, jt + randn(0,1) \times xi, jt$$

$$xi, jt + 1 = xi, jt + (xk, jt - xi, jt) \times FL \times rand(0,1)$$

If each bird flies to a new site in the unit interval simplifies the pseudocode for the I-GA method. GA's attempt to find a happy medium between exploitation and exploration always leads to its being trapped within a narrow rut. This shortcoming is remedied in the proposed Improved-GA (BA-GA) by employing four strategies to boost the algorithm's global searching and local searching capacities: (1) the control randomization (CR) parameter and (2) an advanced nonlinear transfer function to balance exploration and exploitation; (3) varying parameters in the GA exploration phase; and (4) a novel local updating position strategy based on the GA algorithm.

The essence of the bird approach infused into GA is to give the outcome of the hybrid model more precision as compared to what conventional GA would give. In pattern recognition, the

Do Not Copy, Lead City University, Nigeria

more precise the result generated, the more accurate the model is said to be. Again, Algorithm 3.3 is derived from the adjustment done to the GA.

Algorithm 3.3: Improved GA (BA-GA)

Load the Dataset (Sample Population)

N: the total number of images from which this representation is drawn

Start a population off on the path to improved fitness through mutation and crossover.

Algorithm iteration limit, denoted by the letter M

Q: how much dimensions can be done away with?

Introducing the Forager's Success Rate (P) in Searching for Food

Set forth the following five constants: The population size and other important parameters can be set to zero by entering the values S, C, FL, a1, and a2 at time t = 0.

Calculate a reliable approximation of the fitness of N images.

If (t > M), then

If (t% FQ ≠ 0)

For I = 1:N

When rand P

For dimension to be reduced, use

$xi,jt + 1 = xi,jt + (pi,j - xi,jt) \times C \times rand(0,1) + (gj - xi,jt) \times S \times rand(0,1)$

Else

$xi,jt + 1 = xi,jt + A1(meanj - xi,jt) \times rand(0,1) + A2(pk,j - xi,jt) \times rand(-1,1)$ *should be used for dimensions to be retained.*

Stop if Stop for All Else

Introduce Connected Components: Split into two groups, the CC consists of: For i=1:N If foreground region, then The term "generating" is calculated using $xi,jt+1=xi,jt+randn(0,1)\times xi,jt$

Do Not Copy, Lead City University, Nigeria

If not, "scrounging" is calculated using $x_{i,jt+1}=x_{i,jt}+(x_{k,jt}-x_{i,jt})\times FL\times rand(0,1)$.

It's time to call it quits if and only if Find alternative ways to handle the problem The old methods should be abandoned if the new ones prove to be more effective. Find the greatest option for the long term. End while $t = t + 1$

The best foreground region in the population is the result.

Output global best solution

End

Improved GA (BA-GA) (Researcher, Nwufoh C.V:2023)

The following steps further explain Algorithm 3.3.

1. Initialize a population of candidate image processing algorithms randomly. Each candidate algorithm represents a potential solution for detecting blurred text in images.
2. Evaluate the fitness of each candidate in the population. The fitness function should measure how well each algorithm detects blurred text. For example, you can evaluate the algorithm's ability to distinguish between blurred and sharp regions in an image containing text.
3. Select a subset of the candidate's algorithm for the next generation based on their fitness. They used selection techniques like tournament selection or fitness-proportionate selection to favor algorithms with higher fitness.
4. Create new candidate algorithms through crossover and mutations. Here is an example approach:
    1. Define a search radius for each candidate algorithm. This radius represents the exploration range similar to how birds scout for food in a specific area.
    2. For each candidate algorithm, generate a new algorithm by performing a crossover operation with another candidate algorithm within the search radius. The crossover operation should combine the key components or parameters of the parent algorithms to create a new offspring.

Do Not Copy, Lead City University, Nigeria

3. Apply mutation to the new algorithms. Mutation introduces small random changes to the algorithm's components or parameters. In blurred text detection, mutations that simulate adjustments to the algorithm's image processing are considered, such as edge detection or blurriness metrics.

5. Evaluate the fitness of the new candidate algorithms.
6. Repeat steps 3 to 5 for a certain number of generations or until a satisfactory algorithm is found.

By applying the principles of genetic algorithms and incorporating the notion of bird-like precision in the search process, the adapted algorithm aims to guide the population toward an improved algorithm for detecting blurred text in images.

3.4 Classification of DR models

In Machine Learning and, in fact, any field of research, once a model is developed, it is expected that it should be tested to ascertain if the model gives the outcome expected or if it gives otherwise. For this study, to test the outcome of the various instances of the models, some algorithms would be used as classifiers; this phenomenon is called Classification. These algorithms, like every other algorithm used in the development environment, are converted into Python programming language, and executed. The classifiers for this study are Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Ensemble. These classifiers are used separately to test each model's design. This section explains using SVM, KNN, and Ensemble as classifiers.

3.4.1 Support Vector Machine (SVM) for Classification

SVM is a machine learning algorithm widely used in diverse fields, such as data analysis and pattern recognition. SVM is a very effective supervised learning method for Classification and regression. It optimizes model complexity and learning ability based on small amounts of

Do Not Copy, Lead City University, Nigeria

training data. The SVM is constructed from the best classification surface when linear separability holds. The ideal classification surface not only reliably divides data into two groups (training and test) but also widens the margin of error between the two groups.

Various instantiations of SVM algorithms have been developed over the years, but for the case of this study, Algorithm 3.4 will be used. This would be done by converting it into Python programming language, and its input values would be the outcome of the models.

Algorithm 3.4: Support Vector Machine Pseudocode

---

1. Normalise the dataset
2. For each C, ϒ:
   - 2.1 Cross Evaluation using leave-one-out.
     - 2.1.1 Train and test and SVM
     - 2.1.2 Store the success rate.
   - 2.2 Compute the average success rate
   - 2.3 Update the best C and ϒ if needed.
   - 2.4 Return to 2.1 with next C, ϒ.
3. Choose C, ϒ with best average success rate, and perform step (2) using fine scale around the selected parameters.

Where C = categorized sample

ϒ = weighted vector values

Support Vector Machine[3]

3.4.2 K-Nearest Neighbor (KNN) for Classification

Regarding classification methods, K-Nearest Neighbor is the simplest in conceptual and computational complexity while delivering excellent classification accuracy. In k-Nearest Neighbors, the Euclidean distance is used as the distance function, and a voting function

Do Not Copy, Lead City University, Nigeria

forms the basis of the K-NN method. Despite being slower than other standard statistical classifiers, the K-NN has better data accuracy and stability. However, now that we have so much computing power, there will be no need to worry about lousy run-time performance. The k-nearest-neighbor classifier is a standard nonparametric supervised classifier that has been shown to function effectively at optimal k values. Training and testing phases are included in the K-NN method, just as they are in most directed learning algorithms. During the training phase, data points are displayed in an n-dimensional space. These training data points are tagged with labels that indicate their respective categories.

The K-NN algorithm consists of the following steps:

i. First, settle on a reliable distance measurement system.
ii. Second, all set P of training data is stored during training in pairs (based on the features being used) I = 1 and P = 0. (yi; ci). Where yi is a training pattern from the training data set, ci is the class it belongs to, and n is the total number of patterns in the training set.
iii. Third, the testing phase involves calculating the gaps between the newly introduced feature vector and the existing feature set (training data).
iv. Fourth, the k nearest neighbours are selected and polled on the new example's Classification. Accurate classifications made during testing are utilized to gauge the algorithm's efficacy. If this degree of accuracy is unsatisfactory, the k value can be tweaked to produce better results.

Algorithm 3.5 is the KNN algorithm for another set of model classifications in this study. The input data for this algorithm is the outcome from the models developed.

Do Not Copy, Lead City University, Nigeria

Algorithm 3.5: K-Nearest Neighbour KNN Pseudocode

Input: the training set D, test object x, category label set C

Output: the category c_x of test object x, c_x belong to the C

1. **begin**
2. **for** each y belongs to D do
3. Calculate the distance D(y, x) between y and x
4. **end for**
5. select the subset N from the data set D,
   the N contains k training samples which are the k
   nearest neighbors of the test sample x
6. calculate the category of x:

$$c_x = \arg\max \sum_{c \in C} \sum_{y \in N} I(c = class(y))$$

7. **end**

K-Nearest Neighbour[4]

3.4.3 Ensemble for Classification

When a group of classifiers is used to improve classification results, it is called an ensemble of classifiers. An ensemble of classifiers improves precision over a single classifier. Any machine learning algorithm relies on inputs consisting of features of a pre-trained network. To make deep learning models more precise, alternative classifiers such as the support vector machine, logistic regression, K-nearest neighbors, and decision trees can supplement the results. The primary purpose of the classifiers is to assign an appropriate category to an image using the features that have been learned. Results from a classification task are made more reliable and accurate by using an ensemble of classifiers. Voting, averaging, and weighted averaging are the three broad categories into which ensemble techniques fall.

Do Not Copy, Lead City University, Nigeria

Algorithm 3.6: Ensemble

1. **Input:** training data D = $\{x_i , y\}^{m}{}_{i=1}$
2. **Output**: ensemble classifier H
3. *Step 1*: learn base-level classifiers
4. **for** t = 1 to T do
5. learn h_t based on D
6. **end for**
7. *Step 2:* construct new data set of predictions
8. **for** i = 1 to m **do**
9. Dh = $\{x'_i, y_i\}$ where x'_i = { h1 (x_i), ……., h_T (x_i)}
10. **end for**
11. *Step3:* learn a meta-classifier
12. learn H base on D_h
13. return H

Ensemble[5]

3.5 Performance Evaluation

Different performance measures, such as precision, recall, f-measure, accuracy, specificity, and so on, are used to evaluate a Machine Learning system's classification performance. Dissimilar categorization models are often evaluated using these metrics.

The percentage of correct diagnoses is determined by dividing the number of positive results by the sum of all positive and incorrect results. Some of the performance metrics that would be employed for this study would be the following:

Do Not Copy, Lead City University, Nigeria

1. **Accuracy**: is defined as the percentage of positive observations that were accurately predicted relative to the total number of positive observations. Accuracy in mathematical expression is defined **(TP + TN)/ (TP + TN + FP + FN)**. – Equation (1) Where TP, FN, FP, and TN represent the number of true positives, false negatives, false positives, and true negatives, respectively. This helps to ascertain how correct the models to be created will be, and because most studies tend to find the accuracy of their models, this would form a basis for comparing the proposed model and the state of the art. The most natural performance metric in chronic disease diagnosis is accuracy, the percentage of correct predictions relative to the total number of observations.
2. **Precision** is a measure of how much detailed information is given. It measures the degree to which exactness is applied in the model developed. Hence, the smaller the unit used, the more exact the result. Precision in numbers is the total number of significant decimal or other digits. It relates to accuracy in the following ways: low accuracy gives high precision. Calculating TP (True Positive) as a function of TP plus FP (False Positive) yields a measure of precision **TP/TP + FN.** It is quantitatively demonstrated in equation form as the ratio of genuine positives to the combined number of false negatives and positives.
3. **F1 Score** is a machine learning evaluation metric that combines precision and recall scores. The accuracy metric computes how often a model made a correct prediction across the entire dataset. This can be a reliable metric only if the dataset is class-balanced; that is, each dataset class has the same number of samples. Nevertheless, real-world datasets are heavily class-imbalanced, often making this metric unviable. This is where the F1 score comes in. Precision measures how many of the "positive" predictions made by the model were correct. Recall measures how many of the positive class samples present in the dataset were correctly identified by the model.

Do Not Copy, Lead City University, Nigeria

The F1 score combines precision and recall using their harmonic mean, and maximizing the F1 score implies simultaneously maximizing both precision and recall. Thus, the F1 score has become the choice of researchers for evaluating their models in conjunction with accuracy, using the formula 2TP / 2TP + FP + FN = F.

3.5.1 Tools for Illustrating Performance Metrics

These performance metrics will be modeled with Python and be illustrated using the Confusion matrix and ROC graph:

1. **Confusion Matrix:** A **Confusion matrix** is an ***N x N matrix*** used for evaluating the **performance of a classification model,** where **N** is the number of ***target classes***. The matrix compares the actual target values with those predicted by the machine learning model. It is a class-wise distribution of the predictive performance of a classification model—that is, the confusion matrix is an organized way of mapping the predictions to the original classes to which the data belong. This is used mainly for supervised learning frameworks. A confusion matrix can be computed for the same test set of a dataset, but using different classifiers can also help compare their relative strengths and weaknesses and draw an inference about how they can be combined (ensemble learning) to obtain the optimal performance.

Do Not Copy, Lead City University, Nigeria

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Figure 3.1: Confusion Matrix

Do Not Copy, Lead City University,

Confusion matrix deals with four (4) parameters:

a. *True Positive (TP)* refers to a sample of the positive class being classified correctly. When the actual value is Positive and predicted, it is also Positive.

b. *True Negative (TN)* refers to a sample of the negative class being classified correctly. When the actual value is Negative, and the prediction is also Negative.

c. *False Positive (FP)* refers to a sample belonging to the negative class but being classified wrongly as belonging to the positive class. When the actual is negative, but the prediction is Positive. Also known as the **Type 1 error**

d. *False Negative (FN)* refers to a sample belonging to the positive class but being classified wrongly as belonging to the negative class. When the actual is Positive, but the prediction is Negative. Also known as the **Type 2 error**

2. **Receiver Operating Characteristics**: A Receiver Operating Characteristics (ROC) curve is a plot of the "true positive rate" concerning the "false positive rate" at

different threshold settings. ROC curves are usually defined for a binary classification model, although that can be extended to a multi-class. We use a threshold to interpret ROC curves. Different thresholds represent the different possible classification boundaries of a model. The RIGHT side of the decision boundary depicts the positive class, and the LEFT side depicts the negative class**.**

The more the curve is close to the top left corner of the plot, the better the classifier is at distinguishing the categories. The area under the curve (AUC) measures the accuracy of the classifier, with higher values indicating better performance.

3.6 Proposed Model and Algorithm

Having explained and described all the concepts in this study, this section presents a workable workflow for the study and its algorithm (pseudocode).

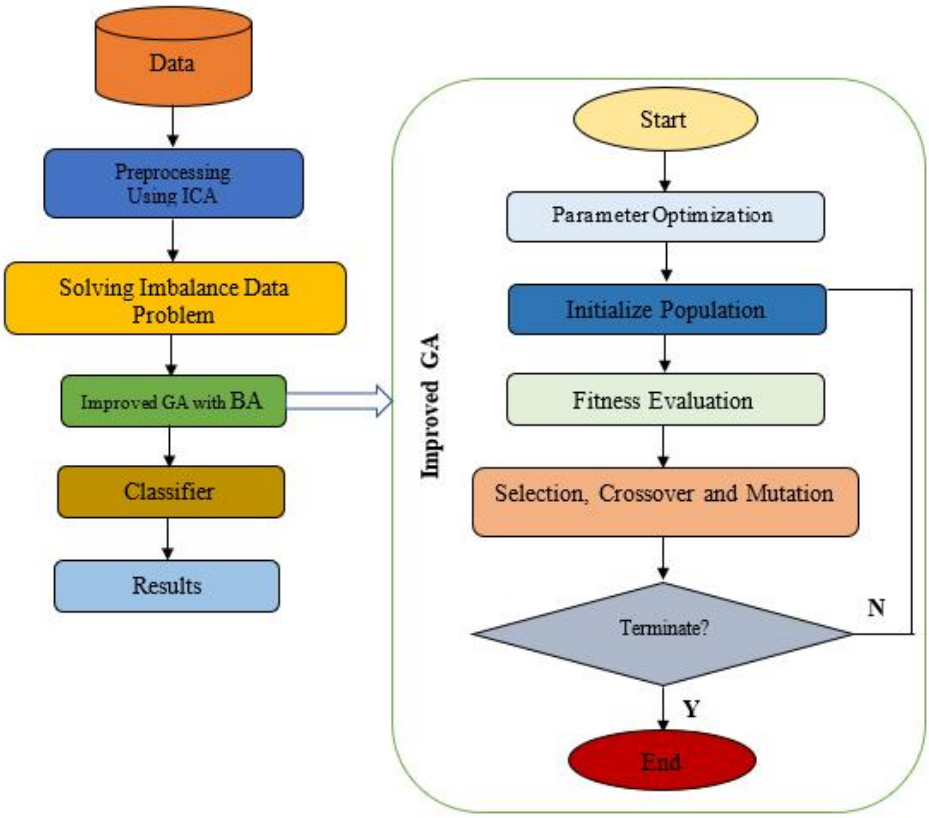


Figure 3.2: Workflow Model (Researcher, Nwufoh C.V: 2023)

University, Nigeria

Algorithm 3.7: Workflow Algorithm

Initialization: Load Dataset

1. Perform ICA
2. mutation_rate = 0.1 //Mutation rate for GA
3. min_mutation_momentum = 0.0001 //Min mutation momentum
4. max_mutation_momentum = 0.1 //Max mutation momentum
5. min_population = 5 //Min population for BA-GA
6. max_population = 10 //Max population for BA-GA
7. num_Iterations = 10 //Number of iterations to evaluate BA-GA
8. Input:
9. Training Set, Evaluation Set
10. Begin
11. num_population = random.randint (min_population, max_population); // Generate initial population for Classification
12. population_Classification = []
13. For i in range (num_population):
14. Classification_parameters = random.randint (min_num_estimators, max_num_estimators) // Classification parameters generation
15. Classification_ model = generate_ Classification (Classification _parameters)
16. population_ Classification.append (Classification _ model)
17. End for
18. max_accuracy = 0
19. best_model = None
20. population_evaluation_accuracy= [[]
21. For i in range (num_Iterations):
22. For j in range (num_population):
23. Classification _model = population_ Classification [j] // population selection // population evaluation

Do Not Copy, Lead City University, Nigeria

24. evaluation_accuracy = evaluate_ Classification (Classification _model, Training_Set, Evaluation_Set)
25. population_evaluation_accuracy.append (evaluation_accuracy)
26. If evaluation_accuracy > max_accuracy:
27. max_accuracy = evaluation_accuracy
28. best_model = Classification _model
29. End if
30. End for
31. // Create a new population with new generations
32. # every generation will use the current best GXGBoost_model to mate
33. For pop_index in range (num_population):
34. model1 = population_GXGBoost [pop_index]
35. model1_evaluation_accuracy = population_evaluation_accuracy [pop_index]
36. model2 = best_model
37. model2_evaluation_accuracy= max_accuracy
38. // Create new generation with crossover
39. new_model = crossover_ Classification (model1, model1_evaluation_accuracy, model2, model2_evaluation_accuracy)
40. mutate_ Classification (new_model) // Mutate new generation
41. population_Classification [pop_index] = new_model // Replace current model
42. End for
43. End for
44. Return best_model, max_accuracy
45. End.

Workflow Algorithm (Researcher, Nwufoh C.V:2023)

Do Not Copy, Lead City University, Nigeria

Endnotes

W. Jia, M. Sun, J. Lian & S. Hou. "*Feature Dimensionality Reduction: A Review.*" **Complex and Intelligent Systems 8**, no. 3 January 2022: 2663–2693.

[2]Y. Sun et al, Z. Ni, C-K Chng, Y. Liu, C. Luo, C. C Ng, J. Han, E. Ding, J. Liu, D. Karatzas, C. S Chan & L. Jin "*ICDAR 2019 Competition on Large-Scale Street View Text with Partial Labeling-RRC-LSVT*," **Proceedings of the International Conference on Document Analysis and Recognition, ICDAR** 2019: 1557–1562.

[3]D. Cao, Y. Zhong, L. Wang, Y. He & J. Dang "*Scene Text Detection in Natural Images: A Review,*" **Symmetry 12**, no. 12 November 2020: 1–26, https://www.mdpi.com/2073-8994/12/12/1956.

[4]D. K Naik, G. R & Kumar, "*An Oversview of Independent Component Analysis and Its Applications,*" **Informatica (Ljubljana) 35,** no. 1 2011: 63–81.

[5]M. M. Mijwel, "*Genetic Algorithm Optimization by Natural Selection,*" **Computer Science,College of Science 1**, no. 1 2016: 1–6.

Do Not Copy, Lead City University, Nigeria

Chapter Four

Results and Discussion

4.1 Introduction

The results and assessment of the study's implementation are shown below. Furthermore, it provides a concise overview of the research behind it and the implications of the suggested model. The assessment results support the study's need to be carried out by its aim and objectives. This section shows the results obtained from the methods used to solve the objectives of this study. Each section shows the result, and a discussion of that result is explained immediately. All the algorithms (DR model algorithms, classification algorithms) and the formulas of the evaluation for parameters are all modeled in the Jupyterlab ecosystem using the Python language, and once the dataset is inputted, the various models are executed sequentially, and various results are generated.

Section 4.2 gives the result and discussions of the outcome that addresses the objective i. Section 4.3 delivers and discusses the results obtained from the classification of the models. Section 4.4 now gives detailed values of the evaluation parameters used in the model's classification. These are achieved using the acceptable formulas for each of these parameters. Section 4.5 gives a comparison of results achieved with already existing works.

4.2 Dimensionality Reduction (DR) Models

This section shows the result of objective 1 but in two phases, split into two sections (4.2.1) and (4.2.2). All these models are developed using the Python language.

Do Not Copy, Lead City University, Nigeria

4.2.1 Dimensionality Reduction (DR) Model using ICA

```
np.random.seed(0)
n_samples = 2000
time = np.linspace(0, 8, n_samples)

s1 = np.sin(2 * time)  # Signal 1 : sinusoidal signal
s2 = np.sign(np.sin(3 * time))  # Signal 2 : square signal
s3 = signal.sawtooth(2 * np.pi * time)  # Signal 3: saw tooth signal

S = np.c_[s1, s2, s3]
S += 0.2 * np.random.normal(size=S.shape)  # Add noise

S /= S.std(axis=0)  # Standardize data
# Mix data
A = np.array([[1, 1, 1], [0.5, 2, 1.0], [1.5, 1.0, 2.0]])  # Mixing matrix
X = np.dot(S, A.T)  # Generate observations

# Compute ICA
ica = FastICA(n_components=3)
S_ = ica.fit_transform(X)  # Reconstruct signals
A_ = ica.mixing_  # Get estimated mixing matrix

# We can `prove` that the ICA model applies by reverting the unmixing.
assert np.allclose(X, np.dot(S_, A_.T) + ica.mean_)
```

Figure 4.1: ICA DR Model Snippet (Researcher, Nwufoh C.V: 2023)

Do Not Copy, Lead

The already existing ICA algorithm is converted into Python language with other Python libraries, as depicted in the model snippet in Figure 4.1. The Jupiter Development Environment has Jupyterlab ecosystem's dependencies, and libraries were set up via the "!pip install" command. PyTorch, NumPy, Pandas, Matplotlib, Sklearn, Shutil, and Pillow are just some of the libraries that were brought in. The ICA is used for the model, and the dataset is loaded onto it. The ICDAR2019 dataset was used for this analysis and is freely available to the public. The dataset loaded comes in .CSV file format and is used for both the testing and training of the model. A cross-section of the CSV exports of the loaded images is displayed in Figure

4.2

0	0.652988	-0.478481	-39.062063	0.098293	1.126915	17.370630	2.473360	-1.364710	1.131468	-0.002516	1.280221
1	0.680325	-0.520884	0.000047	0.104437	1.184565	16.914978	2.792200	-1.306096	1.201208	2201.734642	1.442902
2	0.654511	-0.521990	-78.124979	0.104355	0.918378	15.918480	2.787837	-1.253875	1.176501	-0.001336	1.384155
3	0.634057	-0.541421	-273.437461	0.103709	0.991626	16.377148	2.753444	-1.171099	1.175478	-0.000379	1.381749
4	0.606181	-0.552172	39.062362	0.105350	0.845794	15.462336	2.841272	-1.097813	1.158352	0.002697	1.341780
...	...	...	...	...	...	...	...	...	...	...	...
17400	0.037540	-0.039791	-390.624819	0.015672	-0.005402	-0.671443	0.062877	-0.943438	0.077331	-0.000040	0.005960
17401	0.019695	-0.022420	78.124925	0.009262	-0.165168	-0.691274	0.021962	-0.878484	0.042115	0.000119	0.001774
17402	0.029204	-0.025100	234.374859	0.010622	-0.172708	-0.389803	0.028885	-1.163476	0.054304	0.000045	0.002949
17403	0.028539	-0.025256	78.124818	0.011683	0.165351	-0.753699	0.034942	-1.129990	0.053795	0.000150	0.002894
17404	0.021136	-0.041336	78.125073	0.011684	-1.056987	0.928356	0.034949	-0.511314	0.062472	0.000150	0.003903

Figure 4.2: Loaded Data on the Python Environment (Researcher, Nwufoh C.V: 2023)

The dataset was culled from the Mendeley repository, and a separate directory for the evaluation and training sets was made. Each matrix value is graphically represented by a unique hue in the heatmap depicted in Figure 4.2

Further discussing this model: This original photo collection is converted to a CSV file that looks like a matrix of images made by superimposing natural images over text displayed with arbitrary fonts, sizes, colors, and orientations. Because the overlay is based on carefully organized combinations and a well-defined learning process, these statements sound natural. The training and test images for ICDAR2019 were shot using low-resolution wearable cameras and numbered in the thousands. Multiple sentences in different orientations are marked in each image using the four corners of a square. A screenshot of the development environment showing the interactive computation and DR of the loaded dataset using ICA is shown in Figure 4.3. Figure 4.3 shows a further reduction and processing of the dataset using the ICA model designed in the Python development environment. Figure 4.3 shows a cross-section of the deblurred scenic images in the dataset.

