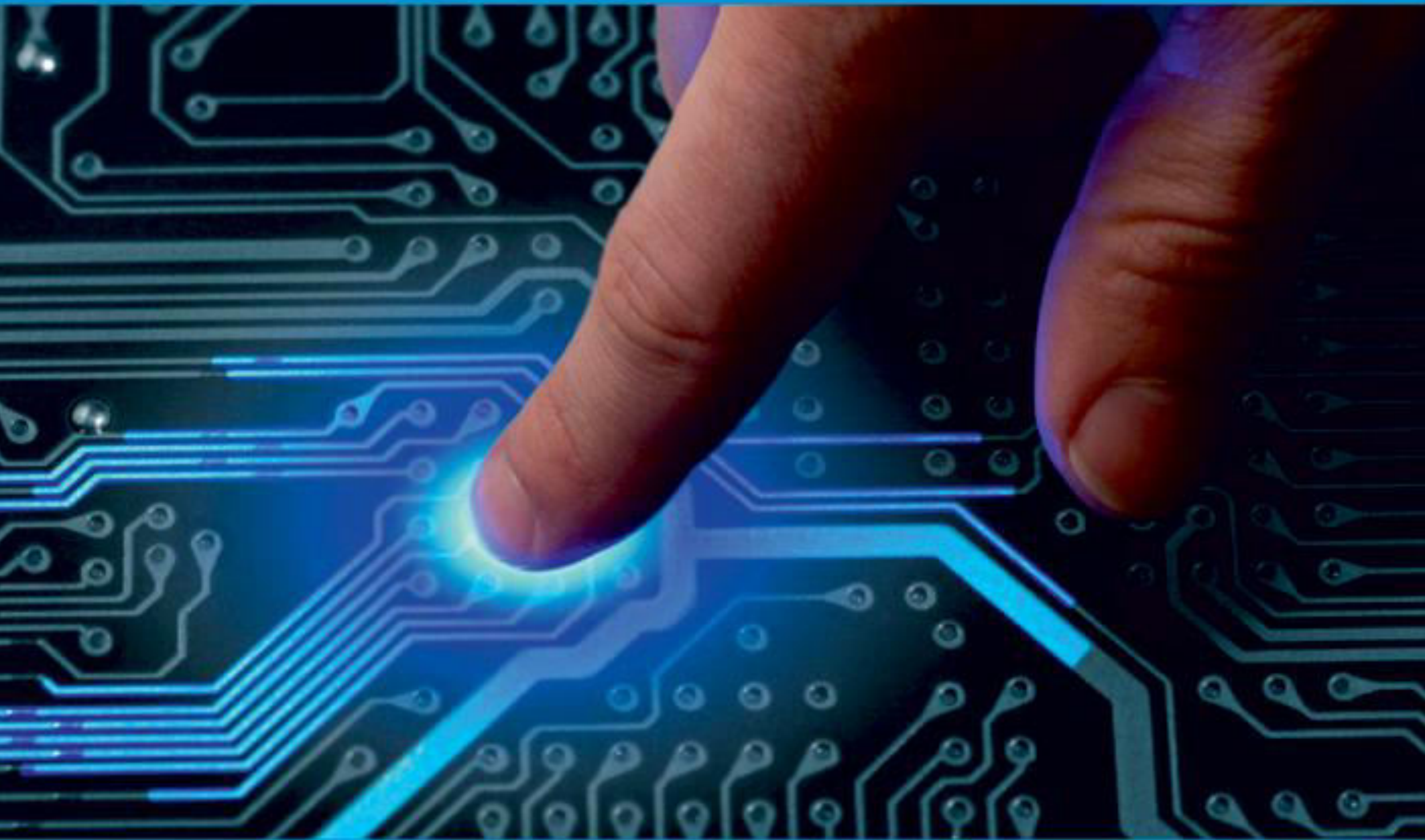
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 5, May 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205263 |

Detection of Chronic Kidney Disease from Retinal Images Using Deep Learning

Gobinathan B[1], Yogendra Kumar AKM[2], Gowtham S[3], Praveenkumar V[4]

UG Student, Department of IT, Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, Tamil Nadu, India[1 2 3]

Assistant Professor, Department of IT Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, Tamil Nadu, India[4]

ABSTRACT: Chronic Kidney Disease (CKD) is a prevalent global health issue, often diagnosed late when treatment options are limited. Early detection can significantly improve outcomes. Recently, retinal imaging has shown promise for detecting systemic diseases like CKD due to its non-invasive nature. This study proposes using transfer learning for CKD detection from retinal images. We utilize a pre-trained convolutional neural network (CNN) model for feature extraction from a large retinal image dataset. These features are then used by a classifier trained specifically for CKD detection. By fine-tuning the pre-trained CNN on a smaller dataset annotated for CKD, we adapt the model to identify CKD-related pathological features. Experimental results demonstrate the method's efficacy in accurately detecting CKD, achieving competitive performance with existing approaches. This approach reduces the need for extensive labelled data and computational resources, making it scalable and applicable in real-world clinical settings. This research advances non-invasive CKD detection methods, potentially enabling timely interventions and improving patient care outcomes.

