Enhanced Optimized CNN Based Automated Diabetic Retinopathy Detection

Sireesha Moturi[1], Teja Sri Vankayala[2], M. Mounika Naga Bhavani[3], Eswar Kalyani Karna[4], Venkayamma Akkapalli[5], and K.V. Narasimha Reddy[6]

Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Palnadu District, Andhra Pradesh, India.
vtejasri.1016@gmail.com, medurimounika4@gmail.com,
eswarkalyani048@gmail.com, vyshnaviakkapalli@gmail.com,
narasimhareddynec03@gmail.com
sireeshamoturi@gmail.com

Abstract. Early identification is essential to prevent serious visual impairment in Diabetic patients as Diabetic Retinopathy (DR) is the main reason for blindness. In this paper, an Optimized Convolutional Neural Network (CNN) model is used to propose an automated approach for categorizing the stages of DR. The pre-trained VGG16 model uses deep feature extraction for the given retinal pictures, which uses scaling and feature selection heuristics by the Grey Wolf Optimizer. Those selected features from GWO belonging to the most relevant features would also give a better boost to the classification performance along with the optimization of both hyperparameters. The proposed model will perform better than the traditional techniques based on the Precision, Recall, and F1 scores' experimental results. It has an accuracy of 99.31% on the DR dataset. The inclusion of GWO in CNN, models hold tremendous potential for use in the analysis of medical images and yields Optimized CNN, an efficient technique that is effective in improving healthcare diagnosis.

Keywords: Diabetic Retinopathy · Optimized Convolutional Neural Network (OpCoNet) · Grey Wolf Optimizer (GWO) · VGG16 · Feature Selection · Hyperparameter Tuning

1 Introduction

Diabetes-related DR can cause blindness and vision impairment. Efficient screening and diagnostic techniques are crucial due to global diabetes incidence. [1] Deep learning and AI, particularly convolutional neural net- works, can automate diagnosis and categorize retinal disorders, improving patient outcomes. This study focuses to probe and improve deep learning methods for DR identification. Recent advancements in Diabetic Retinopathy detection have significantly improved diagnostic accuracy. Techniques like deep neural networks, convolutional neural networks, deep residual networks, DenseNet, and automated methods for detecting microaneurysms have been used. These contributions inform the use of deep learning [2] algorithms for automated drug resistance identification, enhancing the accuracy of DR diagnosis. Diabetes-related Diabetic

Retinopathy (DR) is the main techniques. [3] Automated systems, using CNN, can provide a reliable tool for early diagnosis and treatment, enhancing accessibility and effectiveness in clinical settings. This study explores a hybrid deep learning strategy to refine the efficiency and accuracy of Diabetic Retinopathy categoriza- tion, a serious microvascular consequence of diabetes, and reduce the burden of this crippling condition [4], utilizing recent artificial intelligence advancements in deep learning. Researchers have developed the DeepDR Plus system, using AI to forecast the progression of Diabetic Retinopathy, a serious side effect of diabetes. This innovative approach aims to improve screening methods and personalized patient care, ultimately reducing the burden of diabetes-related visual loss. The DR ResNet plus model automates Diabetic Retinopathy severity grading and diagnosis, improving patient outcomes and enhancing diagnosis accuracy using large datasets. This study explores novel models and strategies for efficient segmentation and classification of Diabetic Retinopathy (DR), the main basis of vision loss in diabetes patients, using machine learning and also it investigates the effectiveness of pre-trained models like VGG16, VGG19 in improving diagnostic precision for DR a major cause of vision impairment.

2 Related Work

Some deep learning models include VGG, ResNet, and DenseNet. These deep learning models were found to suffer from gaps in Diabetic Retinopathy detection. There is a need to optimize the Convolutional Neural Network using methods such as Grey Wolf Optimization for increasing the precision of detection. Shankar et al. have used deep learning algorithms and achieved 99.28% accuracy for Diabetic Retinopathy diagnosis [1]. Other models did less well, such as ResNet34 at 86% sensitivity and 85% accuracy and decision trees at 91% and VGGNet at 92%. Gadekallu et al. used CNNs to predict Diabetic Retinopathy that reached an accuracy of 96% with feature extraction [2]. Gaurav Saxena hyperparameter-optimized InceptionResNetV2 that achieved an AUC of 0.92 on the Messidor-2 dataset [3].

