EYE DISEASE DETECTION BY IMAGE CLASSIFICATION

Husanboy Olimboyev Qahramon o'g'li
Urgench State University, Department of Computer Sciences
E-Mail: olimboyevhusanboy@gmail.com

Abstract This article explores the application of machine learning in ocular healthcare through the classification of eye images. We delve into the process of utilizing Python and various libraries to implement machine learning algorithms for image classification, focusing on distinguishing between eight key classes of eye images: Normal, Diabetes, Glaucoma, Cataract, Age-related Macular Degeneration, Hypertension, Pathological Myopia, and Other diseases. The article highlights the significance of this approach in aiding healthcare professionals in early diagnosis and treatment planning.

Keywords: Normal, Diabetes, Glaucoma, Cataract, Age-related Macular Degeneration, Hypertension, Pathological Myopia, Kaggle.

Introduction It covers data preprocessing, model selection, training, and evaluation, highlighting the use of libraries such as TensorFlow, PyTorch, and OpenCV. Real-world examples and code snippets are included to aid understanding. The effectiveness of machine learning models in ocular decision-making relies heavily on their performance and generalization capabilities. This section discusses various metrics and techniques used to evaluate and validate the models, ensuring their reliability and accuracy in clinical settings. Ocular diseases pose a significant health challenge globally, requiring timely diagnosis and treatment to prevent vision loss. Traditional diagnostic methods often rely on subjective assessments, leading to variability in diagnoses. However, recent advancements in machine learning offer a promising solution by enabling the automated classification of eye images with high accuracy. Ocular diseases, including but not limited to Glaucoma, Cataract, and Diabetic Retinopathy, are major causes of visual impairment and blindness globally.

Early detection and accurate diagnosis are crucial for effective management and prevention of irreversible vision loss. Machine learning, a subset of artificial intelligence, has emerged as a powerful tool in healthcare, particularly in the analysis of medical images. The integration of machine learning models in clinical practice has the potential to revolutionize ocular healthcare. By providing automated and accurate diagnoses, these models can assist healthcare professionals in making informed decisions and designing personalized treatment plans. This section explores the practical applications and implications of machine learning in ocular decision-making. Despite the advancements in machine learning, several challenges remain, such as the need for large and diverse datasets, interpretability of models, and regulatory considerations. This section discusses these challenges and proposes future directions for research and development in the field of ocular decision-making with image classification.

METHODOLOGY

We implemented machine learning algorithms using Python, leveraging custom libraries and frameworks such as TensorFlow, PyTorch, Matplotlib, OpenCV, pandas, numpy, and seaborn. The process involved dividing the images into pixels, classifying them based on interdependence, and splitting them into training and testing sets. This approach allowed us to train the model on a subset of the data and validate its performance on unseen data, ensuring its effectiveness in real-world applications. This section provides a detailed overview of the eight classes of eye images: Normal, Diabetes, Glaucoma, Cataract, Age-related Macular Degeneration, Hypertension, Pathological Myopia, and Other diseases. Each disease is characterized by specific features and manifestations, highlighting the importance of accurate classification for timely intervention.

Machine learning algorithms, particularly deep learning models, have shown remarkable performance in image classification tasks. By analyzing patterns and features within eye images, these algorithms can differentiate between different ocular diseases with high accuracy. This section discusses the key principles and methodologies involved in using machine learning for ocular decision-making. This section provides a step-by-step guide to implementing machine learning algorithms for ocular image classification using Python. The normal eye and damaged eyes with the types of eye diseases caused by diabetes are shown in fig.1. In this research, it is aimed to investigate the effectiveness of transfer learning techniques for the classification of eye diseases caused by diabetes. Section 2 analyzes some related works, while the proposed method is explained in section 3. Conclusion is provided by section 4

DISCUSSION

The prevalent approaches for classification of eye diseases involve learning based structures and transfer learning, both of which employ convolutional neural networks (CNNs). The present study focuses on classifying retina images based on types of eye diseases, which is accomplished by enhancing a CNN model through the use of advanced transfer learning techniques.

