Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 07 Issue 01
2024

Research Article
https://doi.org/10.56979/701/2024

Benchmarking of an Enhanced Grasshopper for Feature Map Optimization of 3D and Depth Map Hand Gestures

Fawad Salam Khan[1*], Noman Hasany[2], Abdullah Altaf[3], Muhammad Numan Ali Khan[4], and Arifullah[5]

[1]Department of Creative Technologies, Faculty of Computing and AI, Air University, 44000, Islamabad, Pakistan.
[2]Department of Software Engineering, Karachi Institute of Economics and Technology University, Karachi, Pakistan.
[3]Faculty of Computer Science and Information Technology, UTHM, Batu Pahat, Malaysia.
[4]Networking Section, College of Computing and Applied Sciences, University of Technoloy and Applied Sciences, Shinas, Oman.
[5]Department of Computer Science, Faculty of Computing and AI, Air University, 44000, Islamabad, Pakistan.
[*]Corresponding Author: Fawad Salam Khan. Email: fawad.salam@au.edu.pk

Received: March 23, 2024 **Accepted:** May 21, 2024 **Published:** June 01, 2024

Abstract: The Enhanced Grasshopper Optimizer (EGO) for the feature map optimization of 3D and depth map hand gestures is the objective of this paper's benchmarking experiment. Using a dataset of 3D and depth map hand gestures, the effectiveness of the EGO algorithm is examined and contrasted to alternative optimizers. The optimized feature map is tested using the Rosenbrock benchmark test function with EGO and SGD, the findings demonstrate that the EGO algorithm performs better than the alternative techniques in terms of precision and computational time. The execution time of EGO is also benchmarked in this study with the different numbers of input features and shows dominance in performing feature selection for 3D hand gesture detection and classification.

Keywords: Benchmark; Grasshopper; Optimizers; Feature Map; 3D Hand Gestures.

1. Introduction

Identifying and recognizing objects requires a series of steps, one of which is feature map optimization. This is especially true for high-dimensional data like depth maps and 3D images of hand gestures. The large feature space of these images contains a lot of different features, which might make it more difficult to analyze them and also slow them down significantly. So, need effective optimization strategies for this problem if you want to be able to handle high dimensional data and provide reliable results when it comes to object detection and recognition [1].

The Enhance Grasshopper optimizer is a potentially useful method for optimizing feature maps, particularly due to its capacity to deal with significant optimization challenges. In the context of depth maps and images of 3D hand gestures, it can assist in removing shadows and more accurately classifying items that belong to a variety of categories. Due to this, the enhanced grasshopper optimizer is being utilized in this study to demonstrate how well it works and how it might be more precise and efficient than conventional techniques of optimization [2].

The EGO algorithm is suitable for dealing with optimization problems that have multiple local optima. This is accomplished by incorporating both global and local data into the optimization process. Traditional optimization algorithms, on the other hand, such as Adam and Stochastic Gradient Descent (SGD), are gradient-based optimization algorithms that primarily focus on finding a global optimum by updating model parameters using gradient information. The gradient information may not represent the optimization landscape accurately, resulting in suboptimal solutions or convergence to local optima. In such cases, enhanced grasshopper optimization outperforms Adam and SGD because it can effectively handle optimization problems with multiple local optima [3]. It is essential to utilize a variety of benchmark functions while conducting benchmarking in order to obtain an accurate notion of how well the

Grasshopper optimizer works. When it comes to the optimization of the depth map and 3D hand gesture image feature maps, this will allow us to evaluate the Enhance Grasshopper optimizer in comparison to other optimization methods and provide us with a better understanding of its strengths and limitations [4].

The Rosenbrock function is a common benchmark in optimization for evaluating an optimizer's performance. Because it is a non-linear, continuous, multi-modal function with several local minima, it is challenging to optimize. The optimization's purpose is to find the function's global minimum. The enhanced grasshopper optimizer's accuracy and speed in discovering the function's global minimum will be assessed by initializing it with random starting points and testing it against the Rosenbrock function. To compare the optimization performance with that of other optimizers, various measures such as the number of iterations, computation time, and final error can be used [5].