Hybrid model that combines ResNet and GoogleNet with adaptive particle swarm optimization achieved 94 percent accuracy by Ayesha Jabbar et al., which approaches demographic bias and reports on accuracy, precision, recall, and F1 score [4]. Farooq's CNN achieves 73%, Shah's hybrid CNN- RNN gets 90%, and Ali's transfer learning model obtains 85%[5]. Javed's optimization produced DR diagnosis automated up to an accuracy of 88%, while Butt's modified VGGNet showed 83.1% accuracy for the EyePACS dataset [6]. Iskandar's DenseNet, consisting of an attention module, obtained 97%, whereas Latif's meta-plasticity approach achieved an accuracy of 94%. Bin Sheng's DeepDR Plus system, in predicting DR, used a range of C-index between 0.823 and 0.862, so it demonstrates its applicability in the early detection of DR [7].

Gaurav Dhiman's optimized DR-ResNet+ achieved an accuracy of 98.98% with 98.29% sensitivity and 99.16% specificity using the grid and random search techniques [8]. The IC2T model developed by Jagadesh et al. was able to attain

98% accuracy in Diabetic Retinopathy detection depending on the features of the biomarkers that include blood vessels and optic discs along with a specific effect of the model architecture and optimization of the hyperparameters [9].

3 Proposed Method

The Methodology provides a detailed structure for the classification of Diabetic Retinopathy. This methodology contains a sequence of steps that help to build the classifier easily. The Fig.1, describes the steps for the DR Detection

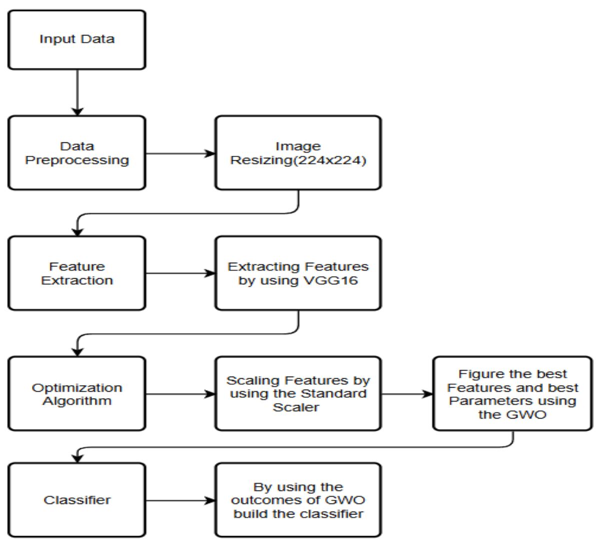

Fig. 1. The Structure of DR Classification

3.1 Dataset Description

The "Diabetic Retinopathy Level Detection" dataset on Kaggle includes retinal images categorized into different levels of severity of Diabetic Retinopathy. The present research used 4396 images to classify the level of DR. The Complete process divides the photographs into the Training and Validation sets. The Fig. 2, explains that the DR Photographs are categorized into 5 Levels.

DR of (Level 0) is absent; no symptoms of Diabetic Retinopathy are present. The retina seems to be in normal condition. (Level 1) Mild NPDR is characterized by microaneurysms, which are tiny blood vessel bulges in the retina. There are no further Retinopathy symptoms. Diabetic Retinopathy with Moderate Non-Proliferative Effects (Level 2) There are more microaneurysms along

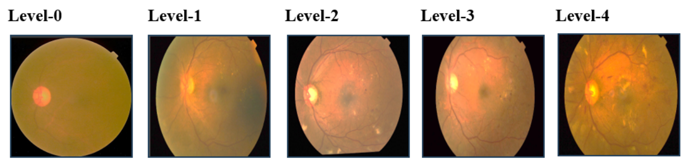

Fig. 2. Severity Of DR

with other abnormalities such as retinal hemorrhages and exudates. There can be some anomalies in the blood vessels as well. Severe DR with No Proliferation (Level 3) More substantial alterations are present at this level, including several hemorrhages in the retina, Venous beading (veins twisted and dilated), and anomalies of the intraretinal microvascular system (IRMA). The likelihood of developing proliferative Diabetic Retinopathy rises at this point. Diabetic Proliferative Retinopathy (Level 4) is distinguished by the development of new blood vessels on the optic disc or retina (neovascularization). This stage can lead to serious complications, including vitreous hemorrhages and retinal detachment, which can cause permanent vision loss if not treated. A total of 3662 Images are