Data Collection

The dataset used in this study is provided by the Kaggle and compares of retinal images belonging to eight distinct classes, namely Normal, Diabetes, Glaucoma, Cataract, Age-related Macular Degeneration, Hypertension, Pathological Myopia, and Other diseases. Each class consists of minimum 250 images, sourced from a variety of repositories such as Ocular Recognition and HRF. Figure 1 displays a set of randomly selected retinal images of the human eye, each labeled with its corresponding eye disease class.

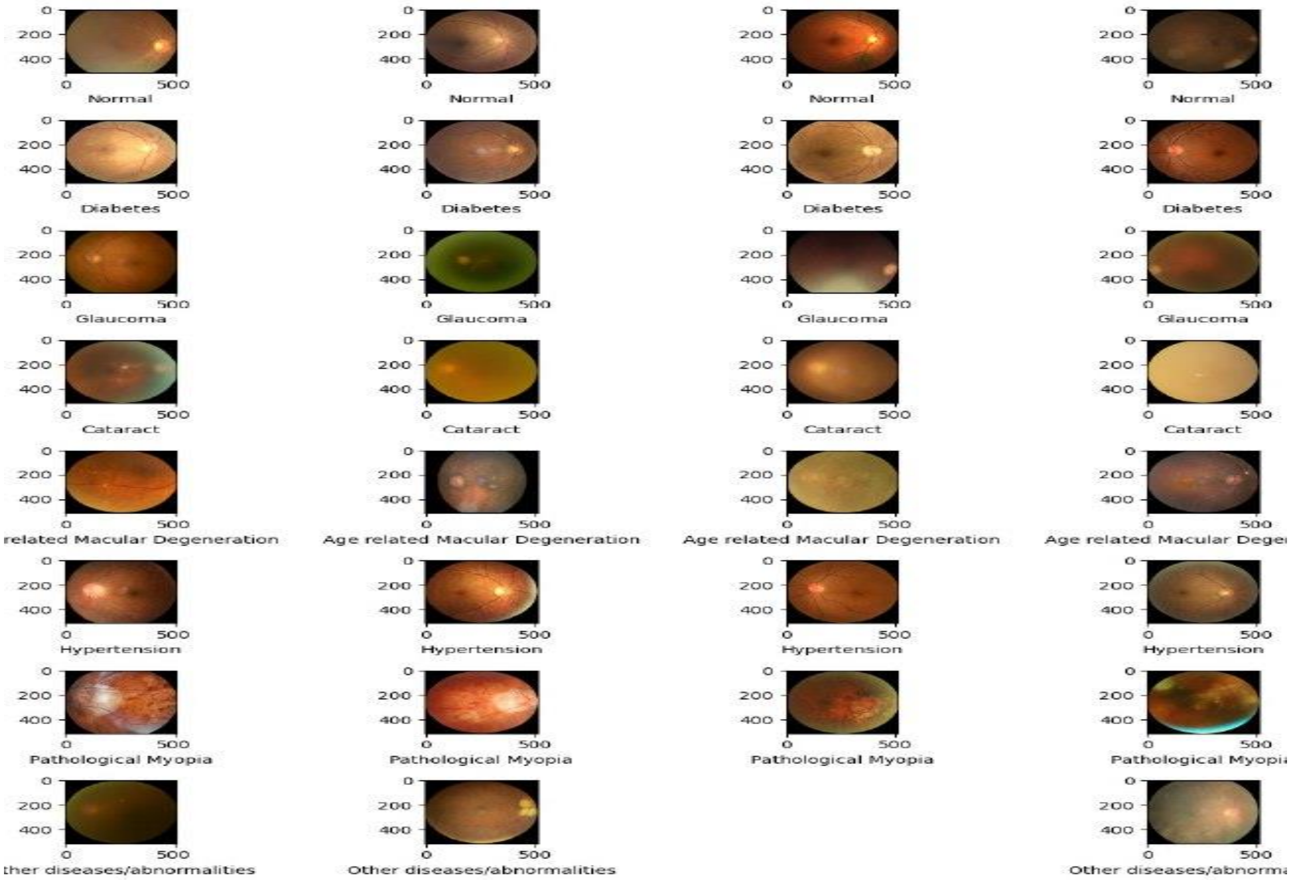

Fig. 1. Random images in the training set

This dataset contains about 6000 training images categorized into eight classes: Normal, Diabetes, Glaucoma, Cataract, Age-related Macular Degeneration, Hypertension, Pathological Myopia, and Other diseases. The distribution of the number of images of each class can be seen in fig. 2.