Do Not Copy, Lead City University, Nige

Figure 4.3: Initial Dimensionality Reduction (Researcher, Nwufoh C.V: 2023)

In summary, Figure 4.3 shows a snippet for the cross-section of the outcome of the first dimensionality reduction (DR) model developed using ICA. Here, features were extracted from the dataset using the ICA model, giving us a nearly readable image, though still unclear.

4.2.2 DR using BA-GA with ICA

```
# Initialize the population
    population = [random.choice(search_space) for _ in range(population_size)]

    # Evolve the population
    for i in range(num_generations):
        # Evaluate the fitness of each individual
        fitness_values = [fitness_function(p, img) for p in population]

        # Select the fittest individuals for breeding
        parents = [population[i] for i in np.argsort(fitness_values)[-2:]]

        # Generate offspring by crossover and mutation
        offspring = []
        for i in range(population_size - len(parents)):
            parent1, parent2 = random.sample(parents, 2)
            offspring.append((
                random.choice([parent1[0], parent2[0]]),
                random.choice([parent1[1], parent2[1]])
            ))
            if random.random() < 0.1:
                offspring[-1] = (
                    random.choice(thresholds),
                    random.choice(kernel_sizes)
                )
```

Figure 4.4: BA-GA DR Model Snippet (Researcher, Nwufoh C.V: 2023)

d City University, Nigeria

Many other experiments were suggested in this study to be performed on the loaded dataset to extract and select other features that are needed to give us more explicit images. These relevant and latent features in the data are retrieved with the help of a Bird Approach - Genetic algorithm (BA-GA) and Independent Component Analysis (ICA) in this implementation. This hybridization is done sequentially and not concurrently. Figure 4.4 shows a snippet of the improved GA (BA- GA) model introduced to the ICA DR model sequentially in order to extract features from the dataset further to have a sharper image collection, as can be seen in Figure 4.5. Figure 4.5 is the outcome of, first and foremost, developing a model using GA, which was modified using a Bird Approach (BA). After the model was designed using already existing GA algorithms infused with the novel Bird Approach, the outcome of the dataset generated from the first DR model with ICA (Figure 4.3) is passed as an input into this BA-GA model, which yields a sharply deblurred image dataset as seen in Figure 4.5.

Lead City University, Nigeria

Figure 4.5: Output of some of the Blur Detections shown (Researcher, Nwufoh C.V: 2023)

In summary, Figure 4.5 shows the outcome of the same snippet of images shown In Figure 4.3. This model is a hybridized DR model using BA-GA and ICA algorithms. BA-GA is an enhanced GA, which then is combined with ICA by passing the outcome of DR using ICA into DR using BA-GA. They clearly show that with our hybridized model, we get a better text deblurring model.

4.3 Classification of Developed Models

At this point, the models developed are tested using three distinctive algorithms: SVM, K-NN, and Ensemble. The choice of using more than one classifier is to give room for comparison; strength and weakness evaluations of the models, and to make inferences. All these algorithms are great on their own, so it is not a function of comparison of the algorithms but of the performance of the models via those classification algorithms. The models are run through these classifiers separately, and the result for each is presented using the Confusion Matrix (which gives us a basis for proper comparison – favorable and Negative). Also, ROC was used to depict further the results obtained because it gives a probabilistic outcome of the performance of the individual models developed. One way to visualize how well a classification model performs across different cut-off points is through a ROC curve. We also used Confusion Matrixes. Other scholars can choose other tools to use for classification and evaluation. A True Positive Rate versus False Positive Rate curve is shown in the ROCs. Results achieved with the DR model using ICA and the DR model developed using BA-ICA are classified utilizing SVM, KNN, and Ensemble, shown in this section. This section presents the results that address the objective 3.

4.3.1 SVM, K-NN, and Ensemble Classifications of ICA DR Model.

To show or represent the balanced nature of the model developed on the imbalanced nature of the data set, we used the ROC curve and the Confusion Matrix. It will be observed that under this section, there are six (6) figures, three (3) confusion matrixes, and three (3) ROCs. The

Do Not Copy, Lead City University, Nigeria

reason for this is that each of the classifiers is used separately to test the first DR model created in objective 1. So, for each of the classifiers that were used to test the ICA DR model, the Jupyterlab ecosystem would produce its confusion matrix and ROC (that are the tools we used for comparison and evaluation of the model in this study). For example, when the ICA DR Model is classified using SVM, the system produces the ICA-SVM confusion matrix and ICA-SVM ROC. The same applies to the other two (2) classifiers.

Figure 4.6a is a confusion matrix showing the predictive positive and negative outcomes of classifying the ICA DR model with SVM. There are four (4) partitions with four (4) values – true positive (TP), true negative (TN), False Positive (FP), and False Negative (FN). The top right portion shows the True Positive outcome of the classification, trying to juxtapose the actual outcome of the model with the predicted outcome of the model. Here, it is seen that the value for TP > TN and FP> FN; because TP = 142 and it is more significant than all the other scores, it is safe to say that the performance of this model using SVM as the classifier is satisfactory.

In Figure 4.6b ROC or decision curve is presented for the same instantiation of the classification – which is ICA-SVM. The ROC curve makes a binary prediction of the four outcomes that the confusion matrix above gave. When SVM is used to classify the ICA DR Model in objective 1, it is noticed that the graph is plotted as a True Positive Rate (Y-axis) against a False Positive Rate (X-axis)– it only evaluates the two (2) positive outcomes of the evaluation process. Figure 4.6b presents the ROC curve of the predictor; that is the blue line, and we can see that the AUC of that curve is close to the control (gray), that is, 1 and the nearer the ROC curve is closer to the value 1 at the Y axis the good the model is said to be.

Do Not Copy, Lead City University, Nigeria

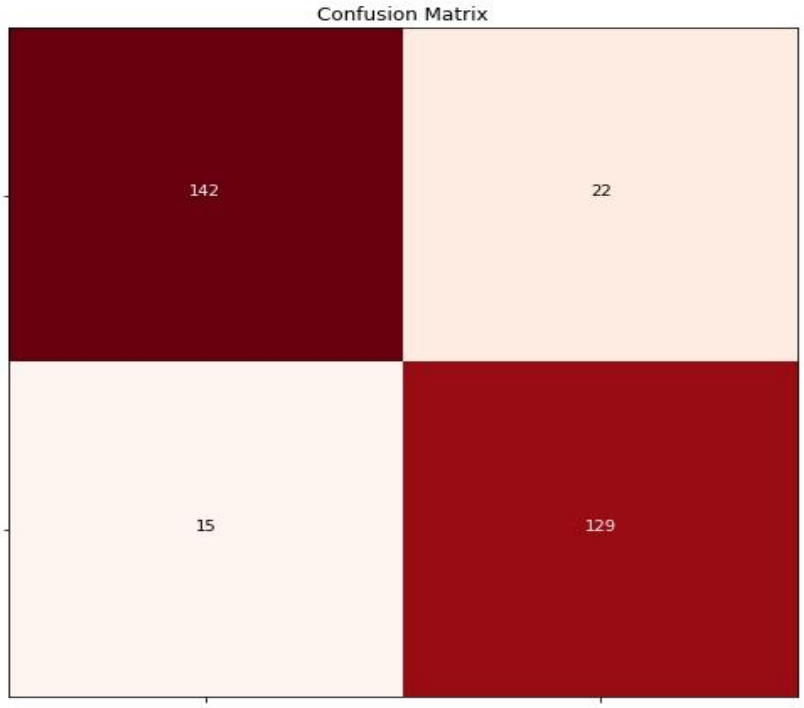


Figure 4.6a: Confusion Matrix Showing the Dataset with ICA and SVM (TP=142; TN=129; FP=22; FN=15) (Researcher, Nwufoh C.V: 2023)

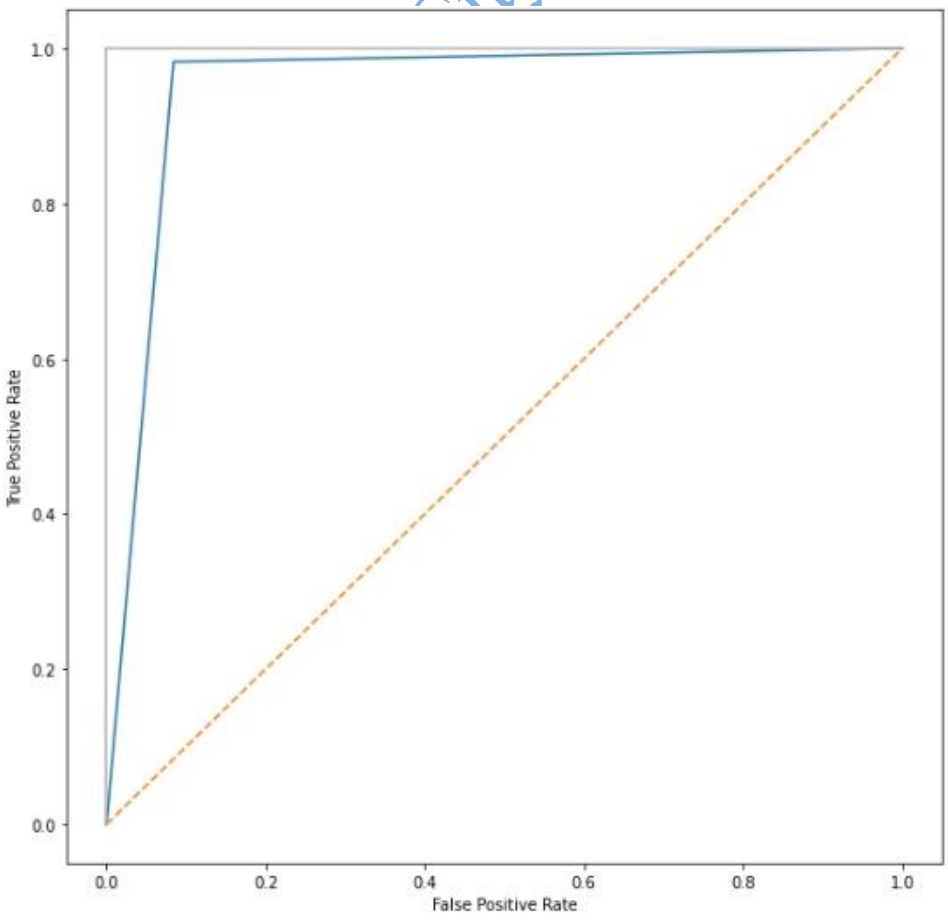


Figure 4.6b: ROC Curve for ICA-SVM (Researcher, Nwufoh C.V: 2023)

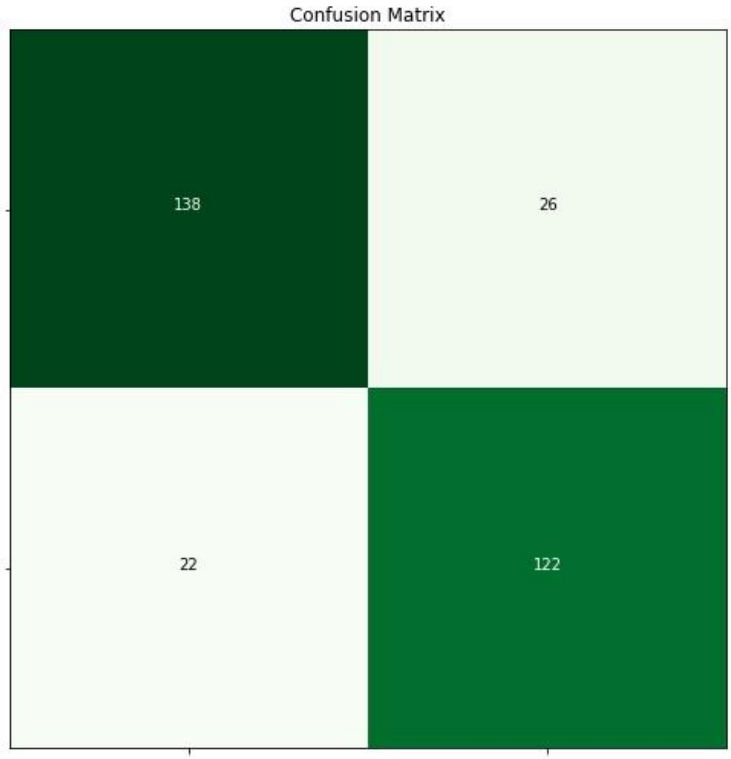


Figure 4.7a: Confusion Matrix Showing the Dataset with ICA and KNN (TP=138; TN=122; FP=26; FN=22) (Researcher, Nwufoh C.V: 2023)

Do Not Copy, Lead City

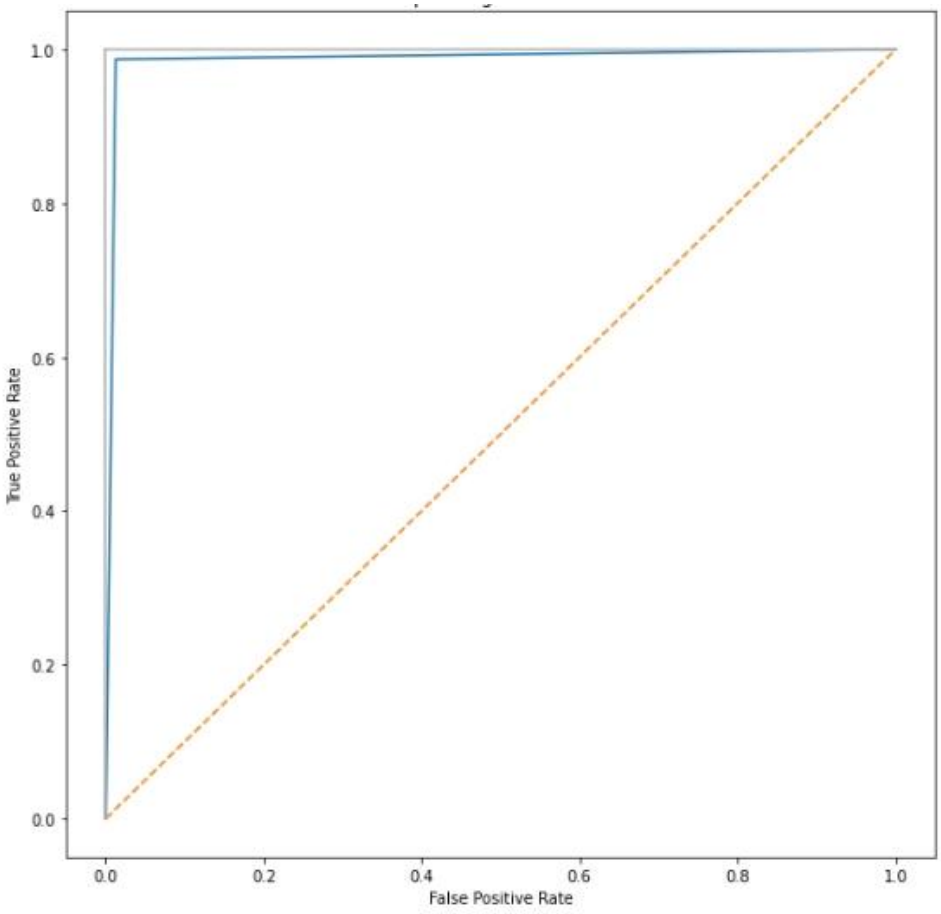
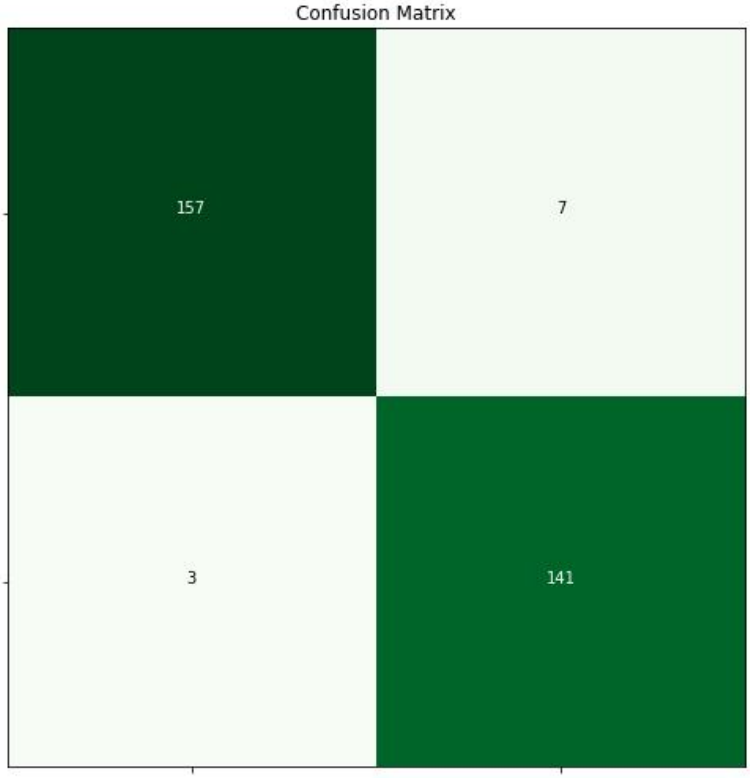


Figure 4.7b: ROC Curve for ICA-KNN (Researcher, Nwufoh C.V: 2023)



Do Not

Figure 4.8a: Confusion Matrix Showing the Dataset with ICA and Ensemble (TP=157; TN=141; FP=7; FN=3) (Researcher, Nwufoh C.V: 2023)

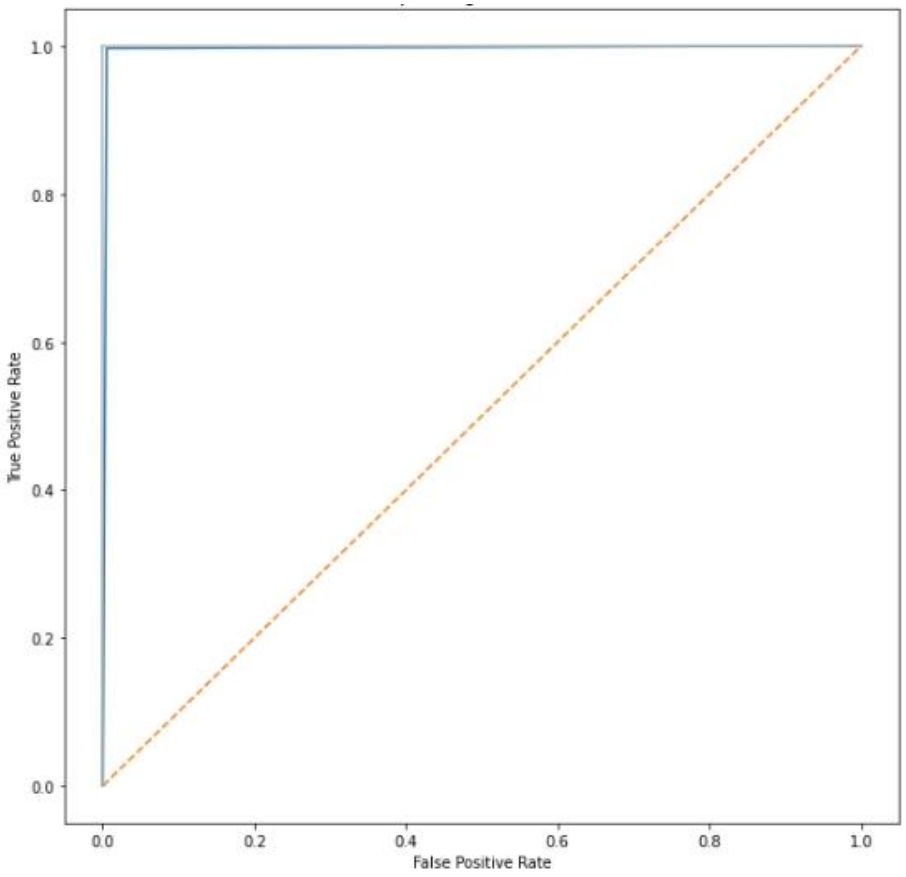


Figure 4.8b: ROC Curve for ICA-Ensemble (Researcher, Nwufoh C.V: 2023)

General Discussion of Results in 4.3.1: Just as is the case for Figure 4.6a, Figure 4.7a, and Figure 4.8a both show that even with different classifiers, KNN and Ensemble – the TP value is greater than the other three values (TN, FP, and FN). This suggests that these algorithms, on their own, are all top performers. However, in comparison to the Confusion Matrixes, it is observed that the classification of the ICA DR Model using Ensemble (Ensemble-ICA in figure 4.8a) has TP = 157, which is greater than ICA-SVM than ICA-KNN follows.

In this study, the confusion matrix represents the outcomes of the classification of the models using two (2) algorithms, SVM and KNN. Here, the discourse will also be presented concerning the values of TP (True Positive), TN (True Negative), FN (False Negative), and

Do Not Copy, Le

FP (False Positive). The confusion matrix for the I-GA with SVM and K-NN are shown in figure 4.7 and 4.8, with the values as follows: I-GA-SVM (TP=3202; TN= 3523; FP=296; FN=537) and I-GA-K-NN (TP= 3420; TN=3965; FP=78; FN=95).

As can be seen in Figure 4.6b, Figure 4.7b, and Figure 4.8b - for the ROC curves for all instances of the classifiers on ICA, it would be observed that the AUCs (Area Under the Curve) tend close to a TP of probability of 1. In ROC, the closer a result is to 1, the better the model. For SVM, K-NN, and Ensemble, we observe that the AUCs tend to be 1, which implies that the model works great, but it is closest to Ensemble (Figure 4.8b). In comparison, the outcome of ROC for ICA-Ensemble is said to be a better model, followed by ICA-KNN and then ICA-SVM because of the degree of nearness of the curve to 1.

4.3.2 SVM, K-NN and Ensemble Classification on BA-GA DR Model

This section presents a novel, improved genetic algorithm by enhancing the crossover and mutation operations of a simple genetic algorithm that combines a replacement operation to preserve population diversity by regularly performing a local initialization operation. The results are classified using SVM, KNN, and Ensemble, and the results are also shown in two forms – confusion matrixes and ROC curve as is seen in Figures 4.9 a, 4.9b, 4.10a, 4.10b. 4.11a and 4.11b. Again, Figures 4.9a, 4.10a, and 4.11a show the outcomes for BA-GA with SVM, KNN, and Ensemble, respectively, and it is observed that all their TPs > TNs, they are all said to have performed excellently in their various capacities.

Also, in the case of the ROCs in figures 4.9b, 4.10b, and 4.11b, it is observed that the ROCs curves (blue line) for the outcome of the three (3) classifications of BA-GA (BA-GA-SVM, BA-GA-KNN, and BA-GA-Ensemble) all tend towards the TP of probability 1. Again, the

Do Not Copy, Lead City University, Nigeria

nearer the curve is to probability 1, the better the model is said to be. Standing alone, all these classification algorithms work tremendously, but then a comparison must be made.