Fig. 3. Class Imbalance of images in Training Image Dataset

used for training the optimized convolutional neural network [1][10]. Most images belong to no DR Stage representing the Class Imbalance. Fig. 3, represents the Class Imbalance of the photographs

3.2 Experimental Setup

The experimental setup used in this work is the Google Colaboratory. This Google Colab provides cloud-based services with TPU and GPU computational resources. The TPU v2-8 has larger RAM at 334.66 GB and 100 GB of disk storage, hence, it executes faster. Hardware Description: AMD Ryzen 5000 with

16 GB RAM was used in conducting this study to build the model in the images. NVIDIA denotes the GPU that is currently installed on the system.

3.3 Data Preprocessing

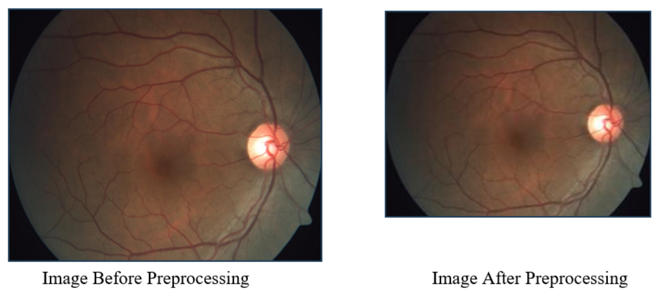

Fig. 4. The Comparison of Image Before and After Preprocessing

The Diabetic Retinopathy Level Detection dataset is made up of high-quality images. To speed up the training process, the images were resized to 224x224 and processed in batches of 32 to ensure consistency and conformity with the VGG16 model used for feature extraction. Fig. 4, outlines the differences before and after preprocessing. These methods improve computational efficiency and the model's ability to identify features when classifying Diabetic Retinopathy. Since the dataset was already divided to accommodate both the training and testing sets, data splitting was not required. Class weights are computed to address an issue referred to as class imbalance [8]. This prevents the model from being biased toward the majority class and thus removes the overuse of one's respective sampling of data to overcome this phenomenon. No data augmentation techniques, such as rotations and flips, were performed in the process, and instead, solely relied upon the given preprocessing procedures.

3.4 Feature Extraction

This process uses the pre-trained VGG16 convolutional neural network (CNN) [14], which is renowned for its depth and simplicity, to extract deep information from photos. Because it can extract high-level abstract features from images, the Oxford Visual Graphics Group developed VGG16, a 16-layer system with learnable weights that is frequently used for image classification. Our special dataset, which consists of image training and validation directories, was subjected to the VGG16 model.

The Images were prepared by scaling them between 0 and 1 and normalizing their pixel values by dividing them by 255.0. This facilitates training at a faster convergence rate. To guarantee correct scaling and normalization, the VGG16-specific preprocessing function preprocess input was also used. As requested by

VGG16, we resized the batches of photos to 224x224 pixels using the Image Data Generator, along with the corresponding labels.

The convolutional basis of the VGG16 model was utilized to extract deep features from the images using pre-trained weights from the ImageNet dataset. Then, features were combined into one array by using numpy.save() with their labels. This speeds up classifier training because it is not necessary to repeatedly extract features. Because the same stored features could be used for other applications, such as model optimization and training machine learning classifiers, thousands of computation time will be saved in total.

3.5 Grey Wolf Optimization

The Grey Wolf Optimizer (GWO) is an optimization algorithm inspired by nature, designed to replicate the hierarchical social structure and cooperative hunting strategies of Grey wolves in their natural habitat. Mirjalili et al.. introduced it in 2014. The primary purpose of GWO is to expedite the search space and compare the Grey wolf hunting process to determine the best solution to a problem. GWO does an excellent job of optimizing parameters [11][12]. Feature selection extracts the most significant features from high-dimensional datasets, simplifying and improving model performance. Global Optimization Finding the global optimum for a particular problem while avoiding local optimal traps is critical in complicated or nonlinear systems [13][15].