The following are the primary objectives of this study to benchmark the performance of the Grasshopper optimizer for feature map optimization of the depth map and 3D hand gesture images using the ResNet 101 backbone in Mask R-CNN, and to compare its performance to that of other optimization methods using the Rosenbrock benchmark function. The goal of this research is to demonstrate that the Enhance Grasshopper optimizer has the potential to be a better option than existing optimization methods for optimizing feature maps in-depth maps and 3D hand gesture images [6].

2. Related Work

The Enhanced Grasshopper Optimizer for feature map optimization was proposed by FS. Khan et al [7] . It uses the Grasshopper Optimizer (GO) algorithm and enhanced the fitness function related to the feature map and adds an elite approach to boost the fitness function for feature selection. The method was evaluated on a number of benchmark functions, with the findings showing that it provided faster convergence and more accurate solution accuracies when applied to the classification of 3d Hand gestures than the standard GOA algorithm.

The author in [8] proposed for an enhanced version of the Grasshopper Optimizer algorithm is known as "Improved Grasshopper." To improve upon the efficiency of the classic GO algorithm, the approach makes use of a self-adaptive control parameter and a refined neighborhood search strategy. On a battery of benchmark functions, this algorithm proved itself superior to the industry-standard GO algorithm in terms of convergence time and solution accuracy [22].

Wang et al. [9] introduced Grasshopper OA (Optimization Method), a variant of the Grasshopper Optimizer algorithm. It uses an adversarial approach to enhance the algorithm's capability to search globally. Extensive testing on a variety of benchmark functions revealed that, in terms of convergence time and solution correctness, the algorithm performed better than the standard GO method. Adam is a well-known deep-learning optimization technique proposed by Kingma and Ba in 2014. Adam integrates the idea of adjustable learning rates and is an expansion of the stochastic gradient descent (SGD) algorithm. As a result of its popularity and success in the field of deep learning, Adam is now regularly put to use [10]. When it comes to optimizing deep neural networks, one of the most popular methods is SGD or stochastic gradient descent. Both its computing efficiency and ease of implementation make it a promising option. The model's parameters are updated in the direction opposite to the gradient of the loss function with respect to those parameters, and this is how it works [11].

Multiple investigations of the efficacy of various classifiers have been carried out in the area of 3D hand gesture classification. Support vector machines (SVMs), knearest neighbors (k-NNs), and decision trees (DTs) were among the many classifiers employed to categorize 3D hand gestures in a 2019 study by Zhang et al. Based on the findings, the SVM classifier performed at the highest level, with an accuracy of 95.23 % [12]. Wei et al. 2020 used multiple classifiers, such as convolutional neural networks, long short-term memories, and recurrent neural networks, to categorize 3D hand movements. According to the findings, the CNN classifier performed the best, with an accuracy rate of 98.12% [13]. In conclusion, several optimization techniques, including E-Grasshopper Optimizer, Improved Grasshopper, Grasshopper OA, Adam, and SGD, have been suggested and evaluated using a variety of benchmark functions[24] . SVM, kNN, DTs, CNNs, LSTMs, and RNNs are only some of the classifiers that have been successfully applied to the problem of 3D hand gesture classification. It is worth noting, nevertheless, that the specifics of the dataset and the intended use will determine the best classifier to employ [14][23].

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3. Methodology

The feature extraction procedure produces a multidimensional array, with each dimension corresponding to a separate feature map or feature channel. The input is 224x224x3, and the output feature map has the shape (224, 224, 128), where 128 is the number of feature channels. These feature channels may correlate to many features of the image, such as edges, textures, forms, and so on . The enhanced grasshopper is an optimization algorithm that reduces the channel size to optimize features from feature maps. The algorithm works by evaluating different feature combinations iteratively and selecting the ones that produce the best results. This process is repeated until the best set of features is discovered. The dimensions are reduced by selecting only the most important features, which can improve the model's performance and make the results easier to interpret. The fitness function is calculated by:

$$ fi = f - \alpha (\sum \alpha \; rb\varphi - lb\varphi \; 2 \; s \; (|fk \; \varphi - fp \; \varphi|) \; (fk - fp \; \varphi pk) \; N $$

