OptiCare AI: Smart At-Home Eye Testing System

Yuvraj Ware
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India
yuvraj.ware24@vit.edu

Dr. Devata Anekar
Department of Information Technology
Vishwakarma Institute of Technology,
Pune, India
devata.anekar@vit.edu

Riya Wankhede
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India
riya.wankhede24@vit.edu

Sampada Yanpallewar
Department of Information Technology
Vishwakarma Institute of Technology Pune, India
sampada.yanpallewar24@vit.edu

Yash Agrawal
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India
yash.agrawal24@vit.edu

Tanishq Shinde
Department of Information Technology
Vishwakarma Institute of Technology Pune, India
tanishq.shinde241@vit.edu

***Abstract-* VisionCare is an AI-based healthcare assistance web-platform designed specifically to assist in the identification of eye disease early diagnosis as well as provide assistance in vision healthcare service delivery. The system uses the convolutional neural network algorithm in analyzing the eye image uploaded by users to identify the condition associated with diseases such as cataracts, bulging eyes, crossed eyes, uveitis, and normal eyes. The prediction module of the system runs on a web-based system that incorporates Flask APIs. Apart from disease prediction, other features included in the system are an AI-based health care advice chatbot, nearby doctor finder based on the OpenStreetMap APIs, and amsler grid testing for preliminary vision testing. The frontend design of the system is done using HTML, CSS, and JavaScript, whereas the backend services use Flask, FastAPI, and Node.js programming languages. SQLite and MongoDB are the database management systems employed in storing the patient data.**

.Keywords— Artificial Intelligence, Deep Learning, Convolutional Neural Network, Eye Disease Detection, Healthcare Assistance, Flask, FastAPI, OpenStreetMap, Medical Image Processing, Web Application.

I. INTRODUCTION

Eye diseases are among the significant contributors to blindness and visual impairments globally. Eye diseases such as cataract, crossed eyes, uveitis, and other diseases associated with vision may have a negative impact on people's lives unless detected and treated early enough. Traditional methods of diagnoses involve use of sophisticated medical tools and consultations by professional eye doctors, a process that might be impossible in remote areas [14].

There have been breakthroughs in the use of technologies such as Artificial Intelligence (AI) and Deep Learning in the health sector recently. Such techniques find applications in the analysis of medical images and the prediction of diseases due to their capabilities of detecting patterns within complex sets of data [1]. CNNs are highly effective in image classification problems and often used to detect disease using images [2], [3].

VisionCare is a proposed system that aims to be a healthcare assistance system built with the application of AI. It is designed for the purpose of aiding early detection of eye diseases and providing healthcare assistance to users. It is capable of taking image uploads from users and generating predictions along with their confidence levels. CNN algorithm will be applied to classify different types of eye diseases like Cataracts, Bulging Eyes, Crossed Eyes, Uveitis, and Normal Eye.

Furthermore, VisionCare can provide healthcare assistance such as an AI-driven Medical Assistance Chatbot, a Nearby Doctor Finder based on OpenStreetMap APIs, and Amsler Grid Test functionality to test eyes. The proposed system can be developed with the use of HTML, CSS, JavaScript, Flask, FastAPI, and Node.js while using SQLite and MongoDB as database management systems for maintaining health and prediction histories [7], [8], [10].

The primary goal of the proposed project is to design and develop a lightweight yet scalable healthcare platform with the combination of AI and web technologies.

II. LITERATURE REVIEW

The field of Artificial Intelligence and Deep Learning has resulted in many improvements in healthcare, including those pertaining to medical imaging and disease prediction systems. A few researchers have developed automated detection systems for eye diseases by utilizing CNNs and image processing techniques. Such systems assist in minimizing human involvement while diagnosing these diseases, which can help detect the eye diseases at an early stage.

CNNs are extensively employed for classifying images since such networks can extract essential features automatically and produce highly accurate predictions about images. For instance, Krizhevsky et al. have introduced deep CNN architectures, which proved to be extremely effective for image classifications tasks and were widely adopted in healthcare [2]. Likewise, Simonyan and Zisserman have designed very deep CNN networks, which led to enhanced recognition abilities in images [3].