We extracted features from our dataset, resulting in a high-dimensional feature space with 25,088 features per Image. Feature Scale We applied a standard scaling method to the extracted data from our training and validation sets to ensure a consistent comparison of features. To narrow down the criteria and select only the most important features, we used GWO. A feature selection mask was then applied to the scaled input data. This mask acts as a binary filter: a value of '1' indicates that the feature is selected, while '0' means it is excluded. The decision to include or exclude each feature is guided by the GWO algorithm, which evaluates the contribution of each feature to the model's performance. Class homogeneity We calculated the class weights to check for the homogeneity of the data set so that the CNN model is not biased toward one class. The CNN model is designed with two connected layers (thick). In addition, a feature selection mask was used to select the most relevant features from the input data. The objective function assessed the model's performance in the validation set by calculating the validation accuracy. The most important optimal parameters are:

- Number of neurons in the first dense layer
- Number of neurons in the second layer
- Learning rate
- Dropout rate
- Selected features

This algorithm considers three variables Alpha (α), Beta (β), and Delta (δ) come from. The rest of the scenarios for these wolves show possible solutions, and the hunting process is compared as an optimization process.

- Alpha wolves (α) lead the hunt, representing the best solutions found so far.
- Beta (β) and Delta (δ) wolves help Alpha and explore the search area properly.
- Omega wolves (ω), representing the rest of the population, follow these leads and converge to the optimal solution.

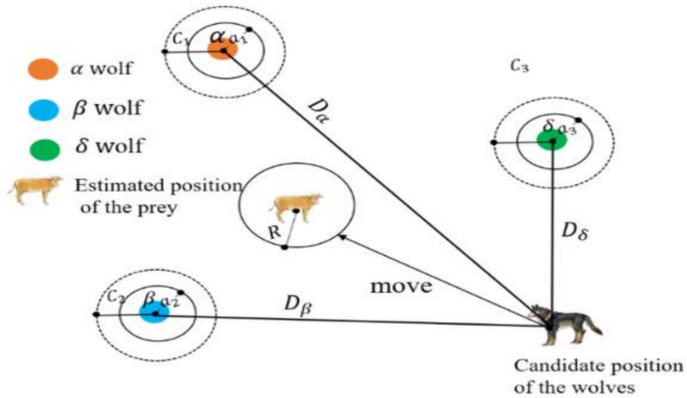

Fig. 5. Schematic Representation of the Grey Wolf Optimizer's Hunting Behavior

The new position of a wolf is calculated as the average of three positions determined by the influence of the alpha, beta, and delta wolves. Fig. 5, provides how calculations will be done in the positions

$$Z(t+1) = \frac{Z_1 + Z_2 + Z_3}{3} \tag{1}$$

Each of the positions Z_1, Z_2, and Z_3 is calculated using the following equations

$$Z_1 = \alpha(t) - P_1 \cdot A_\alpha \tag{2}$$

$$Z_2 = \beta(t) - P_2 \cdot A_\beta \tag{3}$$

$$Z_3 = \delta(t) - P_3 \cdot A_\delta \tag{4}$$

The distance between a wolf's position and the prey (best solution) is calculated as

$$A_\alpha = |Q_1 \cdot \alpha(t) - Z(t)| \tag{5}$$

$$A_\beta = |Q_2 \cdot \beta(t) - Z(t)| \tag{6}$$

$$A_\delta = |Q_3 \cdot \delta(t) - Z(t)| \tag{7}$$

The coefficients P and Q are calculated using the following formulas

$$P = 2 \cdot b \cdot s_1 - b \tag{8}$$

$$Q = 2 \cdot s_2 \tag{9}$$

Where s1 and s2 are unplanned numbers in the range [0, 1]. The parameter α decreases linearly throughout iterations to balance exploration and exploitation

$$a = 2 - \frac{2 \cdot t}{T} \tag{10}$$

t is the current iteration.
T is the total number of iterations allowed.