KEYWORDS: Deep Learning(DL), Convolutional Neural Network(CNN),Residual Network(ResNet), Visual Geometry Group(VGG)

I. INTRODUCTION

Detecting Chronic Kidney Disease (CKD) from retinal images is a critical step in early diagnosis and intervention. Leveraging transfer learning, a powerful technique in machine learning, enhances the efficiency and accuracy of this process. By employing pre-trained convolutional neural networks (CNNs), which have learned rich feature representations from vast datasets, we can adapt them to extract relevant features from retinal images associated with CKD. Transfer learning streamlines the training process by reusing the knowledge gained from a source task (such as image classification) to solve a related target task (CKD detection). This approach significantly reduces the need for extensive labelled data and computational resources, making it particularly advantageous in medical imaging applications where annotated datasets may be limited. The proposed transfer learning framework is specifically tailored for CKD detection from retinal images. By fine-tuning a pre-trained CNN on a CKD retinal image dataset, we aim to optimize the network's performance in accurately identifying pathological features indicative of CKD progression. Through rigorous evaluation and validation, our approach promises to contribute to early diagnosis, timely intervention, and improved patient outcomes in the management of CKD.

II. RELATED WORK

In [1] the author used machine learning and deep learning models to predict ESRD progression in CKD patients, achieving a high AUC-ROC of 0.8991. Significant markers included hematuria, proteinuria, potassium, and urine albumin to creatinine ratio. The study emphasized personalized CKD management through machine learning.

In [2] the author utilized UCI's CKD dataset, applying KNN imputation to handle missing values. Six machine learning models were developed, with random forest achieving 99.75% accuracy. An integrated model combining logistic regression and random forest, using a perceptron, achieved 99.83% accuracy. This approach is promising for complex clinical data diagnostics.

In [3] the authors introduced the DRGC-BSODL algorithm for diabetic retinopathy (DR) grading and classification, using a three-stage process: contrast enhancement, image segmentation with Brain Storm Optimization and multilevel

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205263 |

thresholding, and feature extraction via DenseNet169. A deep neural network then classifies DR. Testing on a fundus image dataset showed the DRGC-BSODL model's superior performance.

In [4] the authors developed a CKD detection model with improved Gaussian filtering for preprocessing, watershed-based segmentation, and feature extraction. Optimized Neural Network (NN) and Long Short-Term Memory (LSTM) classifiers use Self Updated Cat Swarm Optimization (SU-CSO) to enhance prediction accuracy, outperforming other methods.

III. PROPOSED ALGORITHM

In the proposed system the pre-trained CNN model for improved CKD detection using retinal fundus image. Here's an outline of the system:

Data Collection and Preprocessing: The data is collected from the reputed online sources and for data preprocessing involves normalizing and enhancing the images, resizing them for uniformity, and augmenting the dataset to increase variability. Techniques like contrast adjustment, noise reduction, and segmentation are applied to highlight relevant features. This preparation ensures the CNN can effectively learn and identify CKD-related biomarkers, improving the accuracy and reliability of the detection model.

Model Selection: The selection of a suitable CNN model for CKD from retinal images is more important. Here the study uses VGG16(Visual Geometry Group) ResNet50(Residual Network) and Alexnet. VGG consists of several convolutional layers with fully connected layers at the end for classification. VGG16 helps to increase the accuracy and performance metrics of the model. ResNet50(Residual Network) which is mainly used for clearing the gradient problem in images for better quality and understanding of the image. This makes way to construct networks with more convolutional layers of more depth. The deeper the depth the higher the classification accuracy. The Alexnet has eight deep layers to classify the input image and present the features in that image. From the comparative analysis it is adopted the suitable model for prediction based on their accuracy level.