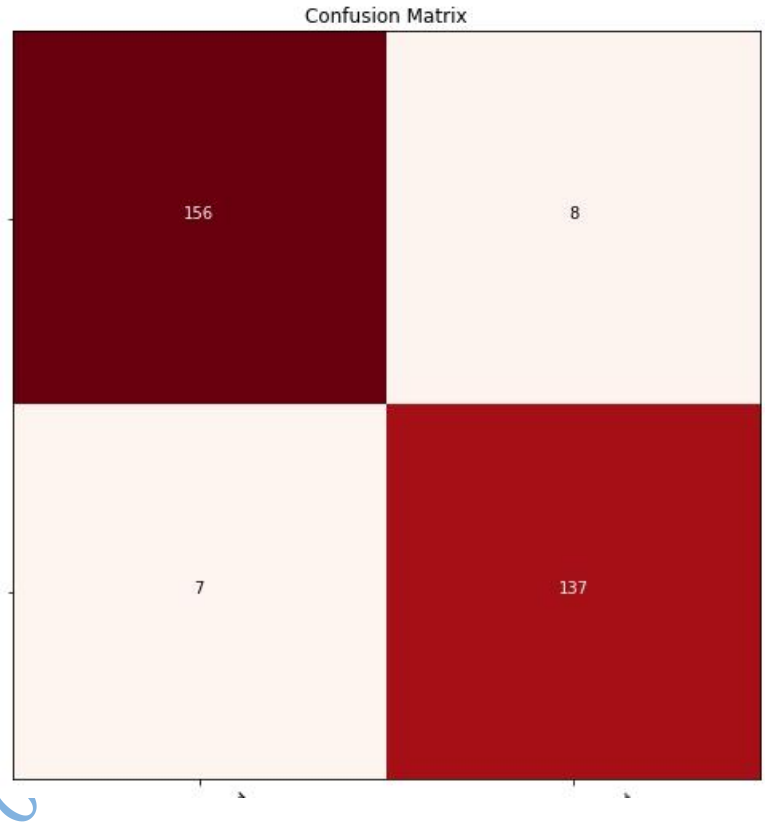


Figure 4.9a: Confusion Matrix Showing the Dataset with BA-GA and SVM (TP=156; TN=137; FP=8; FN=7) (Researcher, Nwufoh C.V: 2023)

Do NC ...geria

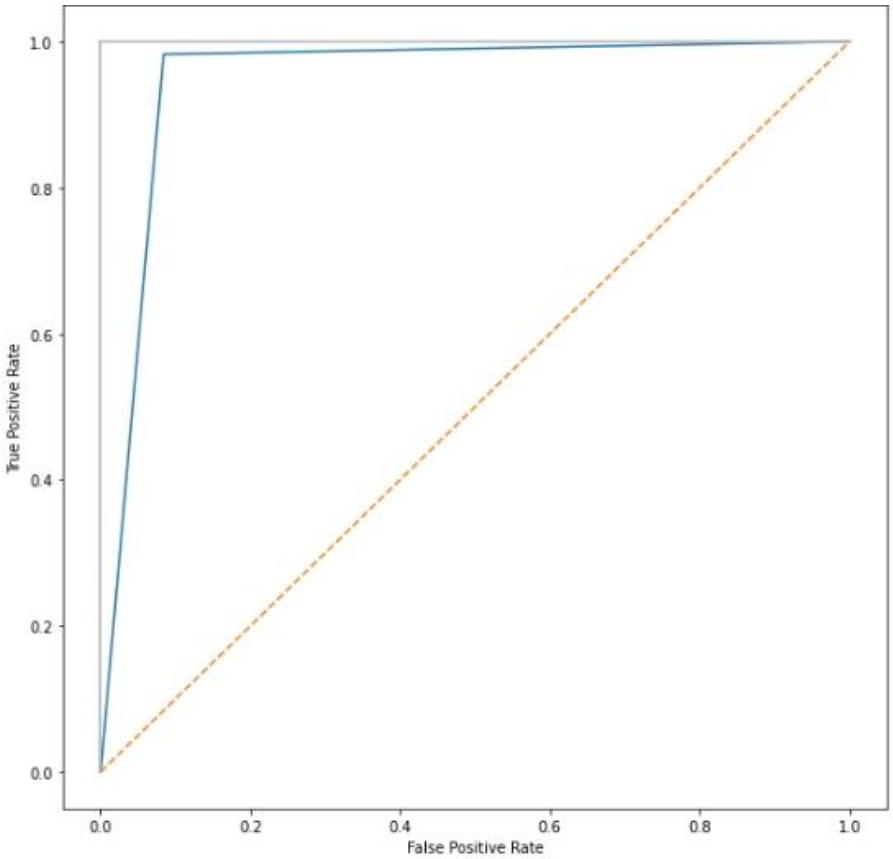


Figure 4.9b: ROC Curve for BA-GA and SVM (Researcher, Nwufoh C.V: 2023)

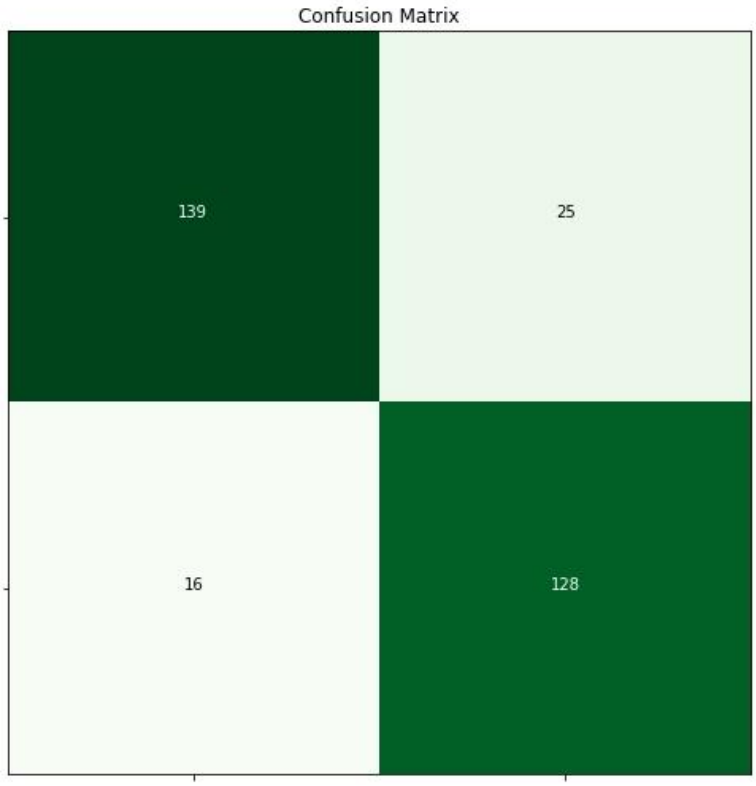


Figure 4.10a: Confusion Matrix Showing the Dataset with BA-GA and KNN (TP=139; TN=128; FP=25; FN=16) (Researcher, Nwufoh C.V: 2023)

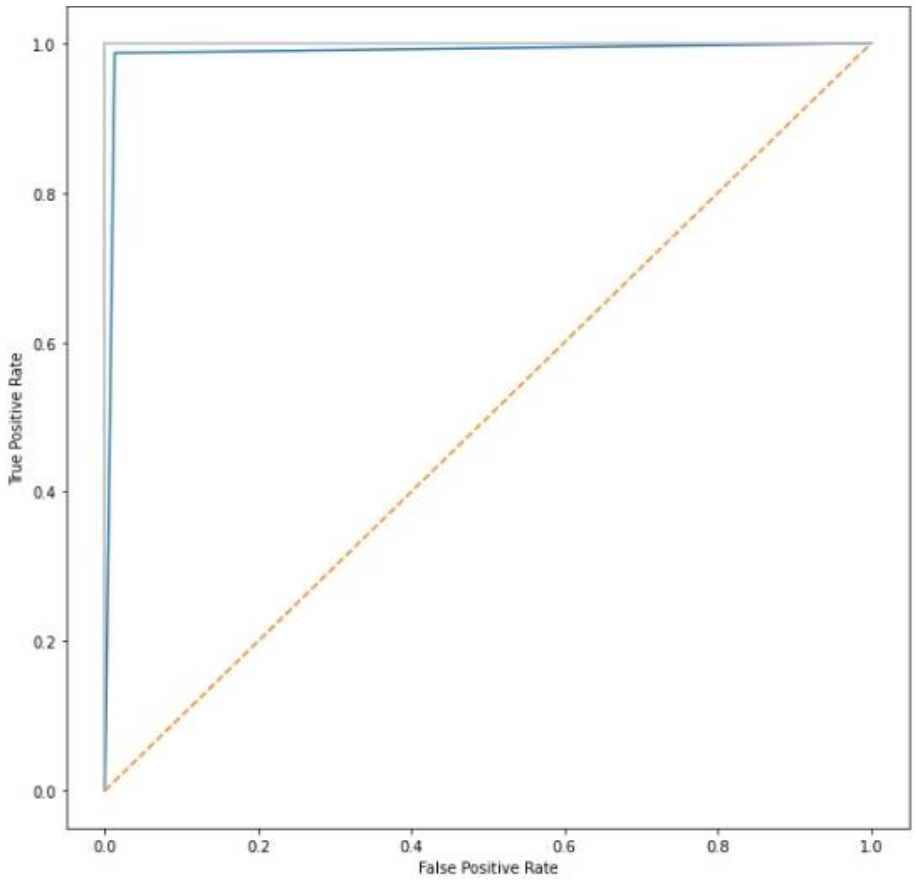


Figure 4.10b: ROC Curve for BA-GA and KNN (Researcher, Nwufoh C.V: 2023)

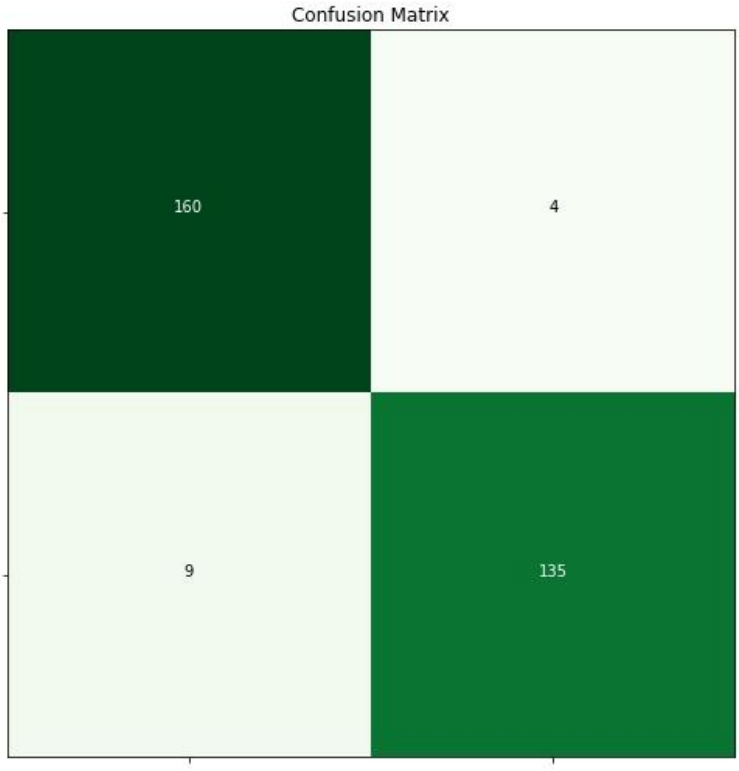


Figure 4.11a: Confusion Matrix Showing the Dataset with BA-GA and Ensemble

(TP=160; TN=135; FP=4; FN=9) (Researcher, Nwufoh C.V: 2023)

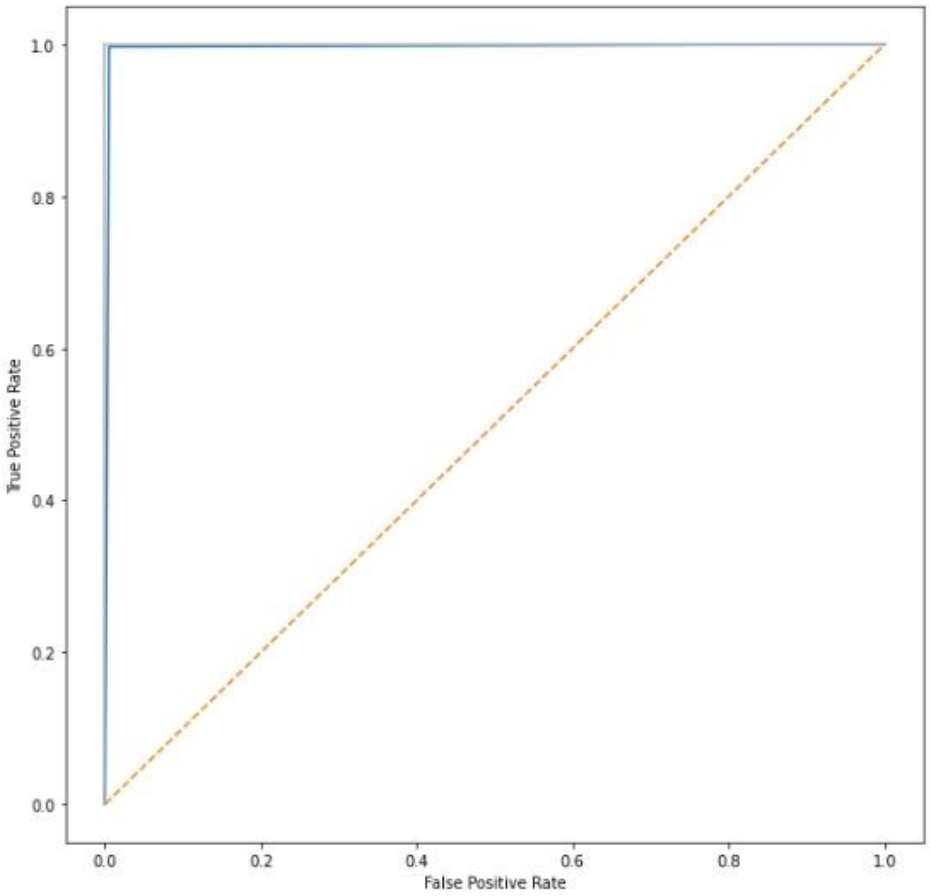


Figure 4.11b: ROC Curve for BA-GA and Ensemble (Researcher, Nwufoh C.V: 2023)

General Discussion of section 4.3.2: The classifiers for this study have excellent outcomes, but the essence of classifying with these three (3) algorithms is to give room for comparison and inferencing. In comparison for the Confusion Matrixes, it is observed that the classification of BA-GA DR Model using Ensemble (Ensemble-BA-GA) has TP = 160 while for SVM-BA-BA, TP = 156 and then for KNN-BA-GA, TP = 139. From these TP values, it can deduced that for the model BA-GA, the performance is better classified using Ensemble than SVM before KNN. As can be seen in Figure 4.9b, Figure 4.10b, and Figure 4.11b - for the ROC curves for all instances of the classifiers on BA-GA, it would be observed that the AUCs (Area Under the Curve) tends close to a TP of probability of 1. In ROC, the closer a result is to 1, the better the model. For SVM, K-NN, and Ensemble, we observe that the

Do Not Copy, Lea

AUCs tend to be 1, which implies that the model works great, but it is closest to Ensemble. In comparison, the outcome of ROC for BA-GA-Ensemble is said to be a better model, followed by BA-GA-KNN and then BA-GA-SVM because of the degree of nearness of the curve to 1.

4.3.3 SVN, K-NN, and Ensemble Classifications on ICA-BA-GA

By merging a variation of an already existing crossover operator with these heuristics, a Hybrid ICA with an Improved Genetic Algorithm has been developed. One heuristic generates the starting population, while the other two are applied to the offspring obtained by crossover or randomization. The improved genetic algorithms (GA) minimize cost functions that are nonconvex and nonlinear. This is a beneficial strategy for incorporating exogenous data into a learning machine when the search for independent components is the primary objective. As the input space dimension rises, the model given in this study can extract independent components faster than earlier independent component analysis techniques, demonstrating great accuracy and robustness. Figure 4.12a, 4.13a, and 4.14a shows the Confusion matrix of the developed model using ICA with BA-GA, while Figures 4.12b, 4.13b, and 4.14b shows the ROC curve for the model ICA-BA-GA.

Do Not Copy, Lead City University, Nigeria

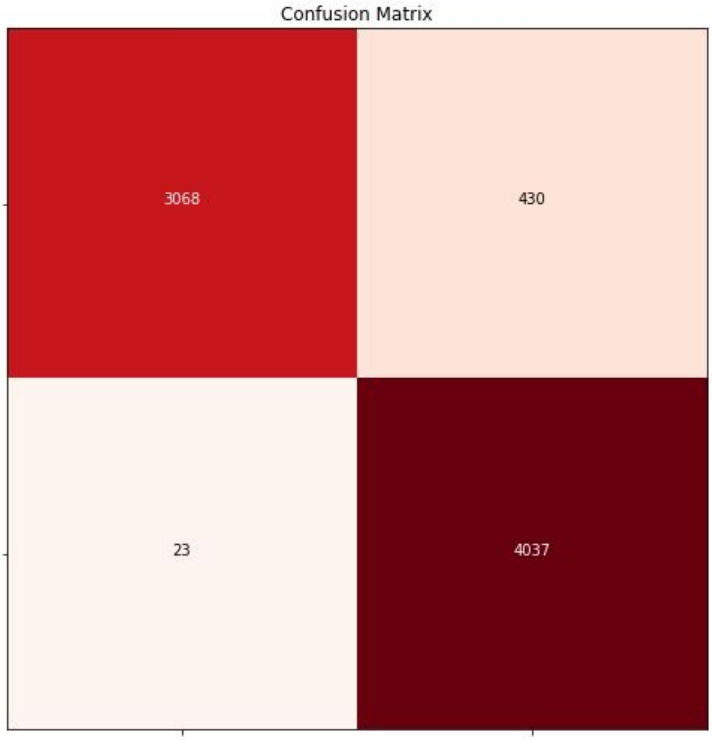


Figure 4.12a: Confusion matrix of ICA-BA-GA with SVM (TP=3068; TN=4037; FP=430; FN=23) (Researcher, Nwufoh C.V: 2023)

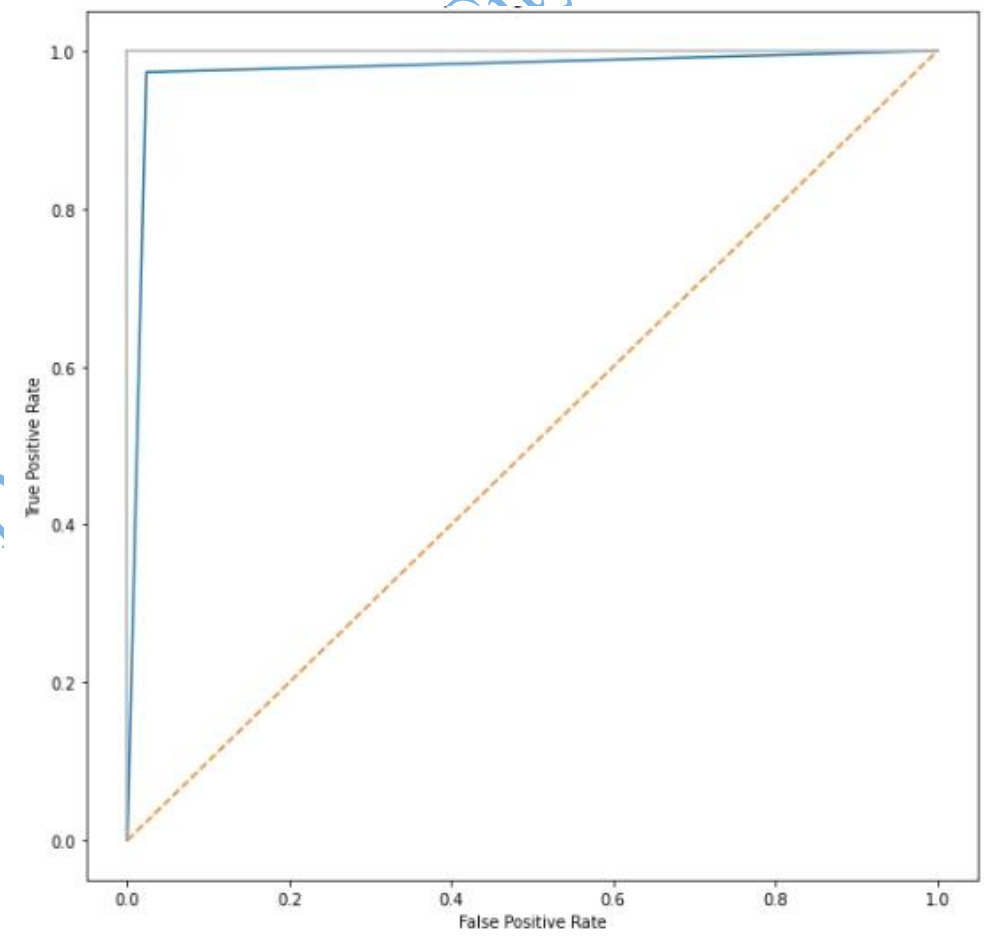


Figure 4.12b: ROC Curve for ICA BA-GA-SVM (Researcher, Nwufoh C.V: 2023)

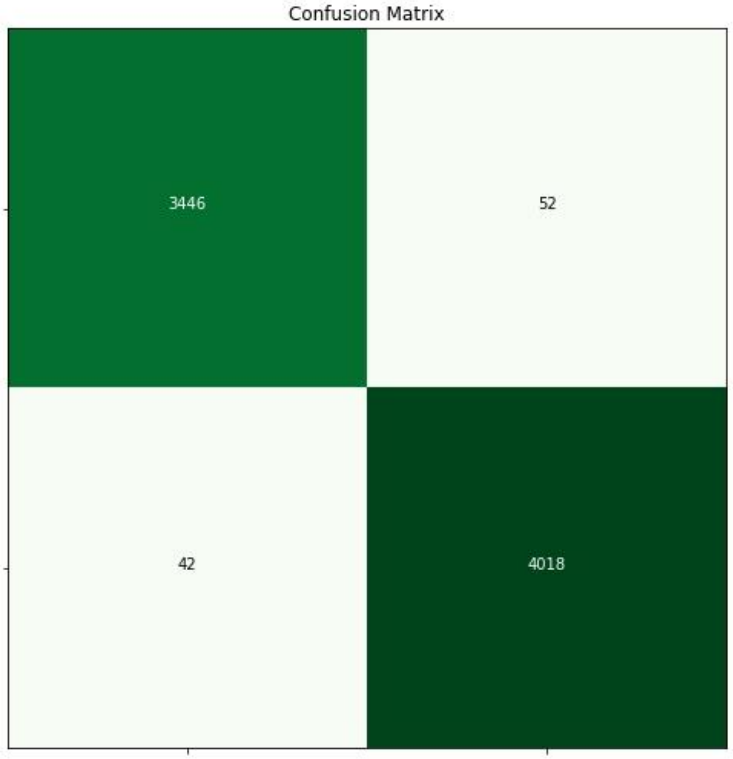


Figure 4.13a: Confusion Matrix Showing the Dataset with ICA-BA-GA and KNN (TP=3446; TN=4018; FP=52; FN=42) (Researcher, Nwufoh C.V: 2023)

Do Not Copy, Lead City University, Nigeria

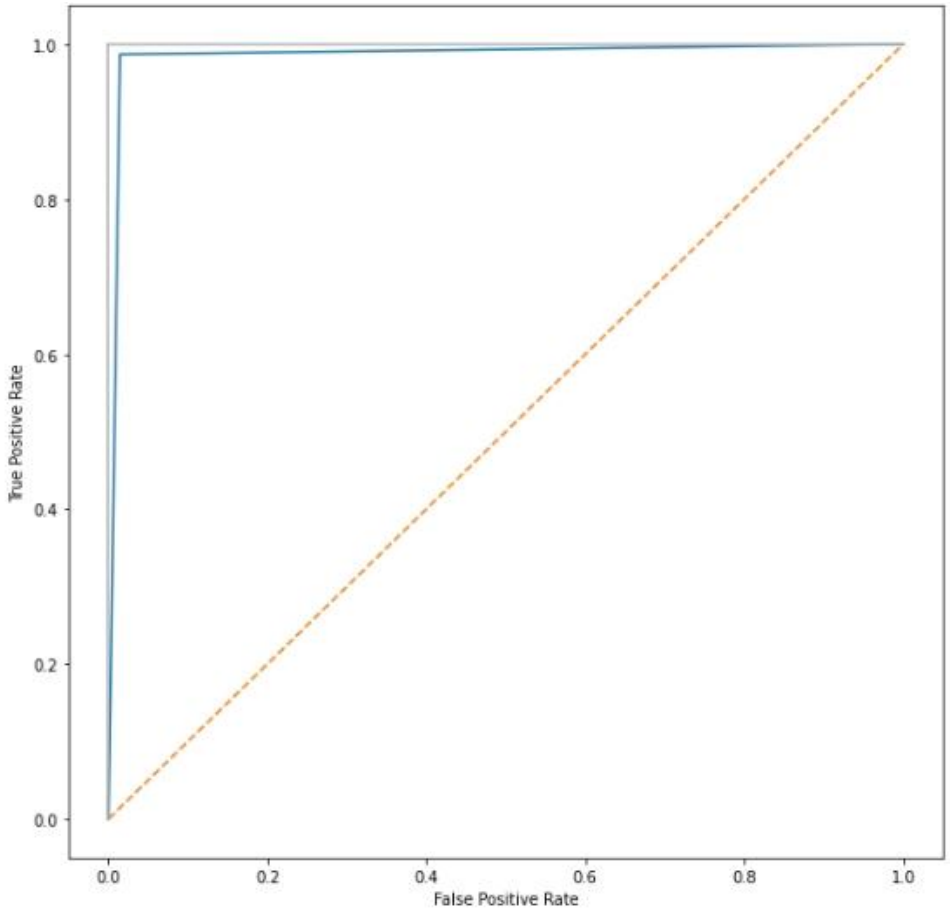


Figure 4.13b: ROC Curve for ICA BA-GA-KNN (Researcher, Nwufoh C.V: 2023)

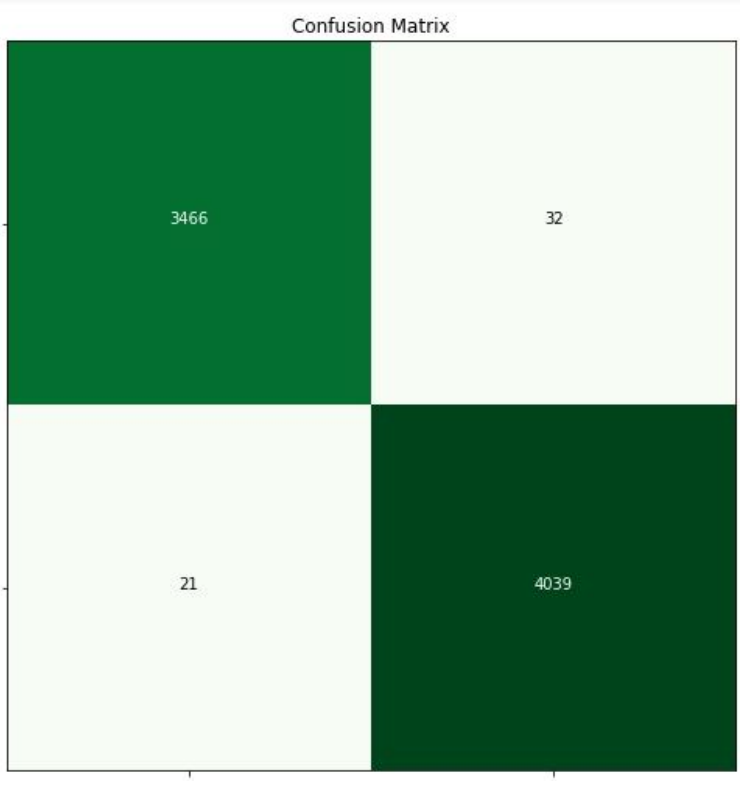


Figure 4.14a: Confusion Matrix Showing the Dataset with ICA-BA-GA and Ensemble (TP=3446; TN=4039; FP=32; FN=21) (Researcher, Nwufoh C.V: 2023)