These equations are fundamental to the operation of the GWO algorithm, guiding the wolves toward the optimal solution in the search space. We performed GWO for ten wolves over 20 iterations of movement, where they moved based on their distance to the optimum solution. The algorithm selected almost 10,316 features out of the original 25,088, reducing the dimensionality and performance of the model. The GWO method effectively learned suitable hyperparameters along with feature selection that positively improved the CNN model at a relatively lower computing cost. This approach may be applied with even more large-scale, high-dimensional datasets for machine learning problems.

3.6 Optimized Convolutional Neural Network

OpCoNet is a designated deep learning model for the classification of stages of Diabetic Retinopathy. Herein, the model made use of CNNs with some advanced optimization techniques to enhance its performance. For hyperparameter tuning, OpCoNet adopted GWO which is inspired by the natural leadership and hunting strategy of Grey wolves. GWO helps optimize the parameters of the network while choosing the features that are most relevant for the task at hand from high-dimensional data of images, improving accuracy together with efficiency.

The model starts with an input layer that computes selected features from a dataset. Then comes the dense layer, where neurons perform weighted sums and apply activation functions to learn complex patterns. To avoid overfitting, methods such as L2 regularization should be applied to ensure that the model generalizes well beyond the training data.

Table 1. The hyperparameters for the model construction.

Hyperparameters	**Values**
Neurons Layer 1	181
Neurons Layer 2	150
Learning Rate	0.000457
Optimizer	Adam
Dropout Rate	0.1
L2 norm regularization	0.001
Loss	Categorical Cross Entropy

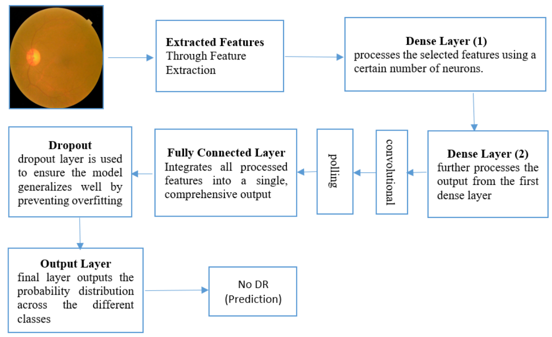

Fig. 6. The Model Architecture

Then, OpCoNet utilizes a batch normalization layer stabilizing and speeding up the training: it normalizes the output of the dense layer. Dropout: neurons are randomly disabled in training for preventing overfitting.

This is repeated in a second dense layer with a different number of neurons and batch normalization followed by dropout as shown in Fig. 6. Then, a convolutional layer applies filters to actually capture complex patterns existing within the data. Shrinkage-based pooling preserves important information within this step, thus improving efficiency. The final layer is a fully connected output layer with a SoftMax activation function producing a probability distribution to classify input.

The key hyperparameters listed in Table 1, including the number of neurons in each layer, dropout rate, learning rate, and L2 norm regularization, were determined based on the outputs of the Grey Wolf Optimizer (GWO) combined with experimental analysis. To train the model, it employs an optimizer where it calculates the weights from categorical cross-entropy loss. Early stopping has also been implemented to prevent overtraining. This stops the model at that point with no further development and applies a well-balanced and accurate predictive model.

4 Result

The best hyperparameters of the Grey Wolf Optimization build the Optimized CNN which accepts the Extracted Features as input and also the Images. The Optimized CNN with GWO Selected Features gives greater results concerning evaluation metrics. The OpCoNet with GWO Selected Features takes the Selected Extracted Features as input. Each image may have around 25088 features using the GWO, the feature set was reduced to 10,316 features. The Optimized CNN With GWO Selected Features gains 99.31% accuracy over the 3662 images Extracted Features Fig. 7, shows the classification report of the OpCoNet on the validation extracted features. Fig. 8, shows the classifier works better on the Validation Dataset