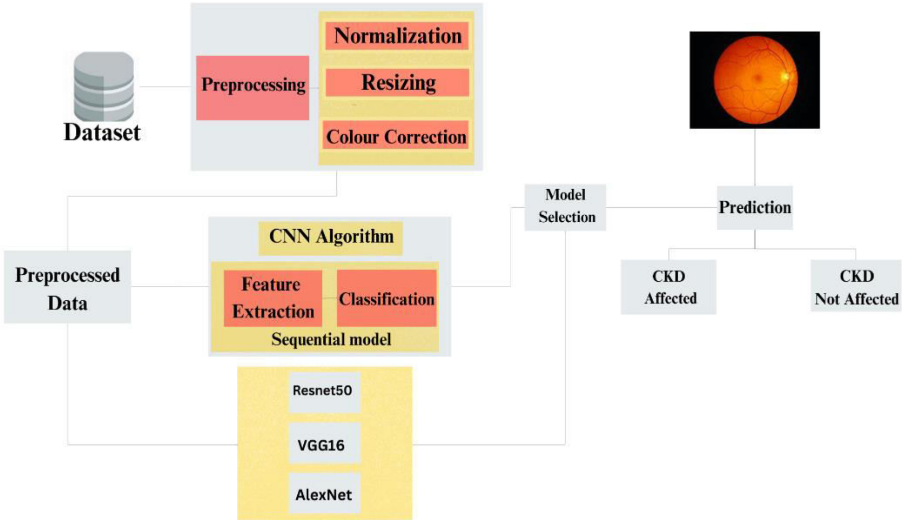
System Deployment: The study proposes a web service that displays the deployed model to the end-users after the completion of data training. The trained CNN model is converted into a deployable format compatible with the chosen deployment environment. Deployed the model to the chosen deployment environment, ensuring that it is accessible web services. The web service should accept the retinal images as input data and perform the best CNN algorithm model chosen from model selection, to show the predicted result.

Prediction: After deploying a model in a Chronic Kidney Disease (CKD) detection system, testing it involves submitting an input image from the dataset to the system. First is to be preprocessed to match the training conditions of the model, such as resizing or enhancing contrast. The system then analyses the image using the deployed model to predict whether CKD is present. The output is typically a classification (CKD or no CKD) indicating the confidence of the prediction, helping clinicians make informed decisions.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205263 |



IV. PSEUDO CODE

Building the ResNet50 pre-trained model:

```
class ResNet(nn.Module):
    def __init__(self, model_name, num_classes):
        super(ResNet, self).__init__()
        self.model_name = model_name
        resnet = models.resnet50(pretrained=True)
        self.features = nn.Sequential(*list(resnet.children())[:-1])  # Remove last layer (classification layer)
        self.fc = nn.Linear(resnet.fc.in_features, num_classes)

    def forward(self, input):
        features = self.features(input)
        features = features.view(features.size(0), -1)
        output = self.fc(features)
        return output
```

Building the VGG16 pre-trained model:

```
class VGG(nn.Module):
    def __init__(self, model_name, num_classes):
        super(VGG, self).__init__()
        self.model_name = model_name
        vgg = models.vgg19(pretrained=True)
        self.features = nn.Sequential(*list(vgg.features.children()))  # Use only the feature extraction layers

        self.avgpool = nn.AdaptiveAvgPool2d((7, 7))  # Adjust the dimensions of the feature maps
        self.fc = nn.Linear(512 * 7 * 7, num_classes)

    def forward(self, input):
        features = self.features(input)
        features = self.avgpool(features)
        features = features.view(features.size(0), -1)
        output = self.fc(features)
        return output
```

Building the Alexnet pre-trained model:

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205263 |

```
class AlexNet(nn.Module):
    def __init__(self, model_name, num_classes):
        super(AlexNet, self).__init__()
        self.model_name = model_name
        alexnet = models.alexnet(pretrained=True)
        self.features = nn.Sequential(*list(alexnet.features.children()))  # Use only the feature extraction layers

        self.avgpool = nn.AdaptiveAvgPool2d((6, 6))  # Adjust the dimensions of the feature maps
        self.fc = nn.Linear(256 * 6 * 6, num_classes)

    def forward(self, input):
        features = self.features(input)
        features = self.avgpool(features)
        features = features.view(features.size(0), -1)
        output = self.fc(features)
        return output
```

Prediction using ResNet50 model:

```
model = ResNet('ResNet',num_classes)
model.to(device)
   train_losses, train_accs, valid_accs = train_model(model, train_dl, valid_dl, num_epochs)
   print()
   print()
   print()
   # plot loss and validation curves
   plot_curves(train_losses, train_accs, valid_accs, num_epochs)
   # saving the best weights to be applied to the test dataset
best_model_state = torch.load('/content/ResNet_best_model.pth')
model = ResNet('ResNet', num_classes)
model.load_state_dict(best_model_state)
model.to(device)
model.eval()
   Visualize results
visualize_predictions(model, test_dl, device, test_ds.classes)
print()
print()
generate_confusion_matrix(model, test_dl, device, num_classes)
print()
print()
confusion_mat = generate_confusion_matrix(model, test_dl, device, num_classes)
print()
print()
display_confusion_matrix(confusion_mat, test_ds.classes)
print()
   print()
generate_confusion_matrix_with_metrics(model, test_dl, device, num_classes)
```