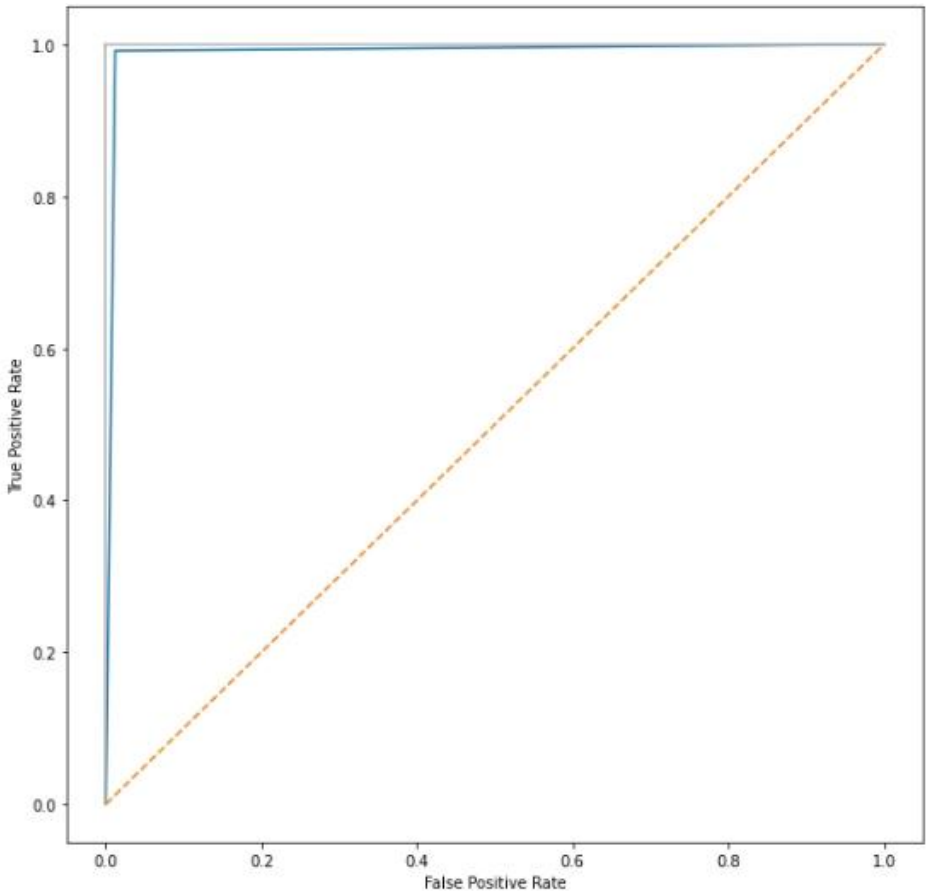


Figure 4.14b: ROC Curve for ICA-BA-GA-Ensemble (Researcher, Nwufoh C.V: 2023)

General Discussion of Section 4.3.3: In comparison of the Confusion Matrixes, it is observed in Figures 4.12a, 4.13a, and 4.14a that classification of ICA-BA-GA DR Model using Ensemble (Ensemble-ICA) has TP = 3466 < TN = 4049; SVM-ICA- BA-BA, TP = 3068 < TN= 4037 and then KNN-ICA -BA-GA, TP = 3446 < TN = 4018. From these TP and TN values, it is observed that in this fusion of ICA DR models and BA-GA DR model as opposed to their singular models, the TN outcome becomes greater than TP, as was the case in the other scenarios. This infers that the hybrid model does not just confirm the existence of blurred text in the image and deblurs. However, due to the Bird Approach novelty – all the image sets that are assumed to be sharp were further sharpened to give an enhanced deblurring of images in the wild. The model does not authenticate what is expected to be

Do Not Copy, L

accurate but searches out its valid values away from the expected. Hence, it can be deduced that the model ICA-BA-GA is a dynamic one.

As can be seen in Figure 4.12b, Figure 4.13b, and Figure 4.14b - for the ROC curves for all instances of the classifiers on ICA-BA-GA, it would be observed that the AUCs (Area Under the Curve) tend close to a TP of probability of 1. In ROC, the closer a result is to 1, the better the model. For SVM, K-NN, and Ensemble, we observe that the AUCs tend to be 1, which implies that the model works great, but it is closest to Ensemble. In comparison, the outcome of ROC for ICA-BA-GA-Ensemble is said to be a better model, followed by ICA-BA-GA-KNN and then ICA-BA-GA-SVM because of the degree of nearness of the curve to 1.

4.4 Evaluation Measures on the Models

This section shows the outcomes of the evaluations done on all the instances of the models using the following parameters: Accuracy, Precision, and F1-Score. It addresses objective iv. All the values were achieved using Python modules such as skitlearn. It shows the values from the calculations of the various performance evaluation parameters, as also seen in the confusion matrixes. The outcomes of these values vary in varying instances of the model (ICA, BA-GA, and ICA-BA-GA) created with the different classifiers (SVM, K-NN, and Ensemble). We use tables and bar charts to represent these values.

Do Not Copy, Lead City University, Nigeria

Table 4.1: Evaluation measures of ICA with Classifiers (Researcher, Nwufoh C.V: 2023)

Measures	ICA-SVM	ICA-KNN	ICA-Ensemble	Derivations
Accuracy	88.47	85.19	96.91	ACC = (TP + TN) / (P + N)
Sensitivity	90.45	86.25	98.13	TPR = TP / (TP + FN)

Specificity	85.43	82.43	95.27	SPC = TN / (FP + TN)
Precision	88.59	84.15	95.73	PPV = TP / (TP + FP)
F1-Score	88.47	85.19	96.91	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	76.02	68.78	93.52	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

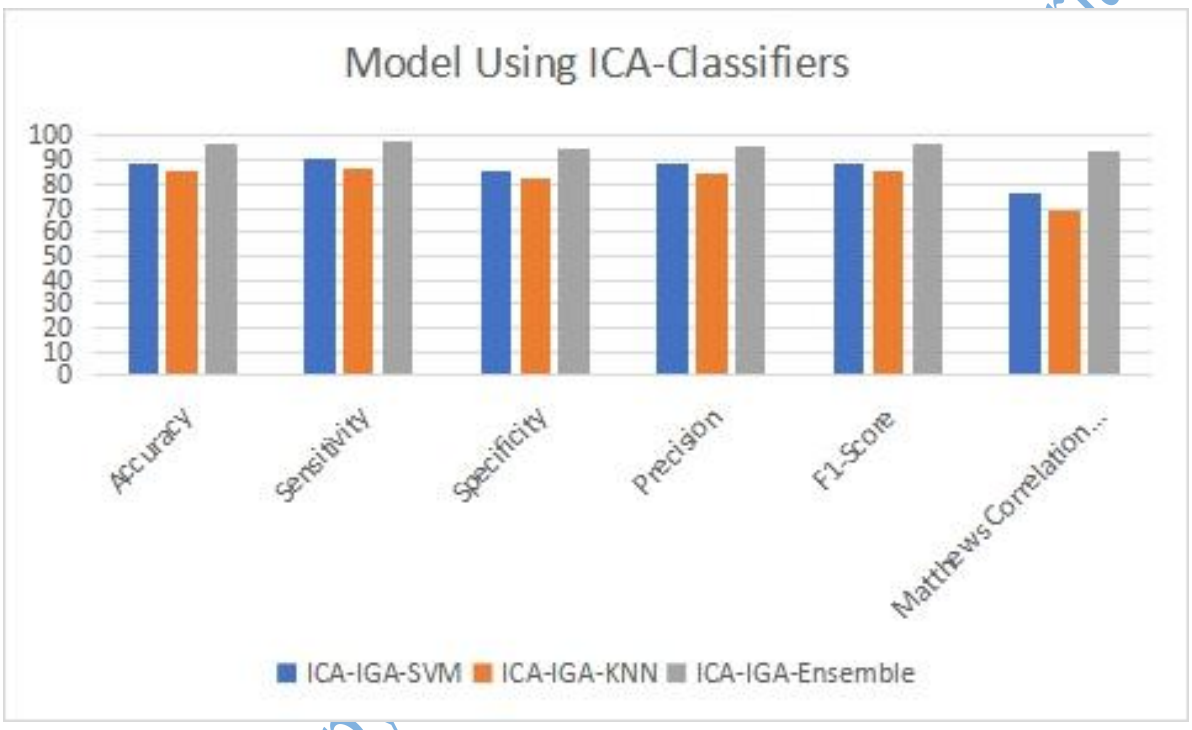


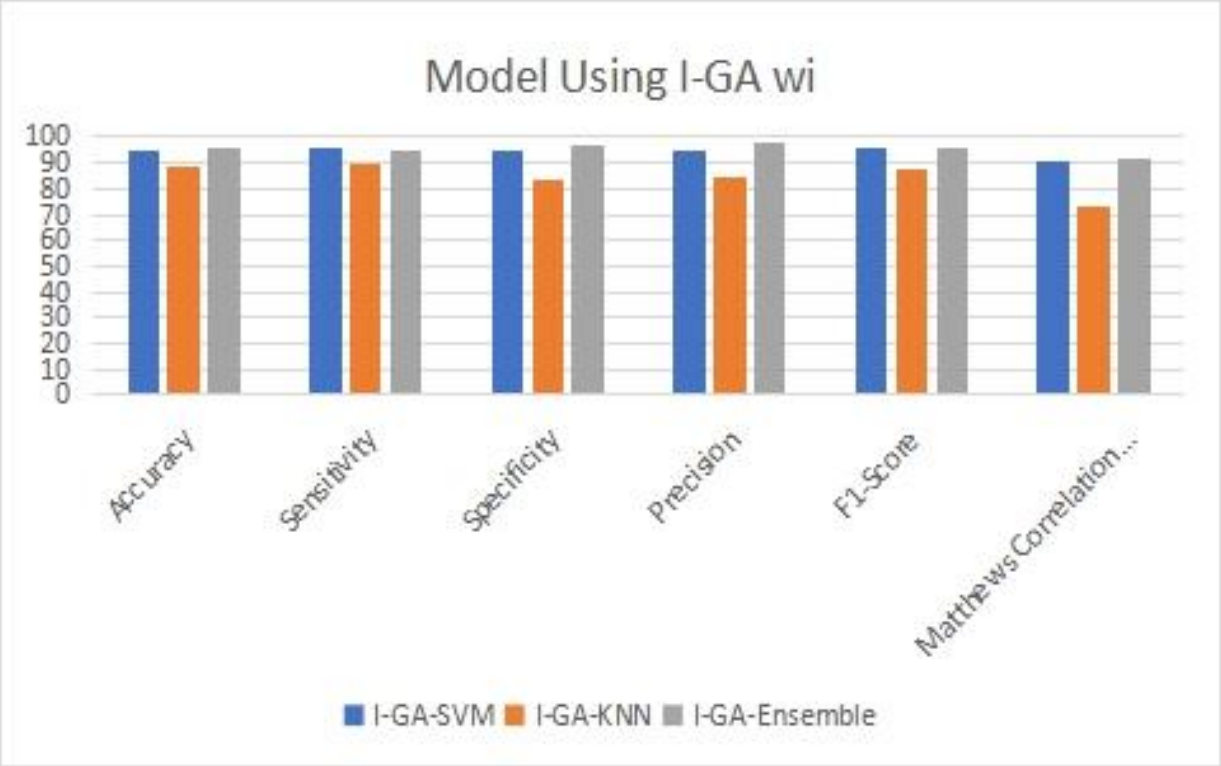
Figure 4.15: Performance Evaluation Model using ICA-Classifiers (Researcher, Nwufoh C.V: 2023)

Table 4.2: Evaluation measures of BA-GA - Classifiers (Researcher, Nwufoh C.V: 2023)

Measures	**BA-GA-SVM**	**BA-GA-KNN**	**BA-GA-Ensemble**	**Derivations**
Accuracy	95.13	88.69	95.78	ACC = (TP + TN) / (P + N)
Sensitivity	95.71	89.68	94.67	TPR = TP / (TP + FN)
Specificity	94.48	83.66	97.12	SPC = TN / (FP + TN)

Do Not Copy

Precision	95.14	84.76	97.56	PPV = TP / (TP + FP)
F1-Score	95.41	87.15	96.10	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	90.22	73.49	91.55	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))



Do Not Co

Figure 4:16: Performance Evaluation Model using BA-GA with the Classifiers (Researcher, Nwufoh C.V: 2023)

Table 4.3: Evaluation measures of ICA – BA-GA-Classifiers (Researcher, Nwufoh C.V: 2023)

Measures	**ICA-BA-GA-**	**ICA-BA-GA-**	**ICA-BA-GA-**	**Derivations**

	SVM	KNN	Ensemble	
Accuracy	94.01	98.65	99.30	ACC = (TP + TN) / (P + N)
Sensitivity	99.26	98.80	99.30	TPR = TP / (TP + FN)
Specificity	90.37	98.72	99.21	SPC = TN / (FP + TN)
Precision	87.71	98.51	99.08	PPV = TP / (TP + FP)
F1-Score	93.12	98.65	99.24	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	88.38	97.50	98.59	TP*TN - FP*FN / sqrt((TP+FP) *(TP+FN) *(TN+FP) *(TN+FN))

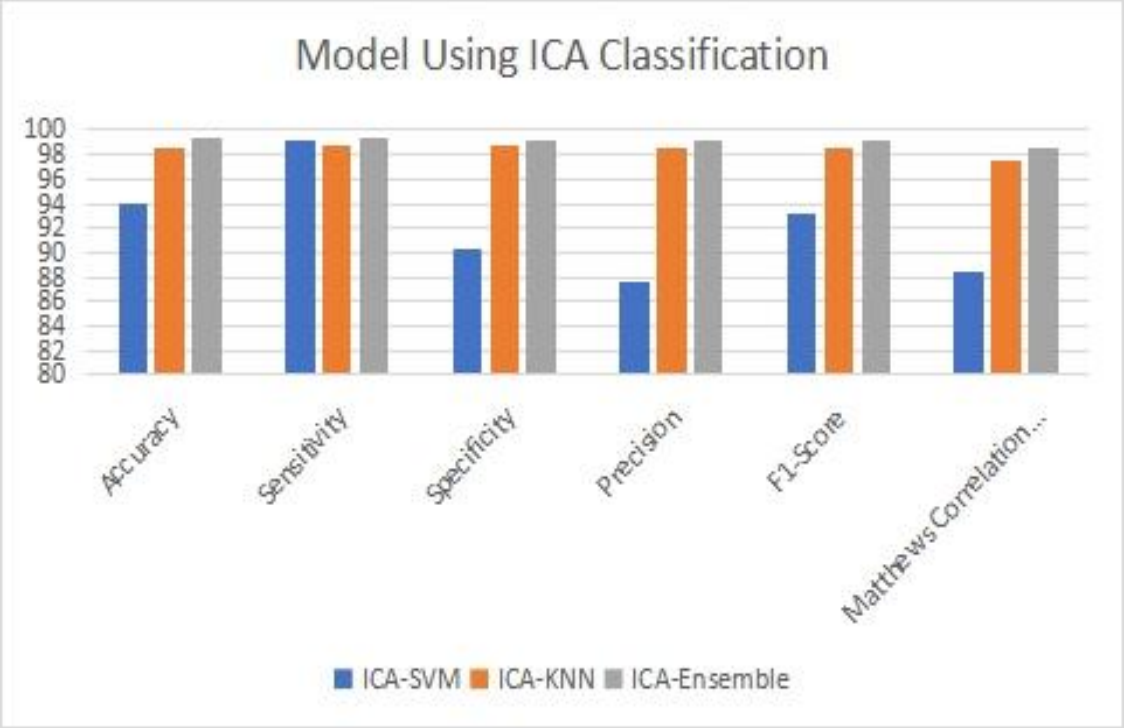


Figure 4.17: Performance Evaluation Model Using ICA-BA-GA Classification. (Researcher, Nwufoh C.V: 2023)

General Discussion on section 4.4: Using the Jupiter ecosystem, six (6) evaluation parameters were evaluated in this section, but in section 4.5, accuracy would be the focus, which is the parameter used to benchmark this study with others,

1. **For ICA-SVM, ICA-KNN, and ICA-Ensemble:** For the ICA-SVM model, it achieved an accuracy of 88.47%, indicating that it correctly classified 88.47% of the instances. The sensitivity (true positive rate) is 90.45%, which means the model successfully identified 90.45% of the positive instances. The specificity (true negative rate) is 85.43%, indicating that the model accurately identified 85.43% of the negative instances. The precision (positive predictive value) is 88.59%, which shows the proportion of correctly predicted positive instances out of all instances predicted as positive. The F1 Score, which considers the balance between precision and sensitivity, is 88.47%.

The ICA-KNN model performed well across all metrics, achieving an accuracy of 85.19%, indicating a satisfactory level of correct classifications. The sensitivity is 86.25%, indicating a remarkable ability to identify positive instances. The specificity is 82.43%, demonstrating an excellent ability to identify negative instances. The precision is 84.15%, indicating a positive proportion of predicting positive instances. The F1 Score, 85.19%, reflects a harmonious balance between precision and sensitivity.

The ICA-Ensemble model shows excellent performance across all metrics. It achieves an accuracy of 96.91%, indicating a high level of correct classifications. The sensitivity was 98.13%, indicating a solid ability to identify positive instances. The specificity is 95.27%, demonstrating a high accuracy in identifying negative instances. The precision is 95.73%, indicating a high proportion of correctly predicted positive instances—the F1 Score is 96.91%, reflecting a harmonious balance between precision and sensitivity.

These results suggest that the ICA-Ensemble model outperformed the ICA-SVM and then ICA-KNN regarding classification accuracy, sensitivity, specificity, precision, and F1 Score. The higher accuracy and F1 Score of the ICA-Ensemble model indicate its better overall performance in correctly classifying instances and maintaining a balance between precision

Do Not Copy, Lead City University, Nigeria

and sensitivity. While the ICA-Ensemble model shows superior performance in this study, the ICA-SVM and ICA-KNN models, on their rights, still achieved reasonably good results and may be more suitable for specific scenarios. The ICA approach combined with SVM, KNN, and Ensemble classifiers demonstrates promising performance in classification tasks.

2. **For BA-GA-SVM, BA-GA-KNN, and BA-GA-Ensemble:** The BA-GA-SVM model achieved an accuracy of 95.13%, indicating that it correctly classified a high percentage of the instances. The sensitivity (true positive rate) is 95.71%, which means the model successfully identified that percentage of the positive instances. The specificity (true negative rate) is 94.48%, indicating that the model accurately identified 94.48% of the negative instances. The precision (positive predictive value) is 95.14%, which shows the proportion of correctly predicted positive instances out of all instances predicted as positive. The F1 Score, which considers the balance between precision and sensitivity, is 95.41%.

The BA-GA-KNN model performed well across all metrics, achieving an accuracy of 88.69%, indicating a satisfactory level of correct classifications. The sensitivity is 89.68%, indicating a remarkable ability to identify positive instances. The specificity is 83.66%, demonstrating an excellent ability to identify negative instances. The precision is 84.76%, indicating a positive proportion of predicting positive instances. The F1 Score, 87.15%, reflects a satisfactory balance between precision and sensitivity.

The BA-GA-Ensemble model shows an excellent performance across all metrics. It achieves an accuracy of 95.78%, indicating a high level of correct classifications. The sensitivity was 94.67%, indicating a solid ability to identify positive instances. The specificity is 97.12%, demonstrating a high accuracy in identifying negative instances. The precision is 97.56%,

Do Not Copy, Lead City University, Nigeria

indicating a high proportion of correctly predicted positive instances—the F1 Score is 96.10%, reflecting a harmonious balance between precision and sensitivity.

Also, these results suggest that the BA-GA-Ensemble model outperformed the BA-GA-SVM and then BA-GA-KNN regarding classification accuracy, sensitivity, specificity, precision, and F1 Score. The higher accuracy and F1 Score of the ICA-Ensemble model indicate its better overall performance in correctly classifying instances and maintaining a balance between precision and sensitivity. While the BA-GA-Ensemble model shows superior performance in this study, the BA-GA-SVM and BA-GA-KNN models, on their rights, still achieved reasonably good results and may be more suitable for specific scenarios.

3. **For ICA-BA-GA-SVM, ICA-BA-GA-KNN, and ICA-BA-GA-Ensemble:** For the ICA-BA-GA-SVM model, it achieved an accuracy of 94.01%, indicating that it correctly classified a high percentage of the instances. The sensitivity (true positive rate) is 99.26%, which means the model successfully identified that percentage of the positive instances. The specificity (true negative rate) is 90.37%, indicating that the model accurately identified 90.37% of the negative instances. The precision (positive predictive value) is 95.14%, which shows the proportion of correctly predicted positive instances out of all instances predicted as positive. The F1 Score, which considers the balance between precision and sensitivity, is 87.71%.

The ICA-BA-GA-KNN model performed well across all metrics, achieving an accuracy of 98.65%, indicating a satisfactory level of correct classifications. The sensitivity is 98.80%, indicating a remarkable ability to identify positive instances. The specificity is 98.72%, demonstrating an excellent ability to identify negative instances. The precision is 98.51%,

Do Not Copy, Lead City University, Nigeria

indicating a positive proportion of predicting positive instances. The F1 Score, 98.65%, reflects a satisfactory balance between precision and sensitivity.

The ICA-BA-GA-Ensemble model shows excellent performance across all metrics. It achieves an accuracy of 99.30%, indicating a high level of correct classifications. The sensitivity was 99.30%, indicating a solid ability to identify positive instances. The specificity is 99.21%, demonstrating a high accuracy in identifying negative instances. The precision is 99.08%, indicating a high proportion of correctly predicted positive instances—the F1 Score is 99.24%, reflecting a harmonious balance between precision and sensitivity.

Also, these results suggest that the ICA-BA-GA-Ensemble model outperforms ICA-BA-GA-SVM and ICA-BA-GA-KNN. The higher accuracy and F1 Score of the ICA-Ensemble model indicate its better overall performance in correctly classifying instances and maintaining a balance between precision and sensitivity. While the ICA-BA-GA-Ensemble model shows superior performance in this study, the ICA-BA-GA-SVM and ICA-BA-GA-KNN models, on their rights, still achieved reasonably good results and may be more suitable for specific scenarios.

From all the evaluation values, it can be observed that for all the instances of the model developed (ICA; BA-GA and ICA-BA-GA), the classification with Ensemble, in most cases, gave higher scores for all the parameters. Hence, with Ensemble, the performance of these models is evaluated efficiently, and scholars should consider using the Ensemble algorithm for classification more often.

Do Not Copy, Lead City University, Nigeria

4.5 Conclusion and Benchmarking with Existing Results

These findings suggest a machine-learning approach combining ICA and an Improved GA to address blur and jitter in scenes. Figure 4.5 shows some of the results obtained from the blurred scenes using the hybrid DR. To begin, the dataset, which consists of images blurred in a variety of ways, is initially deblurred by DR using ICA, then further deblurring using an improved GA (BA-GA). After this, we combined both DR models to create ICA-BA-GA. All the DR models were trained using features taken from both blurred and clear images, and then classifiers SVM, KNN, and Ensemble were used to sort the recognition data. Since the top three solutions on the Charades short video dataset have attained 99% of the results, it is clear that the suggested method performs well in blurring scene recognition.

This research develops an advanced machine learning strategy for performing well on Blur Scene text datasets by combining ICA and an Improved GA using a bird approach. Sequence learning performance has proven influential in recent efforts, particularly in text transcription and speech recognition. The model facilitates the acquisition of quick and straightforward learning and preprocessing datasets, which in turn facilitates an enhanced reduction in error rate. According to this study, 99.99 percent employ a crossbred ICA-BA-GA-Ensemble. The outcomes of this model are superior to those of simpler models. Compared to outcomes obtained using other language modeling and misfit regularization methods, our model performed exceptionally well. Table 4.4 compares the results with existing models.

Do Not Copy, Lead City University, Nigeria

Table 4.4: Comparison with Existing Methods (Researcher, Nwufoh C.V: 2023)

AUTHOR	TECHNIQUE	RESULTS
(Butt et al. 2021)	CNN-RNN	87%
(Kantipudi, Kumar, and Kumar Jha 2021)	LSTM-DNN	98%
(Pandey et al. 2021)	DNN-PSO	95%
(Francis and Sreenath 2022)	SVM	84%
(Ansari et al. 2021)	GA-SVM	92%
(Researcher, Nwufoh C.V. 2023)	ICA-BA-GA-K-NN	98.65
(Researcher, Nwufoh C.V. 2023)	ICA-BA-GA-ENSEMBLE	99.30

4.6 Summary of Results Obtained

In this section, using a table, we will be relating the objectives, research questions, and the results obtained.