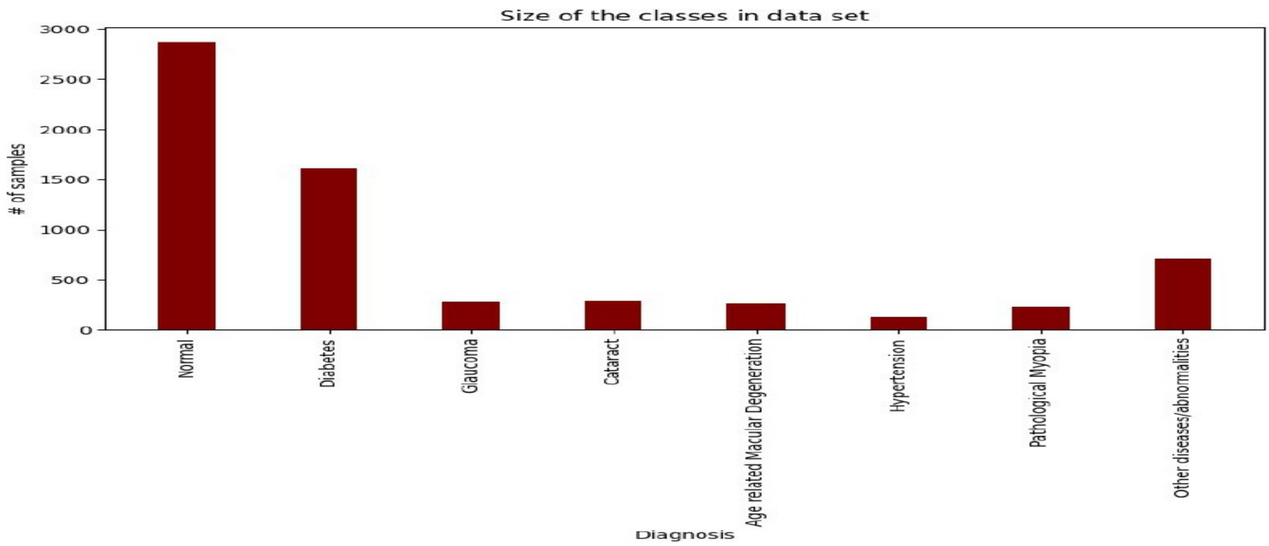

Fig. 3. Distribution of classes

Preprocessing Images

The dataset is divided into batches with size 64 to gains efficient utilization of computational resources and enables the model to be trained on larger datasets without requiring excessive memory. Next step is splitting the dataset into three parts: train (47 batches - 70%) set, validation (13 batches - 20%) set, and test (7 batches - 10%) set. The train set was used to train the model, the validation set was used to tune the hyperparameters and prevent overfitting, and the test set was used to evaluate the final performance of the model. This partitioning allowed to assess the model's ability to generalize to unseen data and provided a more accurate estimate of the model's performance on real-world scenarios.

RESULTS

Overview of Outcomes

Based on our research, we solved the class imbalance problem by taking the same number of images. There was a huge disparity between the classes in the ODIR dataset. By following this method, we are able to significantly improve our accuracy when the number of images is much lower. The relative metrics, accuracy loss graphs, and other equivalent indications of the performance evaluation methodologies were then examined and graphically shown. Using the VGG-19 architecture, we demonstrated our model's ability to accurately predict a particular condition. We used the confusion matrix on the test to show how accurate this model can predict.