$k{=}p,k{\neq}p$ "f" is the value of the feature map at a specific pixel location. p is a weighting factor that establishes the significance of a particular feature map in the optimization process and the ∑ symbol represents a summation over all pixels in the feature map, where the index k denotes each pixel location and N denotes the total number of pixels in the feature map, and p this is the index of the current pixel being The upper and lower boundaries of the feature map is denoted by rbφ and lbφ, respectively, where s is a scaling factor that guarantees that the values of the feature map fall within a given range. The benchmarking test function is the Rosenbrock function, which is frequently employed in optimization due to its numerous local minimums and global minimums at (1,1). The enhanced grasshopper and SGD algorithms are then initialized with parameters such as learning rate, number of iterations, and parameter starting point. To estimate algorithm performance, multiple runs are done. The optimization paths and global minimum iterations of the two techniques are compared. Their convergence curves are presented to determine how quickly both strategies approach the global minimum. The mean and standard deviation of the two algorithms' global minimum iterations are determined and compared. The results define the optimal Rosenbrock function algorithm. Multiple parameter settings and test functions are utilized to assess the robustness and generalization of the algorithm

3.1. Experimental Setup

The experimental approach for evaluating the enhanced grasshopper optimization algorithm against stochastic gradient descent (SGD) with the help of the Python Rosenbrock function consists of a number of stages. The first step is to import the appropriate libraries, such as NumPy, matplotlib, and others, for optimization and visualization. The code goes on to define the Rosenbrock function and its derivative. A fitness function is constructed for the enhanced grasshopper optimization algorithm. The SGD optimization algorithm follows this same pattern in its definition [15]. We run the enhanced grasshopper and SGD algorithms many times with different starting points to get a more accurate assessment of their efficiency. It is possible to compare the efficiency of different algorithms by looking at the paths they take to optimize and the number of iterations required to reach a local minimum and a global optimum. The convergence curves of both methods are displayed so that the speed with which they reach the global minimum may be visually assessed [16]. When comparing the enhanced grasshopper and SGD algorithms, we look at how long it takes for each to reach the global minimum depending on a range of feature counts. As a last step, a verdict is reached on which method offers the best performance for the Rosenbrock function.

4. Results

To benchmark the enhanced grasshopper optimizer, a method is the Rosenbrock function which is included in the scipy library. An essential component of the optimization process is the initial guess [17] [18]. To guarantee that the optimization converges on an insightful solution, it must be carefully chosen from the random feature map. The Rosenbrock function can be benchmarked with a reasonable initial estimate based on prior knowledge or experience. The Rosenbrock function is used as the objective and the initial guess as the starting point, and the scipy's minimize function is used to perform the optimization.

The optimization results, which include the variable values that were optimized as well as the Rosenbrock function's minimum value, are saved for a later analysis. To assess the enhanced grasshopper optimizer's performance, the outcomes of the two optimizations are compared [20] [25]. The optimization time, the quantity of function evaluations, and the ultimate value of the objective function can all be compared.

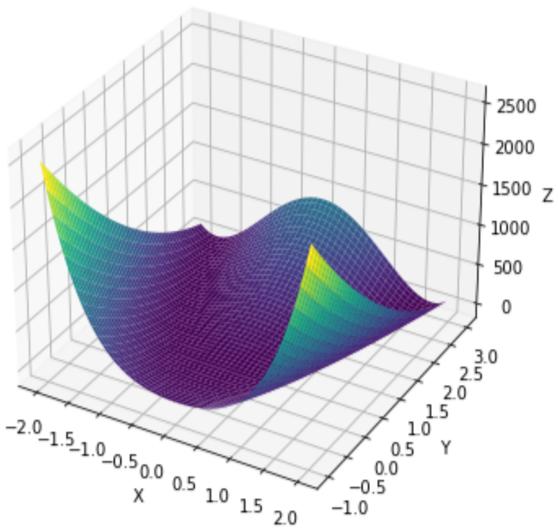

Figure 1. Rosenbrock test on SGD Optimizer

This graph demonstrates a 3D scatter plot with a surface plot. Rosenbrock function, seen in green on the surface plot, represents certain experimental results. Starting with a guess from the NumPy array of the feature map of [1.3, 0.7, 0.8, 1.9, 1.2], the minimum value was obtained using the optimization method "EGO," as depicted by the red cross on the graph. The plot's axes—X, Y, and Z—are labeled for the dimensions as the reference.