Table 4.5 Relating Research Questions and Objectives to the Results Obtained

S/N	Research Questions	Objectives	Method	Results
i	How to fetch out the very high dimensional vector from the dataset	By Designing a DR model for pre-processing	Using ICA	Explained in section 4.2.1. Produces figures 4.1, 4.2 and 4.3
ii	How to improve the model for fetching of high dimensional sparse vector to improve text deblurring?	By hybridizing the DR model in (I) above with an improved GA	Using ICA created in (I) above and then a bird approach improved GA as explained in section 3.3.2.1	Explained in section 4.2.2 Figures 4.4 and 4.5 shows the outcome of the model developed.
iii	How do we test the	Classification	Using SVM, K-	Illustrated in section 4.3,

	model created?	using SVM, K-NN and Ensemble	NN and Ensemble on every instance of the model created.	on all the instances of the models (ICA; BA-GA; ICA BA-GA) in sections 4.3.1, 4.3.2 and 4.3.3 respectively. With confusion matrixes and ROC curve to display the outcomes.
iv	How do we evaluate the performance of the model?	Performance evaluation using accuracy, precision and f1 scores		Seen in section 4.4 with different tables and bar charts for each instance of evaluations.
v	What is the outcome of comparing the result of evaluation with state of the art?	Comparison with existing works		Seen in section 4.5 in Table 4.4

Table: 4. 6: Accuracies of the DR Models Developed in this study (Researcher, Nwufoh C.V: 2023)

TECHNIQUE	ACCURACY
ICA-SVM	88.47
ICA-KNN	85.19
ICA-ENSEMBLE	96.91
BA-GA-SVM	95.13
BA-GA-KNN	88.69
BA-GA-ENSEMBLE	95.78
ICA- BA-GA-SVM	94.01

Copy, Lead

ICA-BA-GA-K-NN	98.65
ICA-BA-GA-ENSEMBLE	99.30

This comparison is made based on the evaluation parameter 'accuracy.' Table 4.4. gives us some accuracies of algorithms that have been combined in studies like this, while Table 4.6 gives a summary of the accuracies of all the instances of the models created in this study. The last two techniques in Table 4.5 with the hybridized models had the best accuracy outcomes. Also, from that table, we can infer that using Ensemble to classify our models always gave us better outcomes.

Do Not Copy, Lead City University, Nigeria

Chapter Five

Conclusion

5.1 Summary of Results

Numerous Recognition and Detection techniques have been presented due to their usefulness in various fields of study. Despite significant advancements in the field, including advanced learning techniques, ad hoc pre- and post-processing procedures, and dimensionality reduction techniques are commonly used to increase the text identification rate by eliminating both false positives and negatives. Another problem is that the contrasting perspectives offered by various text detection techniques are rarely used together. To address these shortcomings, this research infuses various perspectives. It develops a computing framework based on the Independent Component Analysis (ICA) with an improved Genetic Algorithm for machine learning to direct the definition of appropriate post-processing procedures via the combination of basic operators that can be used to enhance text detection results provided by multiple methods simultaneously.

Using Machine Learning (ICA) models, we demonstrate a technique for extracting blurred scenes from natural scenes, the first Dimensionality Reduction model. After this, we employed the use of an improved GA, which is a Hybridized Dimensionality Reduction Machine Learning Model, for selecting the relevant features in the first ICA DR model, blurred scene images, thereby sharpening the latent image. The photos are from a public dataset that any intelligent system can use. The revised Genetic algorithm used in the method is computationally fast and yields superior accuracy results to the alternatives because of the introduction of a Bird Approach (BA). A Bird Approach illustrates how birds keep hovering around, lying in wait to peck on a particle that the human eyes cannot see; so the BA here helps the GA select extra optimal features in the blurred images to detect text in the blurred images. The approach's efficacy is examined using SVM, K-NN, and Ensemble Classification

Do Not Copy, Lead City University, Nigeria

Algorithms. The classification result is evaluated by calculating using metrics like Accuracy, Precision, and the F-measure. The method achieves an impressive 99% Accuracy with the Ensemble Classifier, as seen by the results.

5.1.1 Conclusion

In Conclusion, this study, first and foremost, develops a strategy for successfully combining (hybridization) different types of machine learning for dimensionality reduction. To achieve its goals, this study models fusion as an optimization process and benefits from the framework of a genetic algorithm and independent component analysis. The experimental outcomes show that this strategy produces efficient outcomes for standard benchmarks. According to the findings, our method has the potential to enhance the efficiency of blurred picture text detections and allow the creation of apps for devices with limited processing capabilities.

Hence, high-quality learning methods also make it a potentially useful alternative to state-of-the-art text detectors in deployment settings that permit offline processing, as well as for the development of data-driven post-processing techniques.

5.2 Contribution to Knowledge

This study contributes to the body of existing knowledge in the following areas:

1. First, this study deals with an area of text detection and recognition that is still relatively untapped, which is blurred text recognition in the natural scene. Most scholars have recommended the study of text recognition for irregular text, in which blurred text is one such.
2. Secondly, this study developed a novel model for Dimensionality Reduction using GA infused with the Bird Approach, which helps fine-tune the optimal best function of GA. We also did not just develop BA-GA but also created a model that combines GA with ICA for a hybridized DR. An enhanced integrated dimensional reduction module

Do Not Copy, Lead City University, Nigeria

(ICA-BA-GA) is presented to better utilize picture content via global channel attention, hence resolving the image scale variation problem and yielding more representative features for images.

3. Also, the accuracy of text detection was significantly increased when ICA and BA-GA were applied to scene text detection compared to the already existing usage of related models. This optimized model achieves an impressive 99% Accuracy compared to existing models.
4. Lastly, the optimized ICA-BA-GA models' performance was tested using SVM. K-NN and Ensemble classification. The accuracy result for Ensemble came out best, followed by K-NN, then SVM

5.3 Suggestions for Further Studies

This study presents the following recommendations, though not limited.

1. Future research can focus on developing hybridized dimensionality reduction initiatives using deep learning algorithms like CNN, RNN, and DNN to generate the efficient procedures of features from the contoured image and adding novel operators to continue improving the hybrid dimensional reduction feature investigation.
2. Again, researchers can work with static images and not scenic images using the methods developed in this study. Once scenic images can be used for this study, static images can also pass for any researcher who desires to work on those.
3. On the other hand, scholars can try their hands on other datasets since this study focuses on ICDAR 2019 SLVT, though voluminous and robust.
4. Some other researchers, if ample time and state-of-the-art resources are at their disposal, can curate datasets for themselves, giving them opportunities to work with blurred text images caused by varying conditions such as hazy images, grayscale

Do Not Copy, Lead City University, Nigeria

blurred images, and angle distortion blurred images, all in one dataset. This would birth models that would be efficient across the board.

Do Not Copy, Lead City University, Nigeria

Bibliography

Chapter in a Book

Algren, M, Fisher W., & Landis A. E. "*Machine Learning in Life Cycle Assessment*." In **Data Science Applied to Sustainability Analysis. Elsevier**, 2021:167-190.

Ali-Gombe, A., Elyan, E., Moreno-García, C.& Jayne, C. 2022. *Cross Domain Evaluation of Text Detection Models.* In: Pimenidis, E., Angelov, P., Jayne, C., Papaleonidas, A., Aydin, M. (eds) **Artificial Neural Networks and Machine Learning – ICANN 2022**. ICANN 2022. Lecture Notes in Computer Science, vol 13531. Springer, Cham. doi:10.1007/978-3-031-15934-3_5. 2022

Anand S., Seba S., Aggarwal S., Aggarwal S., & Singla R. "*Scene Text Recognition in the Wild with Motion Deblurring Using Deep Networks.*" In **Communications in Computer and Information Science**, 1378 CCIS, 2021:93 – 103.

Ashour, A. S., & Guo Y.. "*Optimization-Based Neutrosophic Set in Computer-Aided Diagnosis.*" In **Optimization Theory Based on Neutrosophic and Plithogenic Sets**. Elsevier, 2020: 405 – 421.

Awad M., & Khanna R.. "*Support Vector Machines for Classification*." In **Efficient Learning Machines**. Berkeley, CA: Apress, 2015: 39 – 66.

Bansal, P., Lamba R., Jain V. Jain T., Shokeen S., Kumar S., Singh P. K. & Khan B. "*GGA-MLP: A Greedy Genetic Algorithm to Optimize Weights and Biases in Multilayer Perceptron*." Edited by Yuvaraja Teekaraman. **Contrast Media & Molecular Imaging** 2022. February 2022: 1–14.

Bartholomew, D. J. "*Principal Components Analysis*." In **International Encyclopedia of Education**. Elsevier, 2010: 374 – 377

Bautista, D. Atienza, R. *Scene Text Recognition with Permuted Autoregressive Sequence Models*. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) **Computer Vision – ECCV 2022. ECCV 2022**. Lecture Notes in Computer Science, vol 13688. Springer, Cham. doi:10.1007/978-3-031-19815-1_11. 2022.

Bolón-Canedo, V., & Alonso-Betanzos A. "*Foundations of Ensemble Learning*." In **Intelligent Systems Reference Library,** 147, 2018: 39 – 51

Brunton, S. L., & Kutz J. N. "*Singular Value Decomposition (SVD).*" **In *Data-Driven Science and Engineering*, 3–46. Cambridge University Press**, 2019.

Calabrese, B. "*Data Reduction*." In **Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics**, 1–3. Elsevier, 2018: 480 – 485.

Chanal, D., Steiner N. Y., Petrone R., Chamagne D, & Péra M-C. "*Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering*." In **Reference Module in Earth Systems and Environmental Sciences**. Elsevier, 2021.

Do Not Copy, Lead City University, Nigeria

Endalie, D., & Tegegne T. "*Designing a Hybrid Dimension Reduction for Improving the Performance of Amharic News Document Classification.*" Edited by Thippa Reddy Gadekallu. **PLoS ONE 16**, no. 5 May 2021: e0251902.

Hassanat A., Khalid A., Alkafaween E. , Eman A., Awni H., & Surya P. "*Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach*." **Information MDPI** 2019.

Hsu, P., & Chen B. Y. "*Blurred Image Detection and Classification*." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **4903 LNCS:277–286. Berlin, Heidelberg: Springer** Berlin Heidelberg, 2008.

Jain, S., & Salau A. O. "*An Image Feature Selection Approach for Dimensionality Reduction Based on KNN and SVM for AkT Proteins*." Edited by Wei Meng. **Cogent Engineering 6**, no. 1, January 2019.

Prasad K. MVV, Kumar S., & Jha A. K. "*Scene Text Recognition Based on Bidirectional Lstm and Deep Neural Network*." Edited by Gaurav Singal. **Computational Intelligence and Neuroscience** 2021 November 2021: 1–11.

Keserwani P., Saini R., Liwicki M. and Roy P. P."*Robust Scene Text Detection for Partially Annotated Training Data,*" in ***IEEE Transactions on Circuits and Systems for Video Technology, vol***. 32, no. 12, doi: 10.1109/TCSVT.2022.3194835. Dec 2022, pp 8635 - 8645

Khan, W. A., Hamadneh N. N, Tilahun S. L., & Ngnotchouye J. M. T. "*A Review and Comparative Study of Firefly Algorithm and Its Modified Versions*." In **Optimization Algorithms - Methods and Applications.** InTech, 2016.

Kherif F, & Latypova A. '*Chapter 12 - Principal component analysis'*, Edited by Andrea Mechelli & Sandra Vieira, **Machine Learning, Academic Press**, ISBN 9780128157398, doi:10.1016/B978-0-12-815739-8.00012-2, 2020, Pages 209-225.

Knekta, E., Runyon C., & Eddy S. "*One Size Doesn't Fit All: Using Factor Analysis to Gather Validity Evidence When Using Surveys in Your Research*." Edited by Peggy Brickman. **CBE Life Sciences Education 18**, no. 1, March 2019: rm1.

Lewis-Beck, M., Bryman A., & Liao T. F. "*Canonical Correlation Analysis*." In **The SAGE Encyclopedia of Social Science Research Methods**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 385 – 395.

Liao M, Zou Z, Wan Z, Yao C and Bai X, *"Real-Time Scene Text Detection With Differentiable Binarization and Adaptive Scale Fusion,"* in ***IEEE Transactions on Pattern Analysis and Machine Intelligence,*** vol. 45, no. 1, doi: 10.1109/TPAMI.2022.3155612. January 2023: 919 – 931.

Do Not Copy, Lead City University, Nigeria

Liu H, Burnap P, Alorainy W & Williams M. L, "*A Fuzzy Approach to Text Classification With Two-Stage Training for Ambiguous Instances*," in ***IEEE Transactions on Computational Social Systems***, vol. 6, no. 2, doi: 10.1109/TCSS.2019.2892037. April 2019, pp. 227-240.

Mi, J. X. "*A Novel Algorithm for Independent Component Analysis with Reference and Methods for Its Applications*." Edited by Hans A. Kestler. **PLoS ONE 9**, no. 5 May 2014: e93984.

Misra, S., & Wu Y. "*Machine Learning Assisted Segmentation of Scanning Electron Microscopy Images of Organic-Rich Shales with Feature Extraction and Feature Ranking.*" In **Machine Learning for Subsurface Characterization**. Elsevier, 2019: 289–314.

Monteiro, F. C., & Campilho A. C. "*Performance Evaluation of Image Segmentation*." **Lecture Notes in Computer Science** *(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4141 LNCS 2006: 248–259.

Patil, S. V., & Kulkarni D. B. "*A Review of Dimensionality Reduction in High-Dimensional Data Using Multi-Core and Many-Core Architecture*." In **Communications in Computer and Information Science**, 2019. 964:54–63.

Pisner D. A, & Schnyer D. M,. '*Chapter 6 - Support vector machine*,'Edited by Andrea Mechelli & Sandra Vieira. **Machine Learning, Academic Press**, ISBN 9780128157398, B978-0-12-815739-8.00006-7, 2020, Pages 101-121.

Pogorelov, K., Olga Ostroukhova O., Jeppsson M., Espeland H., Griwodz C., Thomas De Lange , Johansen D., Riegler M., & Halvorsen P. "*Deep Learning and Hand-Crafted Feature Based Approaches for Polyp Detection in Medical Videos*." In **Proceedings - IEEE Symposium on Computer-Based Medical Systems**, 2018-June. IEEE, 2018: 381–386.

Ranjani, J. Jennifer, & C. Jeyamala. "*Machine Learning Algorithms for Medical Image Security.*" **In Intelligent Data Security Solutions for E-Health Applications**. Elsevier, 2020: 169–183.

Sadiq, A. S., Faris H., Al-Zoubi A. M., Mirjalili S., & Ghafoor K. Z. "*Fraud Detection Model Based on Multi-Verse Features Extraction Approach for Smart City Applications*." In **Smart Cities Cybersecurity and Privacy**. Elsevier, 2018: 241–251.

Sahoo, M. K., Nayak J., Mohapatra S., Nayak B. K., & Behera H. S. "*Character Recognition Using Firefly Based Back Propagation Neural Network.*" In **Smart Innovation, Systems and Technologies**, 2015, 32:151–164.

Vieira S., Pinaya W H L, Garcia-Dias R. & Mechelli A. "*Multimodal Integration,*" in **Machine Learning Academic Press**, ISBN 9780128157398, 2020, Pages 283-305.

Do Not Copy, Lead City University, Nigeria

Conference Paper

Alkhateeb, J. H., Turani A. A., & Alsewari A. A. "*Performance of Machine Learning and Deep Learning on Arabic Handwritten Text Recognition*." In **ETCCE 2020 - International Conference on Emerging Technology in Computing, Communication and Electronics. IEEE**, 2020: 1–7.

Dror, B., Yanai E., Frid A., Peleg N., Goldenthal N., Schlesinger I., Hel-Or H., & Raz S. "*Automatic Assessment of Parkinson's Disease from Natural Hands Movements Using 3D Depth Sensor*." In **2014 IEEE 28th Convention of Electrical and Electronics Engineers in Israel, IEEEI 2014. IEEE**, 2014: 1- 5.

Ghosh S, Dasgupta A & A. Swetapadma, "*A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification*," ***2019 International Conference on Intelligent Sustainable Systems (ICISS)***, Palladam, India, doi: 10.1109/ISS1.2019.8908018. 2019, pp. 24-28.

Hidayat, A. S., Ramdani F., & Bachtiar F. "*Detection and Classification of Embung Land Cover Using Support Vector Machine*." In **ACM International Conference Proceeding Series**, 2021:179-183.

Hou Y., Chen J. J & Wang Z. "*Multi-Branch Network with Ensemble Learning for Text Removal in the Wild*"**. Proceedings of the Asian Conference on Computer Vision (ACCV)**, 2022, pp. 1333-1349

Huang M, Liu Y, Peng Z, Liu C, Lin D, Zhu, Yuan N, Ding K & Jin L. *SwinTextSpotter: Scene Text Spotting via Better Synergy Between Text Detection and Text Recognition*. **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),** 2022, pp. 4593-4603

Khalid, S., Khalil T., & Nasreen S. "*A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning*." In **Proceedings of 2014 Science and Information Conference, SAI 2014, 372–378. IEEE**, 2014.

Liao, M, Zhaoyi W, Cong Y, Kai C, & Xiang B. "*Real-Time Scene Text Detection With Differentiable Binarization*". **Proceedings of the AAAI Conference on Artificial Intelligence 34,** no. 07 April 3, 2020: 11474-11481. Accessed January 31, 2023.

Ma T., Du X., Wang Y & Cui X. *"Scene Text Recognition with Heuristic Local Attention*," *2022* ***IEEE International Conference on Big Data (Big Data),*** Osaka, Japan, doi: 10.1109/BigData55660.2022.10020269, 2022, pp. 4187-4194.

Murinto, & Agus Harjoko. "*Dataset Feature Reduction Using Independent Component Analysis with Contrast Function of Particle Swarm Optimization on Hyperspectral Image Classification.*" In **Proceeding - 2016 2nd International Conference on Science in Information Technology, ICSITech 2016: Information Science for Green Society and Environment**. IEEE, 2017: 285–290.

Navada, A., Ansari A. N., Patil S., & Sonkamble B. A. "*Overview of Use of Decision Tree*

Do Not Copy, Lead City University, Nigeria

Algorithms in Machine Learning." In **Proceedings - 2011 IEEE Control and System Graduate Research Colloquium, ICSGRC 2011**. IEEE, 2011: 37-42.

Pang, J., Sun W., Ren J. S. J, Yang C., & Yan Q. "*Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching."* In **Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW** *2017*, 2018-January:878–886. IEEE, 2017.

Raisi Z., Younes G & Zelek J. *"Arbitrary Shape Text Detection using Transformers," 2022* ***26th International Conference on Pattern Recognition (ICPR),*** Montreal, QC, Canada, doi: 10.1109/ICPR56361.2022.9956488, 2022, pp. 3238-3245.

Sun, Y., Ni Z., Chng C-K., Liu Y., Luo C., Ng C. C, Han J, Ding E, Liu J, Karatzas D, Chan C. S & Jin L. "*ICDAR 2019 Competition on Large-Scale Street View Text with Partial Labeling-RRC-LSVT*," **Proceedings of the International Conference on Document Analysis and Recognition, ICDAR** 2019: 1557–1562.

Journal

Alajmi A. , & Wright J.. "*Selecting the Most Efficient Genetic Algorithm Sets in Solving Unconstrained Building Optimization Problem*." **International Journal of Sustainable Built Environment 3**, no. 1, June 2014: 18–26.

Alzubaidi, L., Zhang J., Humaidi A. J., Al-Dujaili A., Duan Y., Al-Shamma O., J. Santamaría, Fadhel A. M., Al-Amidie M., & Farhan L. "*Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*." **Journal of Big Data 8, no. 1** December 2021: 53.

Anagnostis A., Tagarakis A. C., Kateris D., Moysiadis V., Sørensen C. G., Pearson S., & Bochtis D. "*Orchard Mapping with Deep Learning Semantic Segmentation."* **Sensors 21, no. 11** May 2021: 3813.

Anowar F., Sadaoui S., & Selim B. "*Conceptual and Empirical Comparison of Dimensionality Reduction Algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE).*" **Computer Science Review,** 2021. Accessed November 4, 2022.

Ansari, G. J., Shah J. H., Farias M. C. Q., Sharif M., Qadeer N., & Khan H. U. "*An Optimized Feature Selection Technique in Diversified Natural Scene Text for Classification Using Genetic Algorithm*." **IEEE Access 9,** April 2021: 54923–54937.

Armi L, & Fekri-Ershad S.. "*Texture Image Analysis and Texture Classification Methods - A Review*" 2019.

Ayaz, M., Shaukat F., & Raja G. "*Ensemble Learning Based Automatic Detection of Tuberculosis in Chest X-Ray Images Using Hybrid Feature Descriptors*." **Physical and**

Do Not Copy, Lead City University, Nigeria

Engineering Sciences in Medicine 44, no. 1, March 2021: 183–194.

Ayesha, S., Hanif M. K., & Talib R. "*Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data*." **Information Fusion 59**, July 2020: 44–58.

Bansal M, Goyal A & Choudhary A. '*A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning,'* **Decision Analytics Journal**, Volume 3, 100071, ISSN 2772-6622, 2022.

Bartenhagen, C., Klein H. U., Ruckert C., Jiang X., & Dugas M. "*Comparative Study of Unsupervised Dimension Reduction Techniques for the Visualization of Microarray Gene Expression Data*." **BMC Bioinformatics 11**, no. 1 December 2010: 567.

Beddiar, D. R., Nini B., Sabokrou M., & Hadid A. "*Vision-Based Human Activity Recognition: A Survey.*" **Multimedia Tools and Applications 79**, no. 41–42, November 2020: 30509–30555.

Bera, A. & Sychel D. "*Features Extraction for Detection of Blurred Image Regions."* **Applied Artificial Intelligence** 30, no. 3, March 2016: 201–215.

Borisov, V., Leemann T, Seßler K, Haug J, Pawelczyk M & Kasneci, G. "*Deep Neural Networks and Tabular Data: A Survey,*" ***in IEEE Transactions on Neural Networks and Learning Systems***, doi: 10.1109/TNNLS.2022.3229161.

Boukthir K., Qahtani A. M, Almutiry O, Dhahri H & Alimi A. M. "*Reduced annotation based on deep active learning for arabic text detection in natural scene images",* **Pattern Recognition Letters**, ISSN 0167-8655, Volume 157, 2022, Pages 42-48.

Braun, C. E., Chiwiacowsky L. D., & Gómez A. T. "*Variations of Ant Colony Optimization for the Solution of the Structural Damage Identification Problem*." **Procedia Computer Science 51,** no. 1 2015: 875–884.

Butt, H., Raza M. R., Ramzan M. J., Ali M. J., & Haris M. "*Attention-Based CNN-RNN Arabic Text Recognition from Natural Scene Images.*" **Forecasting 3**, no. 3 July 2021: 520–540.

Cao, D., Zhong Y., Wang L., He Y., & Dang J. "*Scene Text Detection in Natural Images: A Review*." ***Symmetry* 12,** no. 12 November 2020: 1–26. .

Cao, Q., Lei La, Hongxia Liu, & Si Han. "*Mixed Weighted KNN for Imbalanced Datasets*." **International Journal of Performability Engineering 14**, no. 7 2018: 1391–1400.

Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L & Lopez A.'*A comprehensive survey on support vector machine classification: Applications, challenges and trends*," **Neurocomputing**, Volume 408, ISSN 0925-2312, 2020, Pages 189-215.

Do Not Copy, Lead City University, Nigeria

Cha, D., Pae C., Seong S. B, Choi J. Y., & Jeong Park H. J. "*Automated Diagnosis of Ear Disease Using Ensemble Deep Learning with a Big Otoendoscopy Image Database*." **EBioMedicine** 45 2019: 606–614.

von Chamier, L., Laine R. F., Jukkala J., Spahn C., Krentzel D., Nehme E., & Lerche M. "*Democratising Deep Learning for Microscopy with ZeroCostDL4Mic*." **Nature Communications 12**, no. 1 December 2021: 2276.

Chang, W., Liu Y., Xiao Y., Yuan X., Xu X., Zhang S., & Zhou S.. "*A Machine-Learning-Based Prediction Method for Hypertension Outcomes Based on Medical Data*." **Diagnostics 9**, no. 4 November 2019: 178.

Charbuty, B., & Abdulazeez A. "*Classification Based on Decision Tree Algorithm for Machine Learning*." **Journal of Applied Science and Technology Trends 2**, no. 01 2021: 20–28.

Chen, C., Qin C., Qiu H., Tarroni G., Duan J., Bai W., & Rueckert D. "*Deep Learning for Cardiac Image Segmentation: A Review*." **Frontiers in Cardiovascular Medicine**, March 2020.

Chen, C., & Xie K. "*Face Recognition Based on Two-Dimensional Principal Component Analysis and Kernel Principal Component Analysis*." **Information Technology Journal 11**, no. 12, 2012: 1781–1785.

Chen, R. C., Dewi C., Huang S. W., & Caraka R. E. "*Selecting Critical Features for Data Classification Based on Machine Learning Methods*." **Journal of Big Data 7**, no. , December 2020: 52.

Chen,Y., Tao J., Zhang Q., Yang K., Chen X., Xiong J., Xia R., & Xie J. "*Saliency Detection via the Improved Hierarchical Principal Component Analysis Method*." **Wireless Communications and Mobile Computing** May 2020: 1–12.

Chen, J. & Lian, Z. TextPolar: *"Irregular scene text detection using polar representation*". **IJDAR 24**, 2021:315–323.

Clark, J., & Provost F. "*Unsupervised Dimensionality Reduction versus Supervised Regularization for Classification from Sparse Data*." **Data Mining and Knowledge Discovery 33**, no. 4, July 2019: 871–916.

Crawford, B., Soto R.,Cuesta R., & Paredes F. "*Application of the Artificial Bee Colony Algorithm for Solving the Set Covering Problem*." **The Scientific World Journal** 2014: 1–8.