Performance Evaluation

In our experiment, the CNN-based architecture VGG-19 was used to assess model performance. In the data sequencing and splitting part, first the image is taken from the dataset and the data are converted into train labels and target labels. The scikit-learn library's training-test split method was used. The data were divided into a 70 : 20 :10 ratio, with 70% of the data being utilized for training and validation 20% set and 10% for testing. The performance metrics of both models are presented in this section, as well as their prediction capabilities. Here are some results for each class. In the dataset, we only have 207 cases of glaucoma. To balance this, we took 206 normal cases so that the model does not overfit. After sampling the data from the dataset, we passed the data into the pretrained VGG-19 and got a training accuracy and loss. In fig.4 the initial accuracy for training was about 0.6 and the validation was 0.80. With each epoch, the accuracy of training and validation increases, and model loss decreases. After running 5 epochs, the model achieved training and validation accuracy of 0.97 and 0.90, respectively. Finally, the model training accuracy was about 1.0 and the validation accuracy was 0.92.

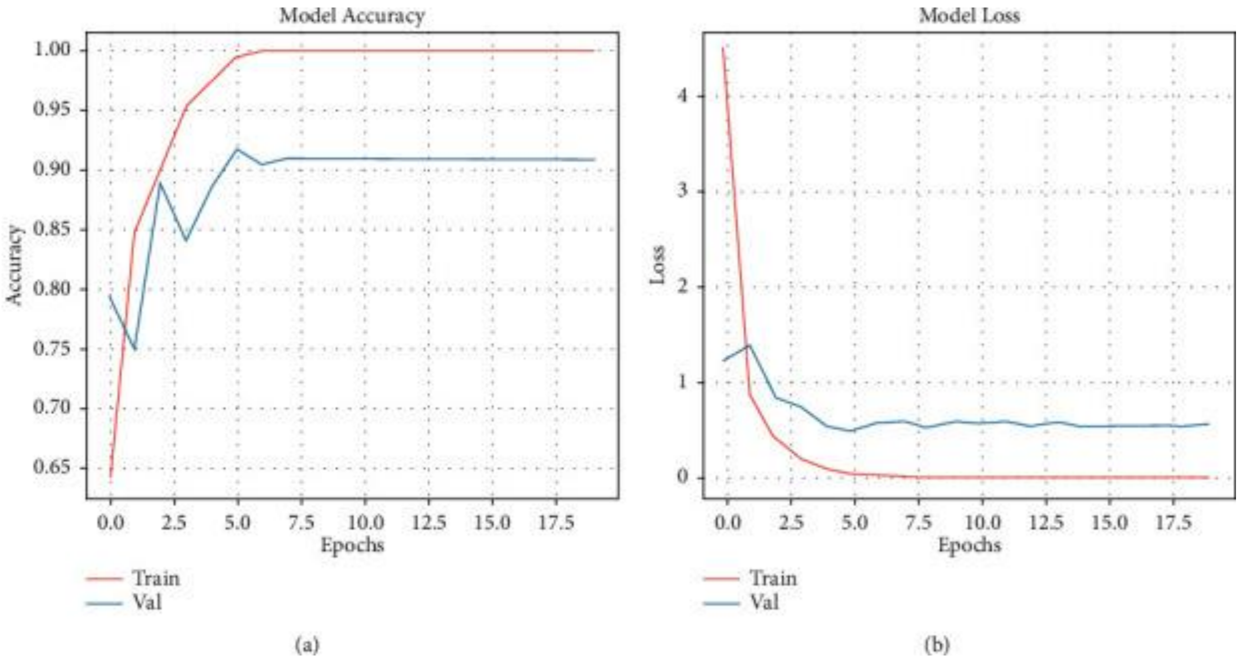

Fig. 4. Model accuracy and loss for N versus G.

We only have 94 hypertension cases in the dataset; to balance this, we took 95 normal cases so that the model does not overfit. We sampled data from the dataset and fed them into a pretrained VGG-19 to calculate training accuracy and loss.