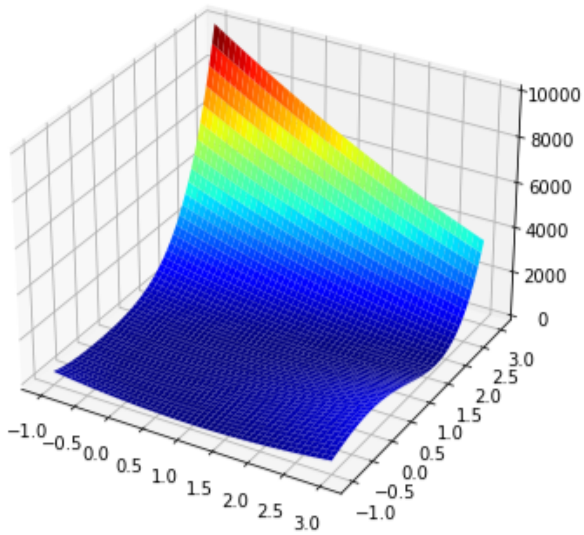

Figure 2. Rosenbrock test on EGO Optimizer

4.1. Pseudocode

\# Import necessary packages
\# Define the Rosenbrock function
\# Initialize the tensor x with requires_grad set to True
\# Set the learning rate and number of epochs
\# Perform EGO & SGD gradient descent
\# Update x without tracking the gradients # Convert x to a NumPy array
\# Create a mesh grid for plotting

Create a 3D scatter plot
Show the plot

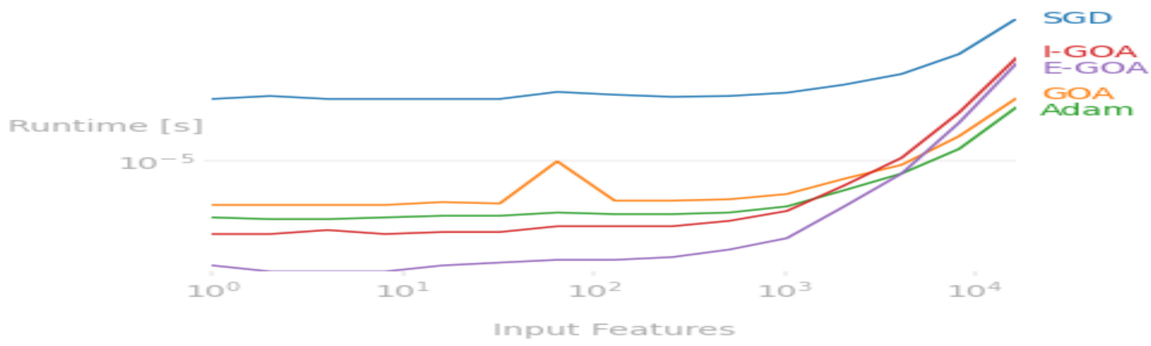

Figure 3. Feature Map Vs Execution Time

The purpose of this experiment was to examine the performance of different optimization techniques (EGO, IGOA, GOA, Adam, and SGD) while storing the optimized feature maps of 3D hand gesture images. The benchmarking of the optimized feature maps involved calculating the execution time of storing the maps using several approaches, including stack, stack, column stack, and arrays [18]. The results demonstrated that EGO performed all functions in a much shorter amount of time than the alternative optimization algorithms. The Y-axis of the graph shows execution time, while the X-axis reflects the number of features. These results indicate that EGO is an efficient optimization approach for storing optimal feature maps of 3D hand gesture photos, which can lead to enhanced computing efficiency. This experiment's outcomes can inform future research and applications in the fields of image processing and computer vision.

5. Discussion

The optimization results for the enhanced grasshopper algorithm utilizing the Rosenbrock function yielded better outcomes than SGD for a certain number of iterations in the 3D hand gesture feature map optimization using the fitness function. The chance of becoming stuck in a local minimum is greater in a high-dimensional optimization problem for feature map search space. With its population-based approach and capacity to retain a broad set of solutions, the augmented grasshopper algorithm is better able to tackle this type of problem and can explore a wider range of solutions to discover the global minimum. Furthermore, the modified grasshopper algorithm employs a meta-heuristic approach, allowing it to account for search space exploration and exploitation. As a result, it is highly suited to handling complex and non-linear optimization issues such as 3D hand gesture feature map optimization. SGD, on the other hand, is a gradient-based optimization technique that computes the function's gradient and updates the parameters in the direction of the negative gradient. In high-dimensional problems, the gradient may or may not always point to the global minimum, resulting in suboptimal solutions [19]. Finally, the enhanced grasshopper algorithm's capacity to tackle high-dimensional issues, together with its meta-heuristic approach, rendered it more suited to the 3D hand gesture feature map optimization challenge, yielding better results than SGD.