Dhar, A., Mukherjee, H., Dash, N.S. & Roy K. "*Text categorization: past and present*". **Artif Intell Rev 54**, 3007–3054 2021.

Dietterich D. T. G. "*Ensemble Methods in Machine Learning*.**" Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)** 1857 LNCS 2000: 1–15.

Do Not Copy Lead City University, Nigeria

Dong, H., Li T., Ding R., & Sun J. "*A Novel Hybrid Genetic Algorithm with Granular Information for Feature Selection and Optimization.*" **Applied Soft Computing Journal 65**, April 2018: 33–46.

Elssied, N. O. F., Ibrahim O., & Osman A. H. "*A Novel Feature Selection Based on One-Way ANOVA F-Test for e-Mail Spam Classification*." **Research Journal of Applied Sciences, Engineering and Technology 7,** no. 3, January 2014: 625–638.

Fang S., Mao Z., Xie H, Wang Y, Yan C & Zhang Y. "*ABINet++: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Spotting,*" in ***IEEE Transactions on Pattern Analysis and Machine Intelligence,*** doi: 10.1109/TPAMI.2022.3223908.

Fantin I, Raj, E., & Balaji M. "*Application of Deep Learning and Machine Learning in Pattern Recognition*.", 2022:63-89.

Francis, L. M., & Sreenath N. "*Robust Scene Text Recognition: Using Manifold Regularized Twin-Support Vector Machine.*" **Journal of King Saud University - Computer and Information Sciences 34**, no. 3, February 2022: 589–604.

Freitas, D, Lopes L. G., & Morgado-Dias F. "*Particle Swarm Optimisation: A Historical Review up to the Current Developments*." **Entropy 22**, no. 3, March 2020: 362.

Ge, Z., Yang C., & Song Z. "*Improved Kernel PCA-Based Monitoring Approach for Nonlinear Processes.*" **Chemical Engineering Science 64**, no. 9, May 2009: 2245–2255.

Golugula, A., Lee G., Master S. R., Feldman M. D., Tomaszewski J. E., Speicher D. W., & Madabhushi A. "*Supervised Regularized Canonical Correlation Analysis: Integrating Histologic and Proteomic Measurements for Predicting Biochemical Recurrence Following Prostate Surgery*." **BMC Bioinformatics 12**, no. 1, December 2011: 483.

Hamad, K., & Kaya M. "*A Detailed Analysis of Optical Character Recognition Technology*." **International Journal of Applied Mathematics, Electronics and Computers 4,** no. Special Issue-1, December 2016: 244–244.

Harimoorthy, K., & Thangavelu M. "*Multi-Disease Prediction Model Using Improved SVM-Radial Bias Technique in Healthcare Monitoring System*." **Journal of Ambient Intelligence and Humanized Computing 12**, no. 3, March 2021: 3715–3723.

Hassanat, A, Almohammadi K., Alkafaween E., Abunawas E., Hammouri A., & Prasath V. B. S. "*Choosing Mutation and Crossover Ratios for Genetic Algorithms-a Review with a New Dynamic Approach*." **Information (Switzerland) 10**, no. 12, December 2019: 390.

He, Z. A., Ma C., Wang X., Li L., Wang Y., Zhao Y., & Guo H. "*A Modified Artificial Bee Colony Algorithm Based on Search Space Division and Disruptive Selection Strategy*." **Mathematical Problems in Engineering** 2014: 1–14.

Hira, Z. M., & Gillies D. F. "*A Review of Feature Selection and Feature Extraction Methods*

Do Not Copy, Lead City University, Nigeria

Applied on Microarray Data." **Advances in Bioinformatics** June 2015: 1–13.

Honeine, P. "*Online Kernel Principal Component Analysis: A Reduced-Order Model*." **IEEE Transactions on Pattern Analysis and Machine Intelligence 34**, no. 9, September 2012: 1814–1826.

Huang, Y., & Li L. "*Naive Bayes Classification Algorithm Based on Small Sample Set*." In **CCIS2011 - Proceedings: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems**. IEEE, 2011:34-39.

Hutter, F., Xu L., Hoos H. H., & Leyton-Brown K. "*Algorithm Runtime Prediction: Methods & Evaluation*." **Artificial Intelligence 206**, no. 1, January 2014: 79–111.

Hyvärinen, A. "*Independent Component Analysis: Recent Advances*." **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371**, no. 1984 February 2013: 20110534.

Ibrayim, M., Yuan L., & Askar H. *"Scene Text Detection Based on Two-Branch Feature Extraction"* ***Sensors 22***, no. 16: 6262. 2022.

Jia, W., Sun M., Lian J., & Hou S. "*Feature Dimensionality Reduction: A Review.*" **Complex and Intelligent Systems 8**, no. 3 January 2022: 2663–2693.

Johnstone, I. M., & Lu A. Y. "*On Consistency and Sparsity for Principal Components Analysis in High Dimensions.*" **Journal of the American Statistical Association 104**, no. 486 June 2009: 682–693.

Jollife, I. T., & Cadima J. "*Principal Component Analysis: A Review and Recent Developments.*" **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374,** no. 2065, April 2016: 2065.

Kanak M, Madhushi V, Gaurav S & Suyel NHassanat. *QEST: Quantized and Efficient Scene Text Detector using Deep Learning*. **ACM Trans. Asian Low-Resour. Lang. Inf. Process.** Just Accepted, March 2022.

Katoch, S., Chauhan S. S., & Kumar V. *A Review on Genetic Algorithm: Past, Present, and Future. Multimedia Tools and Applications*. **Vol. 80. Multimedia Tools and Applications**, 2021.

Kavzoglu, T., and Colkesen I. "*A Kernel Functions Analysis for Support Vector Machines for Land Cover Classification*." **International Journal of Applied Earth Observation and Geoinformation** 11, no. 5, October 2009: 352–359.

Keijsers, N. L.W. "*Neural Networks*." In **Encyclopedia of Movement Disorders**, 257–259. Elsevier, 2010.

Khan, T., Sarkar, R. & Mollah, A.F. *"Deep learning approaches to scene text detection: a comprehensive review"*. **Artif Intell Rev 54**, 2021:3239–3298.

Do Not Copy, Lead City University, Nigeria

Khurma, R. A., Aljarah I., Sharieh A., Elaziz M. A., Damaševičius R., & Krilavičius T. "*A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem*." **Mathematics 10**, no. 3 January 2022: 464.

Kim, C. M., Hong E. J., Chung K., & Park R. C. "*Line-Segment Feature Analysis Algorithm Using Input Dimensionality Reduction for Handwritten Text Recognition*." **Applied Sciences (Switzerland) 10**, no. 19, October 2020: 1–17.

Koo, H. Il. "*Text-Line Detection in Camera-Captured Document Images Using the State Estimation of Connected Components*." **IEEE Transactions on Image Processing 25**, no. 11, November 2016: 5358–5368.

Kotsiantis, S. B. "*Decision Trees: A Recent Overview.*" **Artificial Intelligence Review 39,** no. 4, April 2013: 261–283.

Kotsiantis, S. B., Zaharakis I. D., & Pintelas P. E. "*Machine Learning: A Review of Classification and Combining Techniques.*" **Artificial Intelligence Review 26**, no. 3 November 2006: 159–190.

Kumar,K. K., Chaduvula K., & Markapudi B. R. "*A Detailed Survey On Feature Extraction Techniques In Image Processing For Medical Image Analysis*." **European Journal of Molecular & Clinical Medicine 7**, no. 10, 2021: 2275–2284.

Leardi, R. "*Genetic Algorithms*." In **Comprehensive Chemometrics**, 1:631–653. Boston: Springer, 2009.

Li, Y. Silamu, W. Wang, Z.;Xu, M. *Attention-Based Scene Text Detection on Dual Feature Fusion*. ***Sensors 2022***, *22*, 9072.

Lin, W., Lian Z., Gu X., & Jiao B. "*A Local and Global Search Combined Particle Swarm Optimization Algorithm and Its Convergence Analysis*." **Mathematical Problems in Engineering** 2014: 1–11.

Liu, L., Chen J., Fieguth P., Zhao G., Chellappa R., & Pietikäinen M. "*From BoW to CNN: Two Decades of Texture Representation for Texture Classification.*" **International Journal of Computer Vision 127**, no. 1, January 2019: 74–109.

Liu, L., Ouyang W., Wang X., Fieguth P., Chen J., Liu X., & Pietikäinen M. "*Deep Learning for Generic Object Detection: A Survey*." **International Journal of Computer Vision 128**, no. 2, February 2020: 261–318.

Liu, X., Song L., Liu S., & Zhang Y. "*A Review of Deep-Learning-Based Medical Image Segmentation Methods.*" **Sustainability (Switzerland) 13**, no. 3, January 2021: 1–29.

Liu, Y., Wu J. M., Avdeev M., & Shi S. Q. "*Multi-Layer Feature Selection Incorporating Weighted Score-Based Expert Knowledge toward Modeling Materials with Targeted Properties.*" **Advanced Theory and Simulations 3**, no. 2, February 2020: 1900215.

Do Not Copy, Lead City University, Nigeria

Liu, Z., Wang, L. & Qiao, J. "*Visual and semantic ensemble for scene text recognition with gated dual mutual attention".* **Int J Multimed Info Retr 11**. doi:10.1007/s13735-022-00253-6. 2022:669-680.

Lodge, J. M., Kennedy G., Lockyer L., Arguel A., & Pachman M. "*Understanding Difficulties and Resulting Confusion in Learning: An Integrative Review*." **Frontiers in Education 3** June 2018.

Long, S., He X., and Yao C. "Scene Text Detection and Recognition: The Deep Learning Era." *International Journal of Computer Vision* 129, no. 1 doi:10.1007/s11263-020-01369-0. January 2021: 161–184.

Lu, Y., & Foster D. P. "*Large Scale Canonical Correlation Analysis with Iterative Least Squares.*" **Advances in Neural Information Processing Systems 1**, no. January 2014: 91–99.

Lundervold, A. S., & Lundervold A. "*An Overview of Deep Learning in Medical Imaging Focusing on MRI.*" **Zeitschrift Fur Medizinische Physik**, May 2019.

Van Der Maaten, L J P, Postma E. O., & Van Den Herik H. J. "*Dimensionality Reduction: A Comparative Review.*" **Journal of Machine Learning Research 10**, no. October 2016 (2009): 1–41.

Madhavan, S., & Kumar N. "*Incremental Methods in Face Recognition: A Survey*.", no. 1, January 2021: 253–303.

Mansour, Y., & Schain M. "*Learning with Maximum-Entropy Distributions*." **Machine Learning 45**, no. 2, 2001: 123–145.

Marcos-Zambrano, L. J., Karaduzovic-Hadziabdic K., Turukalo T. J., Przymus P., Trajkovik V., Aasmets O., & Berland M., "*Applications of Machine Learning in Human Microbiome Studies: A Review on Feature Selection, Biomarker Identification, Disease Prediction and Treatment*." **Frontiers in Microbiology** 12, February 2021.

Maxwell, A. E., Warner T. A., & Fang F. "*Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review*." **International Journal of Remote Sensing 39**, no. 9, May 2018: 2784–2817.

Messias, V. R., Estrella J. C., Ehlers R., Santana M. J., Santana R. C., & Reiff-Marganiec S. "*Combining Time Series Prediction Models Using Genetic Algorithm to Autoscaling Web Applications Hosted in the Cloud Infrastructure*." **Neural Computing and Applications 27**, no. 8, November 2016: 2383–2406.

Mijwel, M. M. "*Genetic Algorithm Optimization by Natural Selection."* **Computer Science,College of Science 1**, no. 1 2016: 1–6.

Mishra, P., Pandey C., Singh U., Keshri A, & Sabaretnam M. "*Selection of Appropriate*

Do Not Copy, Lead City University, Nigeria

Statistical Methods for Data Analysis." **Annals of Cardiac Anaesthesia** 22, no. 3, 2019: 297–301.

Mishra, S., Sarkar U., Taraphder S., Datta S., Swain D., Saikhom R., Panda S., & Laishram M. "*Principal Component Analysis."* **International Journal of Livestock Research :** 1, 2017.

Naiemi, F., Ghods, V. & Khalesi, H. "*Scene text detection and recognition: a survey".* **Multimed Tools Appl 81**. Doi:10.1007/s11042-022-12693-7. 2022: 20255–2029

Nordhausen, K., & Oja H. "*Independent Component Analysis: A Statistical Perspective*." **Wiley Interdisciplinary Reviews: Computational Statistics 10**, no. 5, September 2018.

Novitasari, H. B., Sfenrianto H. N., Rahmawati A., Prasetyo R., Miharja, & Gata W. "*K-Nearest Neighbor Analysis to Predict the Accuracy of Product Delivery Using Administration of Raw Material Model in the Cosmetic Industry (PT Cedefindo)*." **Journal of Physics: Conference Series 1367, no. 1** November 2019: 012008.

Ogban, F. U., & Nentui R. "*Pheromone Deposition/Updating Strategy in a Network: Using Ant Colony Optimization (ACO) Approach*." **Global Journal of Pure and Applied Sciences 24**, no. 2, December 2019: 215–222.

Omuya E. O, Okeyo G. O, & Kimwele M. W. '*Feature Selection for Classification using Principal Component Analysis and Information Gain'*, **Expert Systems with Applications, Volume 174**, 114765, ISSN 0957-4174, doi:10.1016/j.eswa.2021.114765. 2021.

Pandey, B. K., Pandey D., Wariya S., Aggarwal G., & Rastogi R. "*Deep Learning and Particle Swarm Optimisation-Based Techniques for Visually Impaired Humans' Text Recognition and Identification*." **Augmented Human Research 6**, no. 1, December 2021: 14.

Prasad, V., & Jayanta Y. "*(PDF) A Study on Method of Feature Extraction for Handwritten Character Recognition*." **Indian Journal of Science and Technology 6,** no. S3 2013: 174–178.

Qin, S. & Chen, L. *"Arbitrary-shaped scene text detection with keypoint-based shape representation*". **IJDAR 25**, 115–127 (2022). https://doi.org/10.1007/s10032-022-00396-6

Rainarli E, Suprapto & Wahyono. "*A decade: Review of scene text detection methods*", **Computer Science Review**, Volume 42, 100434, ISSN 1574-0137. doi:10.1016/j.cosrev.2021.100434. 2021.

Rawat, W., & Wang Z. "*Deep Convolutional Neural Networks for Image Classification: A*

Do Not Copy, Lead City University, Nigeria

Comprehensive Review." **Neural Computation**, September 2017.

Raju K, Rao Y. S, & Yadav M. N. "*Performance Analysis of PCA and LDA."* **International Journal of Innovative Research in Electronics and Communications 2**, no. 2 2015: 17–22.

Rizwan I H, H, & Neubert J. "*Deep Learning Approaches to Biomedical Image Segmentation*." **Informatics in Medicine Unlocked,** 2020.

Rosipal, R., Trejo L. J., & Cichocki A. "*Kernel Principal Component Regression with EM Approach to Nonlinear Principal Components Extraction*." **Computing and Information Systems Technical Reports 12,** no. December 2000: 1–42.

Sanlı, T., Sıcakyüz C., & Yüregir O. H. "*Comparison of the Accuracy of Classification Algorithms on Three Data-Sets in Data Mining: Example of 20 Classes*." **International Journal of Engineering, Science and Technology 12**, no. 3, 2020: 81–89.

Santhanam, G., Yu B. M, Gilja V., Ryu S. I., Afshar A., Sahani M., & Shenoy K. V. "*Factor-Analysis Methods for Higher-Performance Neural Prostheses*." **Journal of Neurophysiology 102**, no. 2, August 2009: 1315–1330.

Sarkar, R., Malakar S., Das N., Basu S., Kundu M., & Nasipuri M. "*Word Extraction and Character Segmentation from Text Lines of Unconstrained Handwritten Bangla Document Images."* **Journal of Intelligent Systems 20**, no. 3, January 2011: 227–260.

Sarker, I. H. "*Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions*." **SN Computer Science 2**, no. 6, November 2021: 420.

Sarker, I. H. "*Machine Learning: Algorithms, Real-World Applications and Research Directions*." **SN Computer Science 2**, no. 3, May 2021: 160.

Sarker, I. H., Kayes A. S.M., & Watters P. "*Effectiveness Analysis of Machine Learning Classification Models for Predicting Personalized Context-Aware Smartphone Usage*." **Journal of Big Data 6**, no. 1, December 2019: 57.

Sayah, S., & Hamouda A. "*A Hybrid Differential Evolution Algorithm Based on Particle Swarm Optimization for Nonconvex Economic Dispatch Problems*." **Applied Soft Computing Journal 13**, no. 4, April 2013: 1608–1619.

Schmidt, J., Marques M. R. G., Botti S., & Marques M. A. L. "*Recent Advances and Applications of Machine Learning in Solid-State Materials Science.*" ***npj Computational Materials 5***, no. 1, December 2019: 83.

Sevinc, E.. "*A Novel Evolutionary Algorithm for Data Classification Problem with Extreme Learning Machines*." 2019: 122419–122427.

Do Not Copy, Lead City University, Nigeria

Shah, K., Patel, H., Sanghvi, D. & Shah M.*" A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification*". **Augment Hum Res 5**, 12. doi:10.1007/s41133-020-00032-0. 2020.

Sharma, H., Bansal J. C., Arya K. V., & Yang X. S. "*Lévy Flight Artificial Bee Colony Algorithm.*" **International Journal of Systems Science 47**, no. 11, August 2016: 2652–2670.

Wu, X., Tang, B., Zhao, M., Wang J. & Guo Y. "*STR Transformer: A Cross-domain Transformer for Scene Text Recognition*". **Appl Intell 53**, 2023. doi:10.1007/s10489-022-03728-5. 2023: 3444–3458.

Wu, Q., Luo, W., Chai, Z., Guom G. "*Scene text detection by adaptive feature selection with text scale-aware loss".* **Appl Intell 52**. doi:10.1007/s10489-021-02331-4. 2022: 514–529.

Xin, M & Wang, Y. "*Research on image classification model based on deep convolution neural network".* **J Image Video Proc**. **2019**, 40. doi:10.1186/s13640-019-0417-8. 2019.

Zhang, S, Caiying Z, Yonggang L, Xianchao Z, Lihua Y, & Yuanwang W. *"Irregular Scene Text Detection Based on a Graph Convolutional Network*" ***Sensors 23,*** no. 3: 1070. doi:10.3390/s23031070. 2023.

Zhong Y., Cheng X., Chen T., JZhang J., Zhou Z. & Guan Huang, "*PRPN: Progressive region prediction network for natural scene text detection*", **Knowledge-Based Systems,** Volume 236, 107767, ISSN 0950-7051, doi:10.1016/j.knosys.2021.107767, 2022.

Zhu J & Wang G. "*TransText: Improving scene text detection via transformer",* **Digital Signal Processing**, Volume 130, 103698, ISSN 1051-2004, doi:10.1016/j.dsp.2022.103698. 2022.

Electronic Sources (Internet)

Hassanat, A. B., Abbadi M. A, Altarawneh G. A, & Alhasanat A. A. "*Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach*" 2014. http://arxiv.org/abs/1409.0919.

Naik, G. R. "*An Overview of Independent Component Analysis and Its Applications."* **Informatica (Ljubljana) 35**, no. 1 2011: 63–81.

Zou, Z., Shi Z., Guo Y., & Ye J. "*Object Detection in 20 Years: A Survey*" May 2019. http://arxiv.org/abs/1905.05055.

Thesis

Mishra, A. "Understanding Text in Scene Images" 200907004, no. May 2014.

Do Not Copy, Lead City University, Nigeria

Appendix A

Source Code

```
!       pip install tqdm update_checker tqdm

!       pip install tpot

import matplotlib

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import seaborn as sns

import sklearn

import imblearn

from sklearn.preprocessing import StandardScaler, MinMaxScaler

from sklearn.preprocessing import LabelEncoder, MultiLabelBinarizer, LabelBinarizer from

sklearn.ensemble import RandomForestClassifier from sklearn.feature_selection import RFE

import itertools

from sklearn.model_selection import train_test_split

from tpot import TPOTRegressor

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,

roc_auc_score, roc_curve

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.manifold import LocallyLinearEmbedding

from sklearn.decomposition import FastICA

from sklearn.svm import SVC

from sklearn.cross_decomposition import PLSRegression
```

Do Not Copy, Lead City University, Nigeria

```
from sklearn.manifold import MDS
import os
from PIL import Image
from PIL import UnidentifiedImageError
import keras
import matplotlib.image as mpimg
from matplotlib.image import imread
import cv2
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (4.64.1)0
Requirement already satisfied: update_checker in /usr/local/lib/python3.7/dist-packages (0.18.0)0
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update_checker) (2.23.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from
https://colab.research.google.com/drive/1jCAkf7NzuylBpVN5j-dxEbH32lBaunUK#printMode=true
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update_che
```

Do Not Copy, Lead City University, Nigeria

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0

Requirement already satisfied: tpot in /usr/local/lib/python3.7/dist-packages (0.11.7)0

Requirement already satisfied: update-checker>=0.16 in /usr/local/lib/python3.7/dist-packages (from tpot) (0.18.0)0

Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.0.2)0

Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (4.64.1)0

Requirement already satisfied: deap>=1.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.3)0

Requirement already satisfied: numpy>=1.16.3 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.21.6)0

Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.5)0

Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.2.0)0

Requirement already satisfied: xgboost>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.6.2)0

Requirement already satisfied: stopit>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.1.2)0

Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.7.3)0

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->tpot) (2022

Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->t

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.

Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update-checker>=0.16->tp

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update-che

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

```
train_csv = pd.read_csv("/content/train.csv")0
test_csv = pd.read_csv("/content/test_WyRytb0.csv")0
print(train_csv.shape,test_csv.shape)0
x = train_csv.iloc[:, :-1]0
y = train_csv.iloc[:, -1]0
```

(17034, 2) (7301, 1)0

Do Not Copy, Lead City University, Nigeria

```
#       Create list to store the data and set the path of the image to load0 data_with_labels =
[]0
labels = []0 data_test = []0
imagePath = '/content/drive/MyDrive/blurredd/blurry/*.jpg'0
#       Create the training dataset 0 for i in train_csv.index: 0
nameOfFile = train_csv['image_name'][i]0 if os.path.exists(imagePath+nameOfFile):0
image = mpimg.imread(imagePath+nameOfFile)0
if (len(image.shape)!=3): # Verify if the image is correct 0 print("L'image N°",i,' :
',nameOfFile,"")0
else : 0
image       =       cv2.resize(image,(150,150))0       data_with_labels.append(image)0
labels.append(train_csv['label'][i])0
print(len(data_with_labels))0 print(len(labels))0
00
00
import glob0
glob.glob(imagePath)0
imagess = [cv2.imread(images) for images in glob.glob(imagePath)]0
type(imagess)
rows = 2
cols = 3
fig = plt.figure(figsize=(20,10))
for j in range(0, rows*cols):
fig.add_subplot(rows, cols, j+1)
plt.imshow(imagess[j])
```

Do Not Copy, Lead City University, Nigeria

scaler = StandardScaler()0

x_sc = scaler.fit_transform(x)0

encoder = MultiLabelBinarizer()0

x_sc_l = encoder.fit_transform(x_sc)0

tpot = TPOTRegressor(generations=10, population_size=5, verbosity=2)0 tpot.fit(x_sc, y)0

Generation 1 - Current best internal CV score: -0.225067930342253670

Generation 2 - Current best internal CV score: -0.225067930342253670

Generation 3 - Current best internal CV score: -0.225067930342253670

Generation 4 - Current best internal CV score: -0.22316440249533070

Generation 5 - Current best internal CV score: -0.207305754703248270

Generation 6 - Current best internal CV score: -0.207305754703248270

Generation 7 - Current best internal CV score: -0.207305754703248270

Generation 8 - Current best internal CV score: -0.207305754703248270

Generation 9 - Current best internal CV score: -0.207305754703248270

Generation 10 - Current best internal CV score: -0.20722149379028780

X_train,XBest_test,Ypipeline:_train,YXGBRegressor(RobustScaler(input_test=train_test_spl

it(x_sc,_matrix),y,trainlearning_size=0.7,_rate=0.01,random_state=2)max_de

input_shape = [X_train.shape[1]]

model_svm = SVC(random_state=1)

model_svm.fit(X_train, Y_train)

y_pred_svm = model_svm.predict(X_test)

print(classification_report(Y_test, y_pred_svm))

	precision	recall	f1-score	support0
0	0.72	0.86	0.79	7310
1	0.88	0.76	0.82	10100

Do Not Copy, Lead City University, Nigeria

accuracy 0.80 17410

macro avg 0.80 0.81 0.80 17410

weighted avg 0.82 0.80 0.80 17410

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Reds):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

Do Not Copy, Lead City University, Nigeria

```
cm = confusion_matrix(Y_test, y_pred_svm)0
target_names = ["True", "False"]0
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0
accuracy_score(Y_test, y_pred_svm)0
0.8024124066628374
# plot for SVM0
false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_svm)0
print('roc_auc_score for SVM: ', roc_auc_score(Y_test, y_pred_svm))0
plt.subplots(1, figsize=(10,10))0
plt.title('Receiver Operating Characteristic - SVM')0
plt.plot(false_positive_rate1, true_positive_rate1)0
plt.plot([0, 1], ls="--")0
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0
plt.ylabel('True Positive Rate')0
plt.xlabel('False Positive Rate')0
plt.show()0
roc_auc_score for SVM:  0.81080846798770170
#KNN model0
model_KNN = KNeighborsClassifier()0
model_KNN.fit(X_train, Y_train)0
y_pred_KNN = model_KNN.predict(X_test)0
print(classification_report(Y_test, y_pred_KNN))0
              precision    recall  f1-score   support0

           0       0.73      0.87      0.80      7310

           1       0.89      0.77      0.83     10100
```