Approaches Proposed by Other Studies

1. CNN-Based Eye Disease Detection System

Some studies propose using the CNN algorithm for detection of eye-related diseases from the images taken using the ophthalmoscope. Such a system offers a high level of accuracy in terms of detecting eye diseases like cataract, glaucoma, and others [1], [2].

Constraint:

They all deal with the problem of prediction of a disease, but offer no additional healthcare assistance features.

2. AI-powered Healthcare Chatbots

Some studies propose developing AI chatbots capable of answering medical questions using NLP techniques [15].

Constraint:

The majority of such chatbots are designed to work in different areas and are not coupled with disease prediction functionality.

3. Telemedicine

Telemedicine application platforms allow users to contact healthcare specialists via internet.

Constraint:

Such an approach implies the need for manual consultation and cannot provide any diagnostic features due to the lack of AI capabilities.

4. Doctor Finder Applications

Doctor finder applications allow finding hospitals and medical centers nearby thanks to OpenStreetMap and other map APIs [10].

Constraint:

Such tools operate independently and do not interact with disease prediction mechanisms at all.

5. Web-Based Healthcare Application

Healthcare applications built using such technologies as Flask, FastAPI, and MongoDB [8].

- **Research Gap**
  - Currently, available systems concentrate exclusively on one functionality for health care.
  - Very few systems offer services including disease prediction, chatbot help, doctor finder, and vision test all at once.
  - Very few platforms are lightweight or web-enabled.
  - Current systems for healthcare are either too costly to operate or need dedicated applications.
  - There is limited availability of integrated systems for AI-based eye healthcare help.

VisionCare, on the other hand, incorporates AI-based disease prediction of eyes, healthcare chatbot support, doctor finding, and vision testing within a lightweight web-enabled application.

III. System design and Methodology

The VisionCare system has been developed as an AI-based healthcare assistant application where eye disease predictions, healthcare chatbots, nearby doctor search services, and vision tests can all be integrated into one web application. This application has been created using Deep Learning, Web Technologies, REST APIs, and database technology, making it quite light-weight.

The methodology involved for this particular application is related to the following activities: dataset creation, image pre-processing, CNN model building, backend designing, frontend design, database integration, system testing, and results analysis.

➢ **Principle of Operation of the System:**

VisionCare operates through the analysis of eye images that are uploaded using a trained Convolutional Neural Network (CNN) algorithm. The prediction algorithm analyzes certain patterns and features from the eye image and classifies them in disease categories such as Cataracts, Bulging Eyes, Crossed Eyes, Uveitis, and Normal Eyes.

The system starts when a person uploads the eye image in the frontend. The uploaded image is sent to the backend (Flask) where some processing operations such as resizing, normalization, and converting into an array are done. The image undergoes training in the trained CNN algorithm for predictions.

Once the prediction is done, the backend produces the disease name and the confidence value which are sent back to the frontend via REST APIs. The frontend dynamically updates the prediction results in real-time. Other healthcare modules include the chatbot, doctor finder, and Amsler grid test.

➢ **Data Set Collection and Processing:**

Data sets for the disease prediction model are categorised eye image data obtained from freely available health care image sources. The data set has several eye diseases to be classified.

The data set is split into three sections:

- Training data set
- Validation data set
- Testing data set

The training data set is used for learning, the validation data set is used for parameter tuning, and the testing data set is used for model evaluation.

Table.I. Dataset Distribution:

Disease Category	Training Images	Validation Images	Testing Images
Cataracts	450	100	80
Bulging Eyes	420	90	75
Crossed Eyes	400	85	70

Uveitis	430	95	78
Normal Eyes	500	110	90

➢ **Image Preprocessing:**

Image preprocessing is crucial since the original medical images might be noisy, of different sizes, and have variable lighting conditions. Effective preprocessing ensures improved model efficiency and accuracy.