Prediction using VGG16 model:

```
model = VGG('VGG', num_classes)
model.to(device)
train_losses, train_accs, valid_accs = train_model(model, train_dl, valid_dl, num_epochs)
Print()
print()
print()
# plot loss and validation curves
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205263 |

```
plot_curves(train_losses, train_accs, valid_accs, num_epochs)
# saving the best weights to be applied to the test dataset
best_model_state = torch.load('/content/VGG_best_model.pth')
model = VGG('VGG', num_classes)
model.load_state_dict(best_model_state)
model.to(device)
model.eval()
# Visualize results
visualize_predictions(model, test_dl, device, test_ds.classes)
print()
print()
generate_confusion_matrix(model, test_dl, device, num_classes)
print()
print()
confusion_mat = generate_confusion_matrix(model, test_dl, device, num_classes)
print()
print()
display_confusion_matrix(confusion_mat, test_ds.classes)
print()
print()
generate_confusion_matrix_with_metrics(model, test_dl, device, num_classes)
```

Prediction using Alexnet model:

```
model = AlexNet('AlexNet', num_classes)
model.to(device)
train_losses, train_accs, valid_accs = train_model(model, train_dl, valid_dl, num_epochs)
print()
print()
print()
# plot loss and validation curves
plot_curves(train_losses, train_accs, valid_accs, num_epochs)
# saving the best weights to be applied to the test dataset
best_model_state = torch.load('/content/AlexNet_best_model.pth')
model = AlexNet('AlexNet', num_classes)
model.load_state_dict(best_model_state)
model.to(device)
model.eval()
# Visualize results
visualize_predictions(model, test_dl, device, test_ds.classes)
print()
print()
generate_confusion_matrix(model, test_dl, device, num_classes)
print()
print()
confusion_mat = generate_confusion_matrix(model, test_dl, device, num_classes)
print()
print()
display_confusion_matrix(confusion_mat, test_ds.classes)
print()
print()
generate_confusion_matrix_with_metrics(model, test_dl, device, num_classes)
```

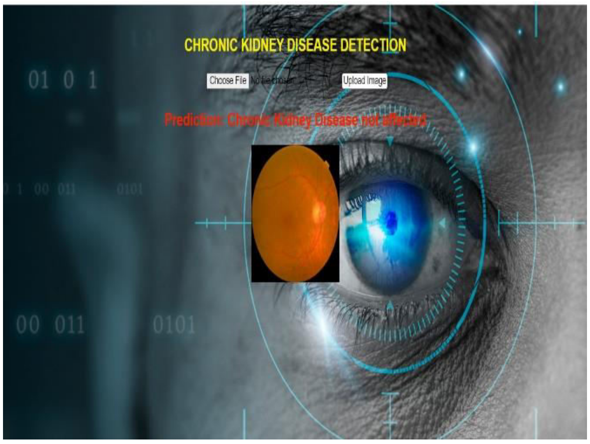
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

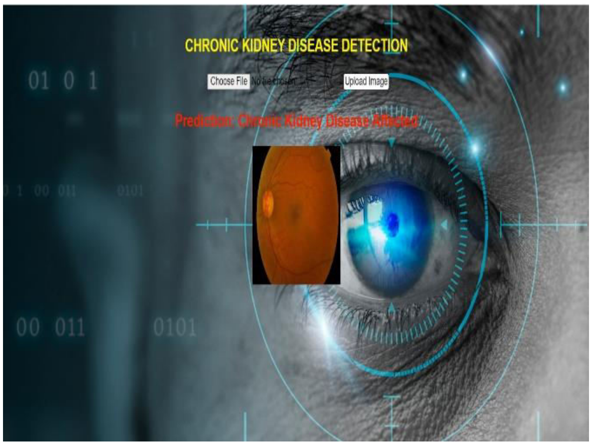
|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205263 |

V. SIMULATION RESULTS

The CKD detection UI for retinal images employs a user-friendly interface for uploading images. After processing through the CNN model in which the ResNet50 model rated an accuracy of 0.97, the VGG16 model rated an accuracy of 0.93 and the Alexnet model rated an accuracy of 0.94. From the comparative analysis the ResNet50 model is used for prediction as it has the highest accuracy rate that results indicating CKD likelihood are displayed. The interface is intuitive, allowing healthcare professionals to quickly assess patient risk based on retinal images. The final UI for CKD detection using retinal images offers a user-friendly interface where users can upload retinal images for analysis. Upon submission, the system utilizes deep learning algorithms to analyze the images and provide instant results indicating the likelihood of CKD presence. The UI displays clear and concise results, enabling quick interpretation and facilitating timely intervention for patients at risk of chronic kidney disease.





VI. CONCLUSION AND FUTURE WORK

Utilizing CNN models for CKD detection via retinal image analysis holds great promise in medical diagnostics. These models, trained on extensive datasets, can identify CKD-related abnormalities with high accuracy, sometimes surpassing human performance. Benefits include rapid processing, scalability, and non-invasive operation, making them suitable for clinical workflows. Automated retinal image analysis streamlines diagnosis, allowing prompt CKD risk identification and treatment. Challenges include the need for diverse datasets, model interpretability, and bias mitigation. Ongoing research aims to enhance model accuracy, validate performance across populations, and achieve regulatory approval. Advancements in this field could significantly improve CKD management and patient outcomes.