In fig.5, the initial accuracy for training was about 0.59 and the validation was 0.64. With each epoch, the accuracy of training and validation increases, and model training and validation loss decreases. After running 5 epochs, the model achieved training and validation accuracy of 0.98 and 0.88, respectively. Finally, the model training accuracy was about 1.0 and the validation accuracy was 0.90.

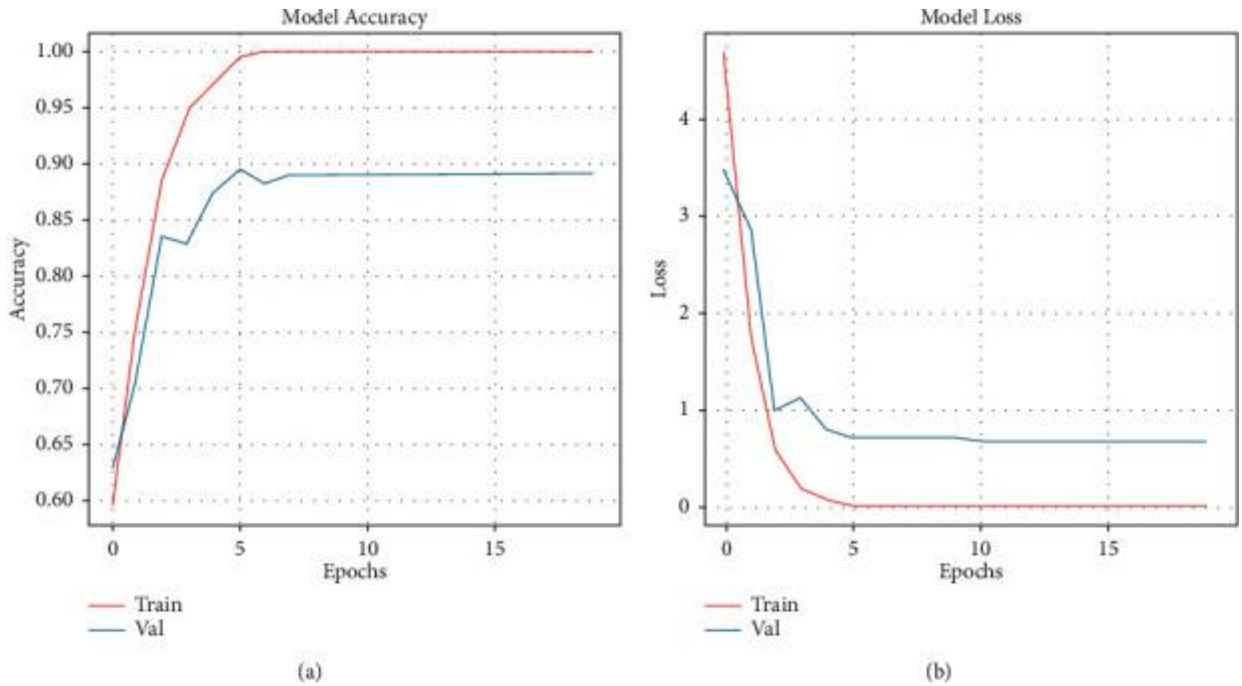

Fig. 5. Model accuracy and loss for N versus H.

We only have 177 pathological myopia cases in the dataset, so we added 175 normal cases to ensure that the model does not overfit. To calculate training accuracy and loss, we sampled data from the dataset and fed them into a pretrained VGG-19.

In fig.6, the initial accuracy for training was about 0.85 and the validation was 0.90. With each epoch, the accuracy of training and validation increases, and model loss decreases. After running 5 epochs, the model achieved training and validation accuracy of 0.96 and 0.98, respectively. Finally, the training accuracy was about 1.0 and the validation accuracy was 0.99.