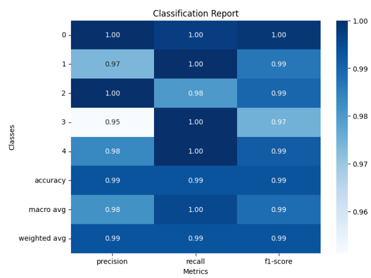

Fig. 7. Classification Report on the Validation Dataset

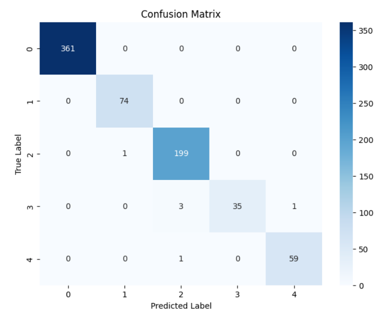

Fig. 8. Confusion Matrix on the Validation Dataset

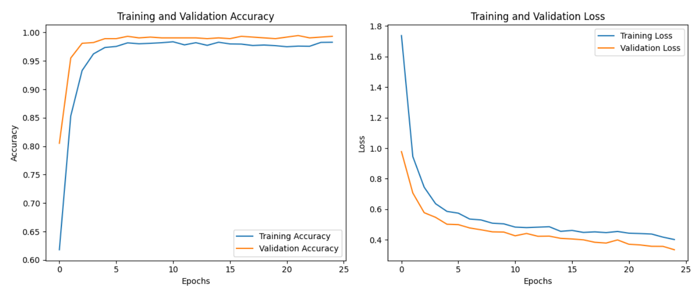

Fig. 9. The Model Performance Plots

The Training and Validation accuracy, loss from Fig. 9. The graphs illustrate strong convergence, with both training and validation accuracy nearing 99.50% while training and validation loss significantly decrease, indicating the model is well-trained without evident overfitting. Table. 2, compares different methodologies based on selected features, epochs, accuracy, and model-building time. The OpCoNet model with GWO-selected features achieves a remarkable accuracy

Table 2. Comparison of Methodologies with Different Feature Selection Approaches

Methodology	**No of Features Selected**	**Epochs**	**Accuracy**	**Time Comparison**
OpCoNet With Images (GWO) [12]	Nill	25	98.23	3 Hours
OpCoNet With All Extracted Features (GWO)	All (25088)	30	99.18	48.221 Sec
OpCoNet With GWO Selected Features	**10316**	**25**	**99.31**	**27.279 Sec**
OpCoNet With ACO Selected Features	12565	25	99.45	28.366 Sec
CNN With Images [1]	Nill	25	96.03	3 Hours
OpcoNet With All Extracted Features (ACO)	All (25088)	25	98.91	76.799 Sec

of 99.31% in 25 epochs, with a fast training time of just 27.279 seconds. This highlights GWO's effectiveness in enhancing both accuracy and computational efficiency. In comparison, the OpCoNet model with Ant Colony Optimization-Selected Features achieves a slightly higher accuracy of 99.45% but takes longer (28.366 seconds). Models using all extracted features (99.18%) or image-based methods (98.23%) deliver lower accuracy and are more computationally demanding, with the image-based approach taking up to 3 hours for training. The OpcoNet with all extracted features using the best parameters of ACO achieves a lower accuracy(98.91%) and takes a long time to build the model. Among the various methods tested, the GWO-selected features demonstrate the best trade-off between accuracy and computational efficiency, making it the most effective approach.

5 Conclusion

This study provides an overview of the identification of Diabetic Retinopathy (DR), taking into account the importance of early detection in preventing vi sion loss. DR is the main reason for blindness in persons with diabetes. This work employed an integrated Optimized Convolutional Neural Network with Grey Wolf Optimization to focus on feature selection and hyperparameter op timization. These findings unambiguously demonstrate that GWO significantly enhances CNN performance by lowering complexity, raising accuracy, and im proving other metrics like feature selection to only pertinent ones from a high dimensional dataset. Simultaneously, the GWO-CNN model outperformed the other models for the DR dataset, with an accuracy of 99.31%. This demonstrates that GWOis among the best models in contemporary computational approaches for deep learning model optimization, making it perfect for intricate medical diagnoses such as deep reinforcement learning. The technique developed in this work does not only reduces the workload for medical staff but also ensures timely and accurate diagnosis, thereby improving the outcome of the patient. Since this technique is more sensitive, highly accurate, and can be integrated easily into clinical practice, it holds great promise for enhancing medical diagnostics in the era of computational medicine

6 Dataset Availability

The Dataset link is `https://www.kaggle.com/datasets/arbethi/Diabetic-Retinopathy-level-detection`