Do Not Copy, Lead City University, Nigeria

accuracy 0.81 17410

macro avg 0.81 0.82 0.81 17410

weighted avg 0.83 0.81 0.82 17410

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

Do Not Copy, Lead City University, Nigeria

```
cm = confusion_matrix(Y_test, y_pred_KNN)0
target_names = ["True", "False"]0
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0
accuracy_score(Y_test, y_pred_KNN)0
0.8139000574382539
# plot for KNN0
false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_KNN)0
print('roc_auc_score for KNN: ', roc_auc_score(Y_test, y_pred_KNN))0
plt.subplots(1, figsize=(10,10))0
plt.title('Receiver Operating Characteristic - SVM')0
plt.plot(false_positive_rate1, true_positive_rate1)0
plt.plot([0, 1], ls="--")0
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0
plt.ylabel('True Positive Rate')0
plt.xlabel('False Positive Rate')0
plt.show()0
roc_auc_score for KNN:  0.82203207324836460
#RandomForestClassifier(ensemble) model0
model_RFC = RandomForestClassifier(random_state = 1)0
model_RFC.fit(X_train, Y_train)0
y_pred_RFC = model_RFC.predict(X_test)0
print(classification_report(Y_test, y_pred_RFC))0
              precision    recall  f1-score   support0

           0       0.76      0.90      0.82      7310

           1       0.91      0.79      0.85     10100
```


Do Not Copy, Lead City University, Nigeria

```
accuracy                           0.84     17410
macro avg       0.84      0.85     0.84     17410
weighted avg    0.85      0.84     0.84     17410
```

==========CONSTRUCTING THE CONFUSION MATRIX==========#

```
def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):
plt.figure(figsize=(10,10))
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()
tick_marks = np.arange(len(classes))
plt.xticks(tick_marks, classes, rotation=45)
plt.yticks(tick_marks, classes)
if normalize:
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
cm = np.around(cm, decimals=2)
cm[np.isnan(cm)] = 0.0
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
plt.text(j, i, cm[i, j],
horizontalalignment="center",
color="white" if cm[i, j] > thresh else "black")
plt.tight_layout()
plt.ylabel('True label')
```

Do Not Copy, Lead City University, Nigeria

```
plt.xlabel('Predicted label')

cm = confusion_matrix(Y_test, y_pred_RFC)

target_names = ["True", "False"]

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')

accuracy_score(Y_test, y_pred_RFC)

0.8368753589890867

type(imagesss)0

rows = 20

cols = 30

fig = plt.figure(figsize=(20,10))0

for j in range(0, rows*cols):0

fig.add_subplot(rows, cols, j+1)0

plt.imshow(imagesss[j])0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_RFC)0

print('roc_auc_score for RFC: ', roc_auc_score(Y_test, y_pred_RFC))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - RFC')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for RFC:          0.845046118838970

Colab paid products0 -0 Cancel contracts here
```

Do Not Copy, Lead City University, Nigeria

```
!       pip install tqdm update_checker tqdm
!       pip install tpot
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
import imblearn
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.preprocessing import LabelEncoder, MultiLabelBinarizer, LabelBinarizer from
sklearn.ensemble import RandomForestClassifier from sklearn.feature_selection import RFE
import itertools
from sklearn.model_selection import train_test_split
from tpot import TPOTRegressor
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,
roc_auc_score, roc_curve
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.manifold import LocallyLinearEmbedding
from sklearn.decomposition import FastICA
from sklearn.svm import SVC
from sklearn.cross_decomposition import PLSRegression
from sklearn.manifold import MDS
import os
```

Do Not Copy, Lead City University, Nigeria

```
from PIL import Image
from PIL import UnidentifiedImageError
import keras
import matplotlib.image as mpimg
from matplotlib.image import imread
import cv2
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (4.64.1)0
Requirement already satisfied: update_checker in /usr/local/lib/python3.7/dist-packages (0.18.0)0
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update_checker) (2.23.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update_che
    Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0
    Requirement already satisfied: tpot in /usr/local/lib/python3.7/dist-packages (0.11.7)0
```

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: deap>=1.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.3)0

Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.7.3)0

Requirement already satisfied: stopit>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.1.2)0

Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.5)0

Requirement already satisfied: numpy>=1.16.3 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.21.6)0

Requirement already satisfied: xgboost>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.6.2)0

Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.2.0)0

Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.0.2)0

Requirement already satisfied: update-checker>=0.16 in /usr/local/lib/python3.7/dist-packages (from tpot) (0.18.0)0

Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (4.64.1)0

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->tpot) (2022

Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->t

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.

Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update-checker>=0.16->tp

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update-che

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

train_csv = pd.read_csv("/content/train.csv")0

test_csv = pd.read_csv("/content/test_WyRytb0.csv")0

print(train_csv.shape,test_csv.shape)0

x = train_csv.iloc[:, :-1]0

y = train_csv.iloc[:, -1]0

(17034, 2) (7301, 1)0

Create list to store the data and set the path of the image to load0 data_with_labels = []0

labels = []0 data_test = []0

imagePath = '/content/drive/MyDrive/blurredd/blurry/*.jpg'0

Create the training dataset 0 for i in train_csv.index: 0

Do Not Copy, Lead City University, Nigeria

nameOfFile = train_csv['image_name'][i]0 if os.path.exists(imagePath+nameOfFile):0

image = mpimg.imread(imagePath+nameOfFile)0

if (len(image.shape)!=3): # Verify if the image is correct 0 print("L'image N°",i,' : ',nameOfFile,"")0

else : 0

image = cv2.resize(image,(150,150))0 data_with_labels.append(image)0

labels.append(train_csv['label'][i])0

print(len(data_with_labels))0 print(len(labels))0

00

00

import glob0

glob.glob(imagePath)0

imagess = [cv2.imread(images) for images in glob.glob(imagePath)]0

type(imagess)0

rows = 20

cols = 30

fig = plt.figure(figsize=(20,10))0

for j in range(0, rows*cols):0

fig.add_subplot(rows, cols, j+1)0

plt.imshow(imagess[j])0

scaler = StandardScaler()0

x_sc = scaler.fit_transform(x)0

#encoder = MultiLabelBinarizer()0

#x_sc_l = encoder.fit_transform(x_sc)0

#enc = LabelEncoder()0

Do Not Copy, Lead City University, Nigeria

#y_l = enc.fit_transform(y)0

tpot = TPOTRegressor(generations=5, population_size=50, verbosity=2)0 tpot.fit(x_sc, y)0

Generation 1 - Current best internal CV score: -0.213930240118888280

Generation 2 - Current best internal CV score: -0.20814092386504460

Generation 3 - Current best internal CV score: -0.20814092386504460 ica = FastICA(n_components=10, 0

Generationmax4_iter=500,-Current 0best internal CV score: -0.207771276734992240

random_state=100)0

x_sc_icaGeneration=ica.fit5_-transform(xCurrentbestsc)0internal CV score: -0.2042404493914020

Best pipeline: XGBRegressor(RobustScaler(input_matrix), learning_rate=0.01, max_de

( )

X_train,X_test,Y_train,Y_test = train_test_split(x_sc_ica, y, train_size=0.70, random_state=2)0

input_shape = [X_train.shape[1]]0

model_svm = SVC(random_state=1)0

model_svm.fit(X_train, Y_train)0

y_pred_svm = model_svm.predict(X_test)0

print(classification_report(Y_test, y_pred_svm))0

	precision	recall	f1-score	support0
0	0.74	0.91	0.82	21720
1	0.92	0.77	0.84	30500
accuracy			0.83	52220
macro avg	0.83	0.84	0.83	52220
weighted avg	0.85	0.83	0.83	52220

Do Not Copy, Lead City University, Nigeria

```
#============================CONSTRUCTING THE CONFUSION MATRIX==============================#0
def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Reds):0 0
plt.figure(figsize=(10,10))0
plt.imshow(cm, interpolation='nearest', cmap=cmap)0
plt.title(title)0
plt.colorbar()0
tick_marks = np.arange(len(classes))0
plt.xticks(tick_marks, classes, rotation=45)0
plt.yticks(tick_marks, classes)0
if normalize:0
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0
cm = np.around(cm, decimals=2)0
cm[np.isnan(cm)] = 0.00
thresh = cm.max() / 2.0
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0
horizontalalignment="center",0
color="white" if cm[i, j] > thresh else "black")0
plt.tight_layout()0
plt.ylabel('True label')0
plt.xlabel('Predicted label')0
cm = confusion_matrix(Y_test, y_pred_svm)0
target_names = ["True", "False"]0
```

Do Not Copy, Lead City University, Nigeria

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_svm)0

0.8299502106472616

plot for SVM0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_svm)0

print('roc_auc_score for SVM: ', roc_auc_score(Y_test, y_pred_svm))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - SVM')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for SVM: 0.841503939860520

#KNN model0

model_KNN = KNeighborsClassifier()0

model_KNN.fit(X_train, Y_train)0

y_pred_KNN = model_KNN.predict(X_test)0

print(classification_report(Y_test, y_pred_KNN))0

	precision	recall	f1-score	support0
0	0.74	0.86	0.80	21720
1	0.89	0.79	0.83	30500
accuracy			0.82	52220
macro avg	0.82	0.82	0.82	52220

Do Not Copy, Lead City University, Nigeria

```
weighted avg   0.83     0.82     0.82     52220

#============================CONSTRUCTING THE CONFUSION MATRIX==============================#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

cm = confusion_matrix(Y_test, y_pred_KNN)0

target_names = ["True", "False"]0
```

Do Not Copy, Lead City University, Nigeria

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_KNN)0

0.817502872462658

plot for KNN0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_KNN)0

print('roc_auc_score for KNN: ', roc_auc_score(Y_test, y_pred_KNN))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - SVM')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for KNN: 0.824022582495546800

#RandomForestClassifier(ensemble) model0

model_RFC = RandomForestClassifier(random_state = 1)0

model_RFC.fit(X_train, Y_train)0

y_pred_RFC = model_RFC.predict(X_test)0

print(classification_report(Y_test, y_pred_RFC))0

	precision	recall	f1-score	support0
0	0.77	0.88	0.82	21720
1	0.90	0.82	0.86	30500
accuracy			0.84	52220
macro avg	0.84	0.85	0.84	52220

Do Not Copy, Lead City University, Nigeria

```
weighted avg   0.85     0.84     0.84     52220

#============================CONSTRUCTING THE CONFUSION MATRIX==============================#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

cm = confusion_matrix(Y_test, y_pred_RFC)0

target_names = ["True", "False"]0
```

Do Not Copy, Lead City University, Nigeria

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_RFC)0

0.8429720413634623

type(imagesss)0

rows = 20

cols = 30

fig = plt.figure(figsize=(20,10))0

for j in range(0, rows*cols):0

fig.add_subplot(rows, cols, j+1)0

plt.imshow(imagesss[j])0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_RFC)0

print('roc_auc_score for RFC: ', roc_auc_score(Y_test, y_pred_RFC))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - RFC')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for RFC: 0.84814524650544930

Colab paid products0 -0 Cancel contracts here

! pip install tqdm update_checker tqdm

! pip install tpot

import matplotlib

Do Not Copy, Lead City University, Nigeria

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
import imblearn
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.preprocessing import LabelEncoder, MultiLabelBinarizer, LabelBinarizer from sklearn.ensemble import RandomForestClassifier from sklearn.feature_selection import RFE
import itertools
from sklearn.model_selection import train_test_split
from tpot import TPOTRegressor
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_auc_score, roc_curve
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.manifold import LocallyLinearEmbedding
from sklearn.decomposition import FastICA
from sklearn.svm import SVC
from sklearn.cross_decomposition import PLSRegression
from sklearn.manifold import MDS
import os
from PIL import Image
from PIL import UnidentifiedImageError
import keras
```

Do Not Copy, Lead City University, Nigeria

```
import matplotlib.image as mpimg
from matplotlib.image import imread
import cv2
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (4.64.1)0
Collecting update_checker0
Downloading update_checker-0.18.0-py3-none-any.whl (7.0 kB)0
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update_checker) (2.23.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update_che
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from
Installing collected packages: update-checker0
Successfully installed update-checker-0.18.00
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0 Collecting tpot0
Downloading TPOT-0.11.7-py3-none-any.whl (87 kB)0
|████████████████████████████████| 87 kB 3.8 MB/s 0
```

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.0.2)0

Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.2.0)0 Collecting deap>=1.20

Downloading deap-1.3.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.w | 139 kB 48.9 MB/s 0

Collecting xgboost>=1.1.00

Downloading xgboost-1.6.2-py3-none-manylinux2014_x86_64.whl (255.9 MB)0

| 255.9 MB 46 kB/s 0

Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.7.3)0

Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.5)0 Collecting stopit>=1.1.10

Downloading stopit-1.1.2.tar.gz (18 kB)0

Requirement already satisfied: update-checker>=0.16 in /usr/local/lib/python3.7/dist-packages (from tpot) (0.18.0)0

Requirement already satisfied: numpy>=1.16.3 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.21.6)0

Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (4.64.1)0

Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->t

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->tpot) (2022

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.

Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update-checker>=0.16->tp

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update-che

Building wheels for collected packages: stopit0

Building wheel for stopit (setup.py) ... done0

Created wheel for stopit: filename=stopit-1.1.2-py3-none-any.whl size=11956 sha256=bae18f566cf3c541e69d850475b5a019f0

Stored in directory: /root/.cache/pip/wheels/e2/d2/79/eaf81edb391e27c87f51b8ef901ecc85a5363dc96b8b8d71e30 Successfully built stopit0

Installing collected packages: xgboost, stopit, deap, tpot0

Attempting uninstall: xgboost0

Found existing installation: xgboost 0.900

Uninstalling xgboost-0.90:0

Do Not Copy, Lead City University, Nigeria

Successfully uninstalled xgboost-0.900

Successfully installed deap-1.3.3 stopit-1.1.2 tpot-0.11.7 xgboost-1.6.20

train_csv = pd.read_csv("/content/train.csv")0

test_csv = pd.read_csv("/content/test_WyRytb0.csv")0

print(train_csv.shape,test_csv.shape)0

x = train_csv.iloc[:, :-1]0

y = train_csv.iloc[:, -1]0

(17034, 2) (7301, 1)0

Create list to store the data and set the path of the image to load0 data_with_labels = []0

labels = []0 data_test = []0

imagePath = '/content/drive/MyDrive/blurredd/blurry/*.jpg'0

Create the training dataset 0

for i in train_csv.index: 0

nameOfFile = train_csv['image_name'][i]0

if os.path.exists(imagePath+nameOfFile):0

image = mpimg.imread(imagePath+nameOfFile)0

if (len(image.shape)!=3): # Verify if the image is correct 0

print("L'image N°",i,' : ',nameOfFile,"")0

else : 0

image = cv2.resize(image,(150,150))0

data_with_labels.append(image)0

labels.append(train_csv['label'][i])0

print(len(data_with_labels))0

print(len(labels))0

Do Not Copy, Lead City University, Nigeria

00

00

import glob0

glob.glob(imagePath)0

imagess = [cv2.imread(images) for images in glob.glob(imagePath)]0

type(imagess)0

rows = 20

cols = 30

fig = plt.figure(figsize=(20,10))0

for j in range(0, rows*cols):0

fig.add_subplot(rows, cols, j+1)0

plt.imshow(imagess[j])0

scaler = StandardScaler()0

x_sc = scaler.fit_transform(x)0

tpot = TPOTRegressor(generations=5, population_size=50, verbosity=2)0 tpot.fit(x_sc, y)0

Generation 1 - Current best internal CV score: -0.216750101684816180

Generation 2 - Current best internal CV score: -0.211174728356651270

Generation 3 - Current best internal CV score: -0.211174728356651270

Generation 4 - Current best internal CV score: -0.211174728356651270

Generation 5 - Current best internal CV score: -0.2100799725679030

Best pipeline: ExtraTreesRegressor(SGDRegressor(input_matrix, alpha=0.01, eta0=0.1

( )

pls = PLSRegression(n_components=10, scale=True, max_iter=500, tol=1e-06, copy=True)0

pls.fit(x_sc, y)0

PLSRegression(n_components=10)

Do Not Copy, Lead City University, Nigeria

X_train,X_test,Y_train,Y_test = train_test_split(x_sc, y, train_size=0.95, random_state=2)0

input_shape = [X_train.shape[1]]0

model_svm = SVC(random_state=1)0

model_svm.fit(X_train, Y_train)0

y_pred_svm = model_svm.predict(X_test)0

print(classification_report(Y_test, y_pred_svm))0

	precision	recall	f1-score	support0
0	0.72	0.88	0.79	3770
1	0.89	0.74	0.81	4940
accuracy			0.80	8710
macro avg	0.81	0.81	0.80	8710
weighted avg	0.82	0.80	0.80	8710

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Reds):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

Do Not Copy, Lead City University, Nigeria

```
cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

cm = confusion_matrix(Y_test, y_pred_svm)0

target_names = ["True", "False"]0

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_svm)0

0.801377726750861

# plot for SVM0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_svm)0

print('roc_auc_score for SVM: ', roc_auc_score(Y_test, y_pred_svm))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - SVM')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0
```

Do Not Copy, Lead City University, Nigeria

roc_auc_score for SVM: 0.81076364651682240

#KNN model0

model_KNN = KNeighborsClassifier()0

model_KNN.fit(X_train, Y_train)0

y_pred_KNN = model_KNN.predict(X_test)0

print(classification_report(Y_test, y_pred_KNN))0

	precision	recall	f1-score	support0
0	0.73	0.88	0.79	3770
1	0.89	0.75	0.81	4940
accuracy			0.80	8710
macro avg	0.81	0.81	0.80	8710
weighted avg	0.82	0.80	0.80	8710

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

Do Not Copy, Lead City University, Nigeria

```
cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

cm = confusion_matrix(Y_test, y_pred_KNN)0

target_names = ["True", "False"]0

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_KNN)0

0.8025258323765786

# plot for KNN0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_KNN)0

print('roc_auc_score for KNN: ', roc_auc_score(Y_test, y_pred_KNN))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - SVM')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0
```

Do Not Copy, Lead City University, Nigeria

roc_auc_score for KNN: 0.8111475638698870

#RandomForestClassifier(ensemble) model0

model_RFC = RandomForestClassifier(random_state = 1)0

model_RFC.fit(X_train, Y_train)0

y_pred_RFC = model_RFC.predict(X_test)0

print(classification_report(Y_test, y_pred_RFC))0

	precision	recall	f1-score	support0
0	0.77	0.90	0.83	3770
1	0.92	0.79	0.85	4940
accuracy			0.84	8710
macro avg	0.84	0.85	0.84	8710
weighted avg	0.85	0.84	0.84	8710

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0

0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

Do Not Copy, Lead City University, Nigeria

```
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0


color="white" if cm[i, j] > thresh else "black")0


plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

cm = confusion_matrix(Y_test, y_pred_RFC)0

target_names = ["True", "False"]0

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_RFC)0

0.8404133180252583

type(imagesss)0

rows = 20

cols = 30

fig = plt.figure(figsize=(20,10))0

for j in range(0, rows*cols):0

fig.add_subplot(rows, cols, j+1)0

plt.imshow(imagesss[j])0
```

Do Not Copy, Lead City University, Nigeria

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_RFC)0

print('roc_auc_score for RFC: ', roc_auc_score(Y_test, y_pred_RFC))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - RFC')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for RFC: 0.84800362976406540

Colab paid products0 -0 Cancel contracts here

! pip install tqdm update_checker tqdm

! pip install tpot

import matplotlib

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import seaborn as sns

import sklearn

import imblearn

from sklearn.preprocessing import StandardScaler, MinMaxScaler

from sklearn.preprocessing import LabelEncoder, MultiLabelBinarizer, LabelBinarizer from sklearn.ensemble import RandomForestClassifier from sklearn.feature_selection import RFE

import itertools

Do Not Copy, Lead City University, Nigeria

```
from sklearn.model_selection import train_test_split
from tpot import TPOTRegressor
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_auc_score, roc_curve
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.manifold import LocallyLinearEmbedding
from sklearn.decomposition import FastICA
from sklearn.svm import SVC
from sklearn.cross_decomposition import PLSRegression
from sklearn.manifold import MDS
import os
from PIL import Image
from PIL import UnidentifiedImageError
import keras
import matplotlib.image as mpimg
from matplotlib.image import imread
import cv2
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (4.64.1)0
Requirement already satisfied: update_checker in /usr/local/lib/python3.7/dist-packages (0.18.0)0
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update_checker) (2.23.0)
```

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update_che

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0

Requirement already satisfied: tpot in /usr/local/lib/python3.7/dist-packages (0.11.7)0

Requirement already satisfied: update-checker>=0.16 in /usr/local/lib/python3.7/dist-packages (from tpot) (0.18.0)0

Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.0.2)0

Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (4.64.1)0

Requirement already satisfied: deap>=1.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.3)0

Requirement already satisfied: numpy>=1.16.3 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.21.6)0

Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.5)0

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.2.0)0

Requirement already satisfied: xgboost>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.6.2)0

Requirement already satisfied: stopit>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.1.2)0

Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.7.3)0

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->tpot) (2022

Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->t

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.

Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update-checker>=0.16->tp

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update-che

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

train_csv = pd.read_csv("/content/train.csv")0

test_csv = pd.read_csv("/content/test_WyRytb0.csv")0

print(train_csv.shape,test_csv.shape)0

x = train_csv.iloc[:, :-1]0

y = train_csv.iloc[:, -1]0

(17034, 2) (7301, 1)0

Create list to store the data and set the path of the image to load0 data_with_labels = []0

labels = []0 data_test = []0

imagePath = '/content/drive/MyDrive/blurredd/blurry/*.jpg'0

Create the training dataset 0 for i in train_csv.index: 0

nameOfFile = train_csv['image_name'][i]0 if os.path.exists(imagePath+nameOfFile):0

image = mpimg.imread(imagePath+nameOfFile)0

if (len(image.shape)!=3): # Verify if the image is correct 0 print("L'image N°",i,' : ',nameOfFile,"")0

else : 0

image = cv2.resize(image,(150,150))0 data_with_labels.append(image)0

labels.append(train_csv['label'][i])0

print(len(data_with_labels))0 print(len(labels))0

00

00

import glob0

glob.glob(imagePath)0

Do Not Copy, Lead City University, Nigeria

imagess = [cv2.imread(images) for images in glob.glob(imagePath)]0

type(imagess)

rows = 2

cols = 3

fig = plt.figure(figsize=(20,10))

for j in range(0, rows*cols):

fig.add_subplot(rows, cols, j+1)

plt.imshow(imagess[j])

scaler = StandardScaler()0

x_sc = scaler.fit_transform(x)0

encoder = MultiLabelBinarizer()0

x_sc_1 = encoder.fit_transform(x_sc)0

tpot = TPOTRegressor(generations=10, population_size=5, verbosity=2)0 tpot.fit(x_sc, y)0

Generation 1 - Current best internal CV score: -0.225067930342253670

Generation 2 - Current best internal CV score: -0.225067930342253670

Generation 3 - Current best internal CV score: -0.225067930342253670

Generation 4 - Current best internal CV score: -0.22316440249533070

Generation 5 - Current best internal CV score: -0.207305754703248270

Generation 6 - Current best internal CV score: -0.207305754703248270

Generation 7 - Current best internal CV score: -0.207305754703248270

Generation 8 - Current best internal CV score: -0.207305754703248270

Generation 9 - Current best internal CV score: -0.207305754703248270

Generation 10 - Current best internal CV score: -0.20722149379028780

X_train,XBest_test,Ypipeline:_train,YXGBRegressor(RobustScaler(input_test=train_test_spl

it(x_sc,_matrix),y,trainlearning_size=0.7,_rate=0.01,random_state=2)max_de

Do Not Copy, Lead City University, Nigeria

input_shape = [X_train.shape[1]]

model_svm = SVC(random_state=1)

model_svm.fit(X_train, Y_train)

y_pred_svm = model_svm.predict(X_test)

print(classification_report(Y_test, y_pred_svm))

	precision	recall	f1-score	support0
0	0.72	0.86	0.79	7310
1	0.88	0.76	0.82	10100
accuracy			0.80	17410
macro avg	0.80	0.81	0.80	17410
weighted avg	0.82	0.80	0.80	17410

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Reds):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

Do Not Copy, Lead City University, Nigeria

```
cm[np.isnan(cm)] = 0.00
thresh = cm.max() / 2.0
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0
horizontalalignment="center",0
color="white" if cm[i, j] > thresh else "black")0
plt.tight_layout()0
plt.ylabel('True label')0
plt.xlabel('Predicted label')0
cm = confusion_matrix(Y_test, y_pred_svm)0
target_names = ["True", "False"]0
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0
accuracy_score(Y_test, y_pred_svm)0
0.8024124066628374
# plot for SVM0
false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_svm)0
print('roc_auc_score for SVM: ', roc_auc_score(Y_test, y_pred_svm))0
plt.subplots(1, figsize=(10,10))0
plt.title('Receiver Operating Characteristic - SVM')0
plt.plot(false_positive_rate1, true_positive_rate1)0
plt.plot([0, 1], ls="--")0
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0
plt.ylabel('True Positive Rate')0
plt.xlabel('False Positive Rate')0
plt.show()0
roc_auc_score for SVM:  0.81080846798770170
```

Do Not Copy, Lead City University, Nigeria

#KNN model0

model_KNN = KNeighborsClassifier()0

model_KNN.fit(X_train, Y_train)0

y_pred_KNN = model_KNN.predict(X_test)0

print(classification_report(Y_test, y_pred_KNN))0

	precision	recall	f1-score	support0
0	0.73	0.87	0.80	7310
1	0.89	0.77	0.83	10100
accuracy			0.81	17410
macro avg	0.81	0.82	0.81	17410
weighted avg	0.83	0.81	0.82	17410

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

Do Not Copy, Lead City University, Nigeria