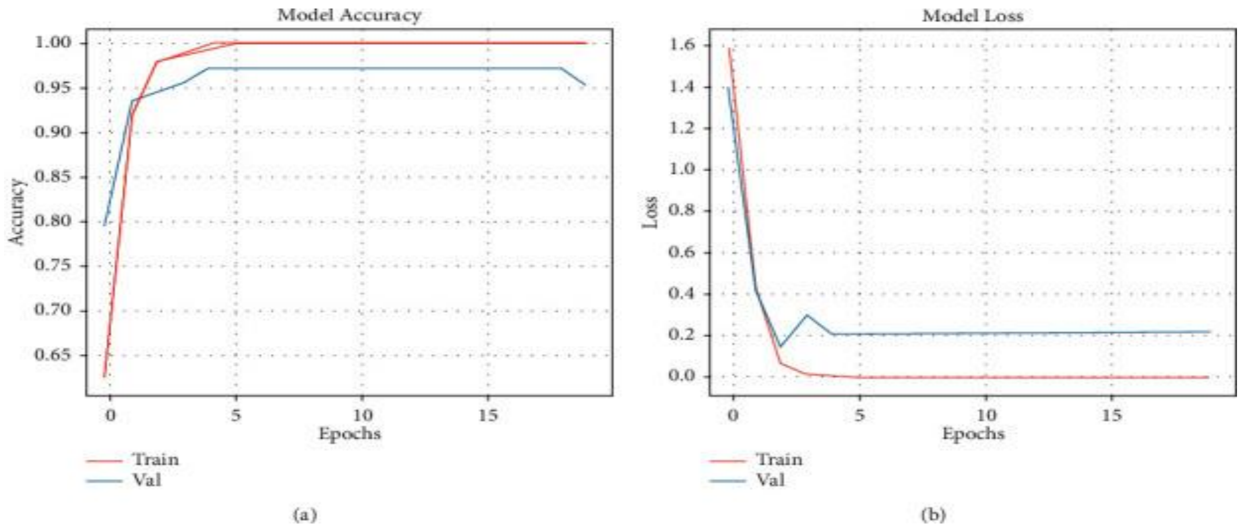

Fig. 6.Model accuracy and loss for N versus M.

We only have 177 pathological myopia cases in the dataset, so we added 175 normal cases to ensure that the model does not overfit. To calculate training accuracy and loss, we sampled data from the dataset and fed them into a pretrained VGG-19.

In fig.7, the initial accuracy for training was about 0.4 and the validation was 0.65. With each epoch, the accuracy of training and validation increases, and model loss decreases. After running 5 epochs, the model achieved training and validation accuracy of 0.96 and 0.86, respectively. Finally, the training accuracy was about 1.0 and the validation accuracy was 0.90.