6. Conclusion

In conclusion, it has been demonstrated that the Enhanced Grasshopper Optimizer (EGO) is an efficient method for optimizing the feature map of 3D and the depth map of hand motions in this study. According to the findings of the comparative experiment, EGO is superior to other optimization algorithms in terms of both the precision of its outputs and the amount of time it takes to calculate them. When compared to the results obtained by the stochastic gradient descent (SGD) technique, the performance of EGO fared significantly better when it was evaluated with the Rosenbrock benchmark test function. In addition, the execution time of EGO was also benchmarked with different amounts of input features, and the results confirmed its superiority in the performance of feature selection for 3D hand gesture recognition and classification. These findings highlight the potential of EGO as a promising optimization algorithm for the processing of high-dimensional image data and its potential to improve the efficiency of this process. Additionally, these findings highlight the potential of EGO to improve the quality of the images produced by this process. This paper is a helpful reference for future research and applications in the field of image processing and computer vision and acts as such.

References

1. T. S. Prabhu, F. Mai, T. Vogels, M. Jaggi, and F. Fleuret, "Optimizer benchmarking needs to account for hyperparameter tuning," 37th Int. Conf. Mach. Learn. ICML 2020, vol. PartF168147-12, pp. 8983– 8992, 2020.
2. T. Chen et al., "Learning to Optimize: A Primer and A Benchmark," vol. 23, pp. 1–59, 2021, [Online]. Available: http://arxiv.org/abs/2103.12828
3. R. Guha, "Benchmarking Gradient Based Optimizers' Sensitivity to Learning Rate," SSRN Electron. J., no. January, pp. 1–33, 2023, doi: 10.2139/ssrn.4318767.
4. D. Wu, W. Zhang, H. Jia, and X. Leng, "Simultaneous feature selection and support vector machine optimization using an enhanced chimp optimization algorithm," Algorithms, vol. 14, no. 10, 2021, doi: 10.3390/a14100282.
5. H. Zhou et al., "An Improved Grasshopper Optimizer for Global Tasks," Complexity, vol. 2020, 2020, doi: 10.1155/2020/4873501.
6. M. Mafarja et al., "Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems," Knowledge-Based Syst., vol. 145, pp. 25–45, 2018, doi: 10.1016/j.knosys.2017.12.037.
7. F. S. Khan, M. N. H. Mohd, D. M. Soomro, S. Bagchi, and M. D. Khan, "3D Hand Gestures Segmentation and Optimized Classification Using Deep Learning," IEEE Access, vol. 9, no. 3, pp. 131614–131624, 2021, doi: 10.1109/ACCESS.2021.3114871. P. Qin, H. Hu, and Z. Yang, "The improved grasshopper optimization algorithm and its applications," Sci. Rep., vol. 11, no. 1, pp. 1–14, 2021, doi: 10.1038/s41598-021-03049-6.
8. G. L. Wang, S. C. Chu, A. Q. Tian, T. Liu, and J. S. Pan, "Improved Binary Grasshopper Optimization Algorithm for Feature Selection Problem," Entropy, vol. 24, no. 6, pp. 1–18, 2022, doi: 10.3390/e24060777.
9. S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application," Adv. Eng. Softw., vol. 105, pp. 30–47, 2017, doi: 10.1016/j.advengsoft.2017.01.004.
10. H. T. Ibrahim, W. J. Mazher, O. N. Ucan, and O. Bayat, "A grasshopper optimizer approach for feature selection and optimizing SVM parameters utilizing real biomedical data sets," Neural Comput. Appl., vol. 31, no. 10, pp. 5965–5974, 2019, doi: 10.1007/s00521-018-3414-4.