```
cm[np.isnan(cm)] = 0.00
thresh = cm.max() / 2.0
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0
horizontalalignment="center",0
color="white" if cm[i, j] > thresh else "black")0
plt.tight_layout()0
plt.ylabel('True label')0
plt.xlabel('Predicted label')0
cm = confusion_matrix(Y_test, y_pred_KNN)0
target_names = ["True", "False"]0
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0
accuracy_score(Y_test, y_pred_KNN)0
0.8139000574382539
# plot for KNN0
false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_KNN)0
print('roc_auc_score for KNN: ', roc_auc_score(Y_test, y_pred_KNN))0
plt.subplots(1, figsize=(10,10))0
plt.title('Receiver Operating Characteristic - SVM')0
plt.plot(false_positive_rate1, true_positive_rate1)0
plt.plot([0, 1], ls="--")0
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0
plt.ylabel('True Positive Rate')0
plt.xlabel('False Positive Rate')0
plt.show()0
roc_auc_score for KNN:        0.82203207324836460
```

Do Not Copy, Lead City University, Nigeria

#RandomForestClassifier(ensemble) model0

model_RFC = RandomForestClassifier(random_state = 1)0

model_RFC.fit(X_train, Y_train)0

y_pred_RFC = model_RFC.predict(X_test)0

print(classification_report(Y_test, y_pred_RFC))0

	precision	recall	f1-score	support0
0	0.76	0.90	0.82	7310
1	0.91	0.79	0.85	10100
accuracy			0.84	17410
macro avg	0.84	0.85	0.84	17410
weighted avg	0.85	0.84	0.84	17410

#==========CONSTRUCTING THE CONFUSION MATRIX==========#

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):

plt.figure(figsize=(10,10))

plt.imshow(cm, interpolation='nearest', cmap=cmap)

plt.title(title)

plt.colorbar()

tick_marks = np.arange(len(classes))

plt.xticks(tick_marks, classes, rotation=45)

plt.yticks(tick_marks, classes)

if normalize:

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

cm = np.around(cm, decimals=2)

Do Not Copy, Lead City University, Nigeria

```
cm[np.isnan(cm)] = 0.0
thresh = cm.max() / 2.
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
plt.text(j, i, cm[i, j],
plt.ylabel('True label')
plt.xlabel('Predicted label')
cm = confusion_matrix(Y_test, y_pred_RFC)
target_names = ["True", "False"]
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')
accuracy_score(Y_test, y_pred_RFC)
0.8368753589890867
type(imagesss)0
rows = 20
cols = 30
fig = plt.figure(figsize=(20,10))0
for j in range(0, rows*cols):0
fig.add_subplot(rows, cols, j+1)0
plt.imshow(imagesss[j])0
false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_RFC)0
print('roc_auc_score for RFC: ', roc_auc_score(Y_test, y_pred_RFC))0
plt.subplots(1, figsize=(10,10))0
plt.title('Receiver Operating Characteristic - RFC')0
plt.plot(false_positive_rate1, true_positive_rate1)0
plt.plot([0, 1], ls="--")0
plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0
```

Do Not Copy, Lead City University, Nigeria

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for RFC: 0.845046118838970

Colab paid products0 -0 Cancel contracts here

! pip install tqdm update_checker tqdm

! pip install tpot

import matplotlib

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import seaborn as sns

import sklearn

import imblearn

from sklearn.preprocessing import StandardScaler, MinMaxScaler

from sklearn.preprocessing import LabelEncoder, MultiLabelBinarizer, LabelBinarizer from sklearn.ensemble import RandomForestClassifier from sklearn.feature_selection import RFE

import itertools

from sklearn.model_selection import train_test_split

from tpot import TPOTRegressor

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_auc_score, roc_curve

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

Do Not Copy, Lead City University, Nigeria

```
from sklearn.manifold import LocallyLinearEmbedding
from sklearn.decomposition import FastICA
from sklearn.svm import SVC
from sklearn.cross_decomposition import PLSRegression
from sklearn.manifold import MDS
import os
from PIL import Image
from PIL import UnidentifiedImageError
import keras
import matplotlib.image as mpimg
from matplotlib.image import imread
import cv2
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (4.64.1)0
Collecting update_checker0
Downloading update_checker-0.18.0-py3-none-any.whl (7.0 kB)0
Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update_checker) (2.23.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update_che
```

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

Installing collected packages: update-checker0

Successfully installed update-checker-0.18.00

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/0 Collecting tpot0

Downloading TPOT-0.11.7-py3-none-any.whl (87 kB)0

Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.0.2)0

Requirement already satisfied: update-checker>=0.16 in /usr/local/lib/python3.7/dist-packages (from tpot) (0.18.0)0 Collecting xgboost>=1.1.00

Downloading xgboost-1.6.2-py3-none-manylinux2014_x86_64.whl (255.9 MB)0

[redacted] 255.9 MB 44 kB/s 0

Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.7.3)0

Requirement already satisfied: numpy>=1.16.3 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.21.6)0

Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.3.5)0 Collecting deap>=1.20

Downloading deap-1.3.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.w [redacted] 139 kB 44.6 MB/s 0

Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.7/dist-packages (from tpot) (1.2.0)0 Collecting stopit>=1.1.10

Downloading stopit-1.1.2.tar.gz (18 kB)0

Do Not Copy, Lead City University, Nigeria

Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.7/dist-packages (from tpot) (4.64.1)0

Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->t

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.24.2->tpot) (2022

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.22.

Requirement already satisfied: requests>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from update-checker>=0.16->tp

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->updat

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->update-che

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.3.0->upda

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from

Building wheels for collected packages: stopit0

Building wheel for stopit (setup.py) ... done0

Created wheel for stopit: filename=stopit-1.1.2-py3-none-any.whl size=11956 sha256=cf9db89e6c31178b2e91fb919846462960

Do Not Copy, Lead City University, Nigeria

Stored in directory:

/root/.cache/pip/wheels/e2/d2/79/eaf81edb391e27c87f51b8ef901ecc85a5363dc96b8b8d71e3

0 Successfully built stopit0

Installing collected packages: xgboost, stopit, deap, tpot0

Attempting uninstall: xgboost0

Found existing installation: xgboost 0.900

Uninstalling xgboost-0.90:0

Successfully uninstalled xgboost-0.900

Successfully installed deap-1.3.3 stopit-1.1.2 tpot-0.11.7 xgboost-1.6.20

train_csv = pd.read_csv("/content/train.csv")0

test_csv = pd.read_csv("/content/test_WyRytb0.csv")0

print(train_csv.shape,test_csv.shape)0

x = train_csv.iloc[:, :-1]0

y = train_csv.iloc[:, -1]0

(17034, 2) (7301, 1)0

Create list to store the data and set the path of the image to load0 data_with_labels =

[]0

labels = []0 data_test = []0

imagePath = '/content/drive/MyDrive/blurredd/blurry/*.jpg'0

Create the training dataset

for i in train_csv.index:

nameOfFile = train_csv['image_name'][i]

if os.path.exists(imagePath+nameOfFile):

image = mpimg.imread(imagePath+nameOfFile)

if (len(image.shape)!=3): # Verify if the image is correct

Do Not Copy, Lead City University, Nigeria

```
print("L'image N°",i,' : ',nameOfFile,"")
else :
image = cv2.resize(image,(150,150))
data_with_labels.append(image)
labels.append(train_csv['label'][i])
print(len(data_with_labels))
print(len(labels))
00
00
import glob
glob.glob(imagePath)
imagess = [cv2.imread(images) for images in glob.glob(imagePath)]
type(imagess)
rows = 2
cols = 3
fig = plt.figure(figsize=(20,10))
for j in range(0, rows*cols):
fig.add_subplot(rows, cols, j+1)
plt.imshow(imagess[j])
scaler = StandardScaler()
x_sc = scaler.fit_transform(x)
# = LocallyLinearEmbedding()
#train_x = scale.fit_transform(train_x)
ica = FastICA(n_components=7,
max_iter=50,
```

Do Not Copy, Lead City University, Nigeria

random_state=100)

x_sc_ica = ica.fit_transform(x_sc)

pls = PLSRegression(n_components=5)

pls.fit(x_sc_ica, y)

PLSRegression(n_components=5)

X_train,X_test,Y_train,Y_test = train_test_split(x_sc_ica, y, train_size=0.7, random_state=2)0 input_shape = [X_train.shape[1]]0

model_svm = SVC(random_state=1)0

model_svm.fit(X_train, Y_train)0

y_pred_svm = model_svm.predict(X_test)0

print(classification_report(Y_test, y_pred_svm))0

	precision	recall	f1-score	support0
0	0.74	0.87	0.80	21720
1	0.90	0.78	0.84	30500
accuracy			0.82	52220
macro avg	0.82	0.83	0.82	52220
weighted avg	0.83	0.82	0.82	52220

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Reds):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

plt.colorbar()0

Do Not Copy, Lead City University, Nigeria

```
tick_marks = np.arange(len(classes))0

plt.xticks(tick_marks, classes, rotation=45)0

plt.yticks(tick_marks, classes)0

if normalize:0

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0

cm = np.around(cm, decimals=2)0

cm[np.isnan(cm)] = 0.00

thresh = cm.max() / 2.0

for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0

horizontalalignment="center",0

color="white" if cm[i, j] > thresh else "black")0

plt.tight_layout()0

plt.ylabel('True label')0

plt.xlabel('Predicted label')0

cm = confusion_matrix(Y_test, y_pred_svm)0

target_names = ["True", "False"]0

plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0

accuracy_score(Y_test, y_pred_svm)0

0.8203753351206434

# plot for SVM

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_svm)

print('roc_auc_score for SVM: ', roc_auc_score(Y_test, y_pred_svm))

plt.subplots(1, figsize=(10,10))

plt title('Receiver Operating Characteristic - SVM')

plt.title( Receiver Operating Characteristic SVM )
```

Do Not Copy, Lead City University, Nigeria

plt.plot(false_positive_rate1, true_positive_rate1)

plt.plot([0, 1], ls="--")

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

roc_auc_score for SVM: 0.8282045708420130

#KNN model0

model_KNN = KNeighborsClassifier()0

model_KNN.fit(X_train, Y_train)0

y_pred_KNN = model_KNN.predict(X_test)0

print(classification_report(Y_test, y_pred_KNN))0

	precision	recall	f1-score	support0
0	0.73	0.84	0.78	21720
1	0.87	0.78	0.82	30500
accuracy			0.80	52220
macro avg	0.80	0.81	0.80	52220
weighted avg	0.81	0.80	0.81	52220

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

Do Not Copy, Lead City University, Nigeria

```
plt.colorbar()0
tick_marks = np.arange(len(classes))0
plt.xticks(tick_marks, classes, rotation=45)0
plt.yticks(tick_marks, classes)0
if normalize:0
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0
cm = np.around(cm, decimals=2)0
cm[np.isnan(cm)] = 0.00
thresh = cm.max() / 2.0
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0
horizontalalignment="center",0
color="white" if cm[i, j] > thresh else "black")0
plt.tight_layout()0
plt.ylabel('True label')0
plt.xlabel('Predicted label')0
cm = confusion_matrix(Y_test, y_pred_KNN)0
target_names = ["True", "False"]0
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0
accuracy_score(Y_test, y_pred_KNN)0
0.8042895442359249
# plot for KNN
false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_KNN)
print('roc_auc_score for KNN: ', roc_auc_score(Y_test, y_pred_KNN))
plt.subplots(1, figsize=(10,10))
plt.title('Receiver Operating Characteristic - SVM')
```

Do Not Copy, Lead City University, Nigeria

plt.plot(false_positive_rate1, true_positive_rate1)

plt.plot([0, 1], ls="--")

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()

roc_auc_score for KNN: 0.80919889502762430

#RandomForestClassifier(ensemble) model0

model_RFC = RandomForestClassifier(random_state = 1)0

model_RFC.fit(X_train, Y_train)0

y_pred_RFC = model_RFC.predict(X_test)0

print(classification_report(Y_test, y_pred_RFC))0

	precision	recall	f1-score	support0
0	0.77	0.88	0.82	21720
1	0.90	0.82	0.86	30500
accuracy			0.84	52220
macro avg	0.84	0.85	0.84	52220
weighted avg	0.85	0.84	0.84	52220

#==========CONSTRUCTING THE CONFUSION MATRIX==========#0

def plot_confusion_matrix(cm, classes, normalize=True, title='Confusion matrix', cmap=plt.cm.Greens):0 0

plt.figure(figsize=(10,10))0

plt.imshow(cm, interpolation='nearest', cmap=cmap)0

plt.title(title)0

Do Not Copy, Lead City University, Nigeria

```
plt.colorbar()0
tick_marks = np.arange(len(classes))0
plt.xticks(tick_marks, classes, rotation=45)0
plt.yticks(tick_marks, classes)0
if normalize:0
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]0
cm = np.around(cm, decimals=2)0
cm[np.isnan(cm)] = 0.00
thresh = cm.max() / 2.0
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):0 plt.text(j, i, cm[i, j],0
horizontalalignment="center",0
color="white" if cm[i, j] > thresh else "black")0
plt.tight_layout()0
plt.ylabel('True label')0
plt.xlabel('Predicted label')0
cm = confusion_matrix(Y_test, y_pred_RFC)0
target_names = ["True", "False"]0
plot_confusion_matrix(cm, target_names, normalize=False, title='Confusion Matrix')0
accuracy_score(Y_test, y_pred_RFC)0
0.8422060513213329
type(imagesss)0
rows = 20
cols = 30
fig = plt.figure(figsize=(20,10))0
for j in range(0, rows*cols):0
```

Do Not Copy, Lead City University, Nigeria

fig.add_subplot(rows, cols, j+1)0

plt.imshow(imagesss[j])0

false_positive_rate1, true_positive_rate1, threshold1 = roc_curve(Y_test, y_pred_RFC)0

print('roc_auc_score for RFC: ', roc_auc_score(Y_test, y_pred_RFC))0

plt.subplots(1, figsize=(10,10))0

plt.title('Receiver Operating Characteristic - RFC')0

plt.plot(false_positive_rate1, true_positive_rate1)0

plt.plot([0, 1], ls="--")0

plt.plot([0, 0], [1, 0] , c=".7"), plt.plot([1, 1] , c=".7")0

plt.ylabel('True Positive Rate')0

plt.xlabel('False Positive Rate')0

plt.show()0

roc_auc_score for RFC: 0.8470918998829510

Colab paid products0 -0 Cancel contracts here

Do Not Copy, Lead City University, Nigeria

Appendix B

Figure 1.1

Figure 1. 1: Proposed System Flow Diagram

Do Not Copy, Lead City University, Nigeria

Biodata

CHINONYELUM VIVIAN NWUFOH

Residential Address: Plot 7, Obasa/Oloko Residential Area Oluyole Estate Extension, Ibadan, Oyo State, Nigeria

E-mail: chinonyelum.tabansi@yahoo.com

Phone Number: +234-809-9441-887

PORTFOLIO:

Orcid - **https://orcid.org/my-orcid?orcid=0000-0002-1663-5137**

Research gate- **https://www.researchgate.net/profile/Chinonyelum-Nwufoh-2**

LinkedIn – **https://www.linkedin.com/in/chinonyelum-nwufoh-553507160/**

PERSONAL DATA:

Previous Name:	Chinonyelum Vivian TABANSI
Date of Birth:	15th December, 1987.
Place of Birth:	Imo State, Nigeria.
Nationality:	Nigerian
State of Origin:	Anambra State, Nigeria
Marital Status:	Married
Name of Spouse:	Dr. Onyeka Chidiebele Nwufoh.
Address of Spouse:	Same as mine.

SKILLS:

Html, CSS, JavaScript, Data Scientist, Python (intermediate), MySQL, Teamwork, Time Management, Problem Solving, Innovative, Leadership, Empathy, Work Ethics, Verbal and Written Communication

WORK EXPERIENCE:

Federal College of Animal Health and Production Technology, Ibadan, Nigeria

Assistant Lecturer	***2013 – 2017***
Lecturer III	***2017 – 2020***
Lecturer II	***2020 – 2022***
Lecturer I	***2023 – till date***

Do Not Copy, Lead City University, Nigeria

- Led the team that restructured and developed a state-of-the-art Software Laboratory used till date for hands on lecture sessions
- Designing programme and course for the computer science department.
- Ensuring that the programme designed and its delivery complies with the quality standards and regulations of the Institution of NBTE.
- Contributing to the development of academic processes across the institution.
- Contributing to the development of learning and teaching strategies.
- I teach a range of courses such as Data Structure an Algorithm, Python language, Mobile App Development, Web Technology, Multimedia, Artificial Intelligence amongst others.
- I develop and apply innovative and appropriate teaching techniques and material which create interest, understanding and enthusiasm amongst students.
- I oversee the monitoring of student progress and provide advice and guidance to personal tutors and students as appropriate.
- Vetting examination questions making sure they meet standards.
- In charge of all the students' projects and researches in the computer science department, ensuring these researches meets quality and standards.
- I have taught over 5000 students and personally mentored over 150 students since I took up this job.
- Undertaking personal tutoring, overseeing the monitoring of my students progress and provide advice and guidance to students as appropriate.
- Always carrying out personal research and publications in line with personal objectives agreed in Staff review process
- I engage with the broader scholarly and professional communities national and international.
- I supervise and assist with supervision of students' research.
- I contribute to the development, planning and implementation of a high quality curriculum.

Emerging Markets Telecommunications Services (EMTS) (Now known as 9mobile Networks) Nigeria

Customer Care Executive ***2011 – 2013***

As a CCE I was saddled to perform the following duties and much more;

Do Not Copy, Lead City University, Nigeria

- Catering to customer phone calls and diverting the call to the relevant department for a more advanced form of query resolution.
- Curating streamlined email and social media communication mediums for offers, updates and much more.
- Dealing with customer issues and churning out an easy-to-follow solution
- Managing payment and delivery of customer orders.
- Helping customers choose the right product for their requirements and budget.
- Handling customer concerns and complaints in a timely manner.
- Informing customers of upcoming promotions or deals.
- Establishing a positive rapport with all clients and customers in person or via phone.
- Forming reports based on customer satisfaction statistics and helping their team to develop new skills.
- Fixing appointments based on the availability of customers and clients.
- Interacting with customers to ensure they have a desirable and shareable experience.

ECOWAS (Economic Community of West African States)

IT Support Assistant ***2010 – 2011***

As an assistant IT support in ECOWAS I was assisting the permanent staff to carry out the following responsibilities; Most of the times I carried out these duties in languages I was not used to such as French.

- Installing and configuring computer hardware, software, systems, networks, printers, and scanners.
- Monitoring and maintaining computer systems and networks
- Responding in a timely manner to service issues and requests.
- Providing technical support across the community (this may be in person or over the phone)
- Setting up accounts for new users in the community.
- Repairing and replacing equipment as necessary.
- Testing new technology.
- Training staff on how to use their normal and new systems and technologies.

EDUCATION & QUALIFICATIONS

July 2023 **Lead City University, Nigeria**

Do Not Copy, Lead City University, Nigeria

	Ph.D (Computer Science)
December 2020	**Lead City University, Nigeria**
	M.Sc Computer Science
April 2017	**University of Liverpool, UK.**
	PGDE. (Information System Management)
2005-2009	**University of Nigeria, Nsukka**
	B.Sc. Computer Science
1998-2004	**Federal Government Girls' College, Owerri**
	West African Senior School Certificate.
1993-1998	**Alvan Ikoku College of Education (AICE) Staff Primary School, Owerri**
	First School-Leaving Certificate

PROFESSIONAL / CERTIFICATION

- Oracle Certified Professional (OCP)
- Oracle Certified Associate (OCA) - With Oracle University
- System Database Administrator (SQL Fundamental 1)

WORKSHOP/ TRAINNIG ATTENDED WITH DATES.

- Digital Business Management (Google) 2015
- Information Search training for Information Technology Professionals 2016
- Presentation Techniques training for Information Technology Professional 2016
- Scratch Programming training for Information Technology Professionals 2016
- Embedded Systems 2018
- Internet of Things (IoT) 2018
- Circuit Modelling and Scientific Computing 2019
- Basic Networking using Routers 2020
- Python Programming 2021
- R-Programming 2021
- Role of Administrative Processes in Performance Improvement and Productivity Enhancement 2022

MEMBERSHIP OF PROFESSIONAL BODIES

Do Not Copy, Lead City University, Nigeria

Member: Nigeria Computer Society (NCS) - **Registration Number 11825 2020**

Member: Computer Professional Registration Council of Nigeria **8900/2023**

PUBLICATIONS (Published and Unpublished)

A. PROJECT/ THESIS/DISSERTATION

1. **Tabansi, C. V. (2009):** "Web-based Cargo Management Services" Bachelor of Science in Computer Science Project. Department of Computer Science, University of Nigeria, Nsukka. Enugu State. Nigeria.
2. **Nwufoh, C.V. (2017): "**Effects of Management Information System on Managerial Performance - Decision Making and Service Rendering: Case Study of The University of Ibadan, Nigeria". A Dissertation to the University of Liverpool, for the award of Post Graduate Diploma (PGDE) in Information Systems Management (ISM).
3. **Nwufoh, C. (2020):** "University Course Timetabling System using Constraint Satisfaction Problem (CSP) Algorithm". A Project for the award of Master of Science in Computer Science. Department of Computer Science, Lead City University, Ibadan, Oyo State. Nigeria.

B. JOURNAL ARTICLES

1. **Nwufoh C. V**, Olanrewaju O.T, Johnson Alabi. Design and Implementation of an Institution Based Emergency Alert Management System. 2022.
2. Oluwatolani Achimugu, Philip Achimugu and **Chinonyelum Nwufoh**. "An Improved Approach for Generating Test Cases during Model-Based Testing Using Tree Traversal Algorithm". Journal of Software Engineering and Applications. 2021. DOI: 10.4236/jsea.2021.146015 ISSN: 1945- 3116, ISSN: 1945-3124
3. Olanrewaju O. T, Akinosho G. A., Togun O. A., Ayobioloja S. P., Adewale F. O., **Nwufoh C. V**, Esuola F. B, Oluwasegun Z. P. "An Android Application Project Repository System for Department of Computer Science, Federal College of Animal Health and Production Technology". Annals of Research Journal Vol. 1 page 238-243, 2021
4. RFID Reader Collision Avoidance Using CSMA/CA with Fibonacci Backoff Algorithm. November 2021.

Do Not Copy, Lead City University, Nigeria

5. Dada T.O, Togun O.A, Adegbile A. A, Olanrewaju O. T, Idowu I. R, Adewale F. O, **Nwufoh C. V**, Oluwasegun Z. P. "Design and Construction of Stand-Alone Weather Monitoring System". Annals of Research Journal Vol. 1 page 232-237, 2021
6. Adegbile, A.A., Ayobiolaja, S.P., Olanrewaju, O.T., Togun, O.A., **Nwufoh, C.V.** Design and Implementation Of A Virtual Project Repository System International Journal of Computer Science and Information Security (IJCSIS), Vol. 16, No. 7, July 2018

C. CONFERENCE PROCEDINGS

1. **Nwufoh Chinonyelum Vivian** and Sakpere Wilson. "An independent component Analysis model for classifying blurred text detection in wild Scene". 3rd FASCON Conference, 2nd – 4th November, 2022 at the International Conference Centre, Lead City University, Ibadan, Oyo State, Nigeria.

2. **Nwufoh C. V,** Achimugu P. O, Achimugu O. and Chollom T. D. **"**A Hard Constraint Satisfaction Problem (HCSP) Algorithm for University Course Time Tabling". 1st International Conference on Data Science and Engineering, Nigeria. Held at Air Force Institute of Technology Kaduna, Nigeria (6th-8th December, 2021) 203-211
3. **Nwufoh C.V,** Ridwan Kolapo, Olajide Ogunsanwo, Temilola John-Dewole. "A Secured, Efficient and Simplified Symmetric Application for Encryption and Decryption of Text Files using a One Time Key". 2nd International Conference on Applied ICT, Lead City University, Ibadan. October 2019.

REFEREE

Dr. Jolade Sansi
Department of Science Laboratory Technology,
Federal College of Animal Health & Production Technology Moor Plantation, Ibadan. Oyo State. Nigeria
rsansi17@gmail.com
+2348087999258

Do Not Copy, Lead City University, Nigeria

Signature	Date

Do Not Copy, Lead City University, Nigeria

The University Compliance Certification

This is to certify that this thesis by **Nwufoh Chinonyelum Vivian** with Matriculation Number **LCU/PG/000136** in the Department of Computer Science, Faculty of Natural and Applied Sciences, Lead City University, Ibadan is in full compliance with the approval of the University's format and style.

…… ……

Signature Date

Do Not Copy, Lead City University, Nigeria

Chinonyelum Vivian NWUFOH LCU LIBRARY

ORIGINALITY REPORT

5%	4%	5%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian School of Business Student Paper	3%
2	Submitted to Middlesex University Student Paper	2%

Exclude quotes	Off	Exclude matches	< 2%
Exclude bibliography	On		

Do Not Copy, Lead C