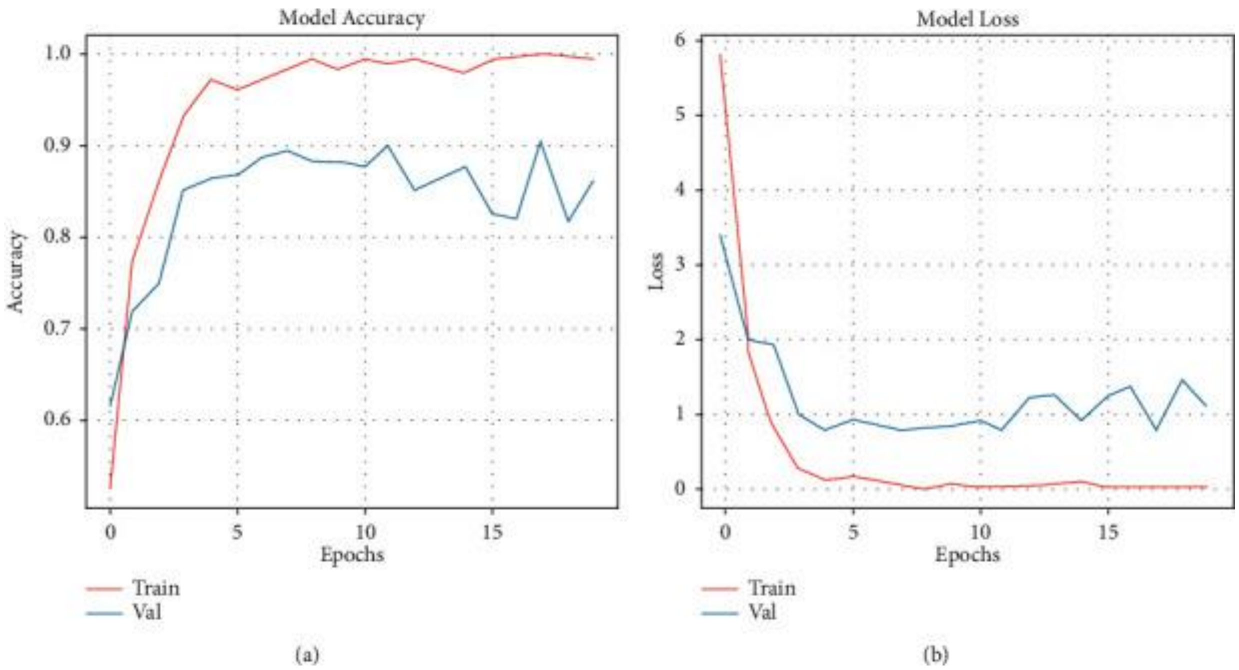

Fig. 7. Model accuracy and loss for N versus O.

Evaluate Accuracy Metrics

For measuring goodness, we will use some metrics relating to accuracy in determining whether a particular image represents a disease. After all, we have used classification models. So we will use the most widely used metrics for classification problems.

Our experiments demonstrated the effectiveness of the machine learning model in classifying eye images into the eight predefined classes. The model achieved high accuracy rates, indicating its potential in aiding healthcare professionals in diagnosing ocular diseases. Furthermore, the model's ability to generalize to unseen data suggests its robustness and reliability in diverse clinical settings.

CONCLUSION

In conclusion, machine learning offers a promising avenue for enhancing ocular healthcare through accurate and timely image classification. By leveraging the power of Python and various libraries and frameworks, healthcare professionals can improve diagnostic accuracy and patient outcomes, ultimately leading to a brighter future for individuals with ocular diseases. The application of machine learning in ocular healthcare has the potential to revolutionize the field by providing automated, accurate, and timely diagnoses. By leveraging Python and a range of libraries and frameworks, we have shown that machine learning algorithms can effectively classify eye images, paving the way for enhanced patient care and outcomes in ocular healthcare.

REFERENCES.

[1] U. Salaev, E. Kuriyozov, G. Matlatipov, "Design and Implementation of a Tool for Extracting Uzbek Syllables," in 2023 IEEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE), 2023, pp. 1750-1755.

[2] J. Mattiev, U. Salaev, and B. Kavsek, “Word Game Modeling Using Character-Level N-Gram and Statistics,” *Mathematics*, vol. 11, no. 6, p. 1380, 2023.

[3] M. Sharipov, E. Kuriyozov, O. Yuldashev, and O. Sobirov, “UzbekTagger: The rule-based POS tagger for Uzbek language,” *arXiv preprint arXiv:2301.12711*, 2023.

[4] U. Salaev, E. Kuriyozov, and C. Gómez-Rodríguez, “SimRelUz: Similarity and Relatedness scores as a Semantic Evaluation Dataset for Uzbek Language,” in *1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages, SIGUL 2022 - held in conjunction with the International Conference on Language Resources and Evaluation, LREC 2022 - Proceedings*, 2022, pp. 199–206. [Online]. Available: www.scopus.com

[5] J. Mattiev, J. Sajovic, G. Drevenšek, and P. Rogelj, “Assessment of Model Accuracy in Eyes Open and Closed EEG Data: Effect of Data Pre-Processing and Validation Methods,” *Bioengineering*, vol. 10, no. 1, p. 42, 2023.

[6] Suetens, Paul. Fundamentals of medical imaging. Cambridge university press, 2017.

[7] Hounsfield, Godfrey N. "Computed medical imaging." Science 210.4465 (1980): 22-28.

[8] Hendee, William R., and E. Russell Ritenour. Medical imaging physics. John Wiley & Sons, 2003.