References

1. Sebti, R., Zroug, S., Kahloul, L., & Benharzallah, S. (2021, November). "A deep learning approach for the Diabetic Retinopathy detection". In *The Proceedings of the International Conference on Smart City Applications* (pp. 459-469). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-94191-8_37

2. Ayala, J. A., Ortiz Figueroa, T., Fernandes, B., & Cruz, F. (2021). "Diabetic Retinopathy improved detection using deep learning". *Applied Sciences*, 11(24), 11970. https://doi.org/10.3390/app112411970
3. Saxena, G., Verma, D. K., Paraye, A., Rajan, A., & Rawat, A. (2020). "Improved and robust deep learning agent for preliminary detection of Diabetic Retinopathy using public datasets". *Intelligence-Based Medicine*, 3, 100022.
4. Jabbar, A., Liaqat, H. B., Akram, A., Sana, M. U., Azpíroz, I. D., Diez, I. D. L. T., & Ashraf, I. (2024). "A Lesion-Based Diabetic Retinopathy Detection Through Hybrid Deep Learning Model". *IEEE Access*.
5. Farooq, M. S., Arooj, A., Alroobaea, R., Baqasah, A. M., Jabarulla, M. Y., Singh, D., & Sardar, R. (2022). "Untangling computer-aided diagnostic system for screening Diabetic Retinopathy based on deep learning techniques". *Sensors*, 22(5), 1803.
6. Butt, M. M., Iskandar, D. A., Abdelhamid, S. E., Latif, G., & Alghazo, R. (2022). "Diabetic Retinopathy detection from fundus images of the eye using hybrid deep learning features". *Diagnostics*, 12(7), 1607. https://doi.org/10.3390/diagnostics12071607
7. Dai, L., Sheng, B., Chen, T., Wu, Q., Liu, R., Cai, C., & Jia, W. (2024). "A deep learning system for predicting time to progression of Diabetic Retinopathy". *Nature Medicine*, 30(2), 584-594. https://doi.org/10.1038/s41591-023-02702-z
8. Baba, S. M., Bala, I., Dhiman, G., Sharma, A., & Viriyasitavat, W. (2024). "Automated Diabetic Retinopathy severity grading using novel DR-ResNet+ deep learning model". *Multimedia Tools and Applications*, 1-43.
9. Jagadesh, B. N., Karthik, M. G., Siri, D., Shareef, S. K., Mantena, S. V., & Vatambeti, R. (2023). "Segmentation Using the IC2T Model and Classification of Diabetic Retinopathy Using the Rock Hyrax Swarm-Based Coordination Attention Mechanism". *IEEE Access*.
10. Kallel, F., & Echtioui, A. (2024). "Retinal fundus image classification for Diabetic Retinopathy using transfer learning technique". *Signal, Image and Video Processing*, 18(2), 1143-1153.
11. Moturi, S., Vemuru, S., Tirumala Rao, S.N., Mallipeddi, S.A. (2023). Hybrid Binary Dragonfly Algorithm with Grey Wolf Optimization for Feature Selection. In: Hassanien, A.E., Castillo, O., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. ICICC 2023. Lecture Notes in Networks and Systems, vol 703. Springer, Singapore.https://doi.org/10.1007/978-981-99-3315-0_47
12. Minija, S. J., Rejula, M. A., & Ross, B. S. (2024). "Automated detection of Diabetic Retinopathy using optimized convolutional neural network". *Multimedia Tools and Applications*, 83(7), 21065-21080.https://doi.org/10.1007/s11042-023-16204-0
13. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). "Grey wolf optimizer". *Advances in Engineering Software*, 69, 46-61.
14. M.Sireesha, S. N. Tirumala Rao, Srikanth Vemuru, Optimized Feature Extraction and Hybrid Classification Model for Heart Disease and Breast Cancer Prediction International Journal of Recent Technology and Engineering Vol - 7, No 6, Mar - 2019 ISSN - 2277-3878, Pages – 1754 – 1772
15. Sireesha Moturi, S.N.Tirumala Rao, Srikanth Vemuru, Grey wolf assisted dragonfly-based weighted rule generation for predicting heart disease and breast cancer, Computerized Medical Imaging and Graphics, Volume 91, 2021, 101936, ISSN 0895-6111.