In future enhancing model interpretability by integrating explainable AI techniques could provide insights into the features driving predictions. Additionally, leveraging larger and more diverse datasets, including longitudinal data, could improve model generalizability. Incorporating multi-modal data fusion, such as combining retinal images with clinical data or genetic information, may enhance predictive accuracy. Furthermore, exploring transfer learning from related tasks or domain adaptation techniques could facilitate model adaptation to different populations or healthcare settings. Finally, deploying the developed models in real-world clinical settings and assessing their impact on patient outcomes is essential for validation and further refinement.

REFERENCES

1. Liang, Ping, et al. (2023) ‘Deep learning identifies intelligible predictors of poor prognosis in chronic kidney disease’, IEEE Journal of Biomedical and Health Informatics.
2. Qin, Jiongming, et al. (2019) ‘A machine learning methodology for diagnosing chronic kidney disease’, IEEE Access 8: 20991-21002.
3. Ramesh, R., and S. Sathiamoorthy et al. (2023) ‘Deep Learning with Heuristic Optimization Driven Diabetic Retinopathy Detection on Fundus Images’, 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC). IEEE, 2023.
4. Patil, Smitha, and Savita Choudhary et al. (2023) ‘Hybrid classification framework for chronic kidney disease prediction model’, The Imaging Science Journal: 1-15.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205263 |

5. Ali, Ghulam, et al. (2023) ‘A hybrid convolutional neural network model for automatic diabetic retinopathy classification from fundus images’, IEEE Journal of Translational Engineering in Health and Medicine (2023).
6. Elkholy, Shahinda Mohamed Mostafa, Amira Rezk, and Ahmed Abo El Fetoh Saleh, et al (2021) ‘Early prediction of chronic kidney disease using deep belief network’, IEEE Access 9: 135542-135549.
7. Hossain, Mohammad Sakib, et al. (2023) ‘Kidney Disease Detection from CT Images using a customized CNN model and Deep Learning’, 2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS). IEEE, 2023.
8. Sri, Vemu Santhi, and GR Jothi Lakshmi, et al. (2023) ‘Detection Analysis of Abnormality in Kidney using Deep Learning Techniques and its Optimization’, 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT). IEEE, 2023.
9. Khan, Bilal, et al. (2020) ‘An empirical evaluation of machine learning techniques for chronic kidney disease prophecy’, IEEE Access 8 55012-55022.
10. Liu, Tieyuan, et al. (2021) ‘A novel diabetic retinopathy detection approach based on deep symmetric convolutional neural network’, IEEE Access 9 :160552-160558.
11. Ali, Dalia, Mohamed Aouf, and M. Fouad, et al. (2019) ‘Diabetic exudate detection in color retinal images’, International Journal of Computers & Technology 19: 7510-7518.
12. Mitani, Akinori, Naama Hammel, and Yun Liu, et al. (2021) ‘Retinal detection of kidney disease and diabetes’, Nature biomedical engineering 5.6: 487-489.
13. Saranya, P., et al. (2021) ‘Red lesion detection in color fundus images for diabetic retinopathy detection’, Proceedings of International Conference on Deep Learning, Computing and Intelligence: ICDCI 2021. Singapore: Springer Nature Singapore, 2022.
14. C. Anusha, Deshmukh Sanket Surendra. et al. (2021) ‘Early prediction of chronic kidney disease using deep belief network’, IEEE Access 9: 135542-135549.
15. Swain, Debabrata, et al. (2023) ‘A robust chronic kidney disease classifier using machine learning’, Electronics 12.1: 212.