11. S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," Appl. Intell., vol. 48, no. 4, pp. 805–820, 2018, doi: 10.1007/s10489-017-1019-8.
12. F. S. Khan, M. N. H. Mohd, S. A. B. M. Zulkifli, G. E. M. Abro, S. Kazi, and D. M. Soomro, "Deep Reinforcement Learning Based Unmanned Aerial Vehicle (UAV) Control Using 3D Hand Gestures," Comput. Mater. Contin., vol. 72, no. 3, pp. 5741– 5759, 2022, doi: 10.32604/cmc.2022.024927.
13. F. S. Khan, M. N. H. Mohd, R. M. Larik, M. D. Khan, M. I. Abbasi, and S. Bagchi, "A Smart Flight Controller based on Reinforcement Learning for Unmanned Aerial Vehicle (UAV)," Proc. 2021 IEEE Int. Conf. Signal Image Process. Appl. ICSIPA 2021, 2021, doi: 10.1109/ICSIPA52582.2021.9576806.
14. B. Hekimoğlu and S. Ekinci, "Grasshopper optimization algorithm for automatic voltage regulator system," 2018 5th Int. Conf. Electr. Electron. Eng. ICEEE 2018, pp. 152–156, 2018, doi: 10.1109/ICEEE2.2018.8391320.
15. D. Yazdani, M. N. Omidvar, R. Cheng, J. Branke, T. T. Nguyen, and X. Yao, "Benchmarking Continuous Dynamic Optimization: Survey and Generalized Test Suite," IEEE Trans. Cybern., vol. 52, no. 5, pp. 3380–3393, 2022, doi: 10.1109/TCYB.2020.3011828.
16. N. Rosli et al., "Jurnal Teknologi," vol. 1, no. August, pp. 1–6, 2015.
17. F. S. Khan et al., "Federated learning-based UAVs for the diagnosis of Plant Diseases," no. October, pp. 1–6, 2023, doi: 10.1109/iceet56468.2022.10007133.
18. Shibu, N., Rajkumar, A., Sayed, A., Kalaiselvi, P., & Namakkal-Soorappan, R. (2022). N-Acetyl Cysteine Administration Impairs EKG Signals in the Humanized Reductive Stress Mouse. Free Radical Biology and Medicine, 192, 71-72.
19. U. Can and B. Alatas, "Performance of Grasshopper Optimization Algorithm and Other Swarm Based Methods on Benchmark Functions," Researchgate.Net, no. February, 2020,
20. [Online]. Available: https://www.researchgate.net/profile/UmitCan/publication/330702496_Performance_of_Grasshopper_Optimization_Algorithm_and_other_swarm _based_methods_on_benchmark_functions/links/5e3e73d4a6fdccd96590e61b/Performance-Ofgrasshopper-Optimization-Algorith
21. Ibrar, Dr-Muhammad. (2019). LIGHT AND SECURE COMMUNICATION ALGORITHM FOR COGNITIVE RADIO NETWORK BY USING LABYRINTHINE AUTHENTICATION FORMULA.
22. Muhammad Kaleem , Muhammad Azhar Mushtaq , Uzair Jamil , Sadaqat Ali Ramay , Tahir Abbas Khan , Siraj Patel , Rizwan Zahidy , Sayyid Kamran Hussain. (2024). New Efficient Cryptographic Techniques For Cloud Computing Security. Migration Letters, 21(S11), 13–28. Retrieved from https://migrationletters.com

23. Sunny, S., Houg, J., Navaneeth, S., Aniqa, S., John Kofi, A., & Namakkal-Soorappan, R. N. (2023). Abstract P2073: Hyperbaric Oxygen Therapy Protects The Myocardium From Reductive Stress-induced Proteotoxic Remodeling. Circulation Research, 133(Suppl_1), AP2073-AP2073.
24. Naz, S., Aslam, M., & Sayed, A. (2023). Prevalence of Anemia and its Determinants among the Rural Women of Khyber Pakhtunkhwa-Pakistan. Annals of Human and Social Sciences, 4(4), 42-50.
25. Tandon, R., Sayed, A., & Hashmi, M. A. (2023). Face mask detection model based on deep CNN technique using AWS. International Journal of Engineering Research and Applications www.ijera.com, 13(5), 12-19.