The following image pre-processing methods are employed:

- Resizing of images to a specific size
- Pixel scaling
- Data augmentation
- Noise reduction
- Conversion of images to arrays

Data augmentation methods include rotating images, zooming in on images, and flipping images to diversify the data.

➢ **Implementation of the Back End:**
The back end of the VisionCare system is developed through the use of Flask, FastAPI, and Node.js.

- **Flask Back End:**

The Flask server is responsible for carrying out the process of machine learning prediction on the input images.

Tasks Performed by Flask Back End:

- Handling image uploads
- Preprocessing images
- Executing the CNN model
- Predicting the disease
- Generating JSON responses
- FastAPI Back End

FastAPI is used for developing REST APIs and functionality for the healthcare industry.

Features of FastAPI:

- Chatbots with AI
- Doctor finder APIs
- Interfacing with external APIs
- Request handling with high speed
- CORS support

The use of FastAPI allows for higher speed and efficiency in API development.

- **Node.js Back End**

The Node.js server handles tasks related to the users and backend communication.

Functions of Node.js Server:

- Authentication of the user
- Handling routes
- Managing the user's history
- Communication with the database

➢ **Frontend Development:**

Frontend development takes place by coding in HTML, CSS, and JavaScript. The frontend acts as an easy-to-use interactive interface for users.

Frontend Components:

- Eye image upload interface
- Result output for disease prediction
- Interface for AI chatbot
- Amsler Grid testing component
- Nearby doctor lookup interface

Backend APIs are accessed through the JavaScript component, allowing prediction results to be updated dynamically.

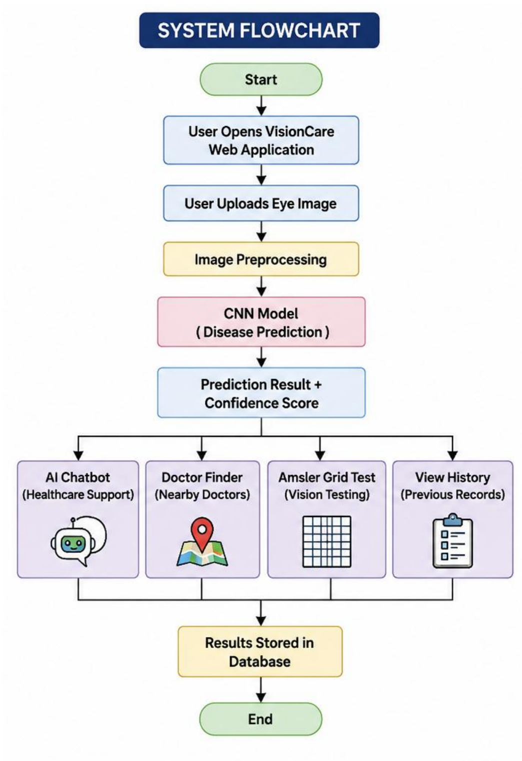

Fig.1. Working Flow of Project

➢ **Models in Project:**
The other healthcare features available in the VisionCare framework include the Nearby Doctor Finder, AI Chatbot, and Amsler Grid Testing. The Nearby Doctor Finder module employs OpenStreetMap APIs to identify nearby eye specialists, clinics, and hospitals based on user location. The AIs Chatbot module uses Groq APIs and natural language processing to assist users in their healthcare decision-making process. Further, the Amsler Grid Testing feature helps the user conduct a visual distortion test and stores healthcare records for early detection of visual disorders.

Table.II. Performance Analysis:

Module	**Average Response Time**
Disease Prediction	2–4 seconds
Chatbot Response	1–3 seconds

Doctor Finder	2 seconds
Database Storage	Less than 1 second

➢ **Workflow of Complete System**

Working flow of entire VisionCare system can be represented as follows:

- User visits the web portal.
- Eye image is uploaded using front-end interface.
- Image is pre-processed by Flask backend.
- Disease type is predicted using CNN model.
- Predictions and confidence score are shown.
- Chatbot and healthcare features are used by user.
- Find doctor feature provides nearby hospitals information.
- Results and predictions are saved to database.

The designed methodology has been able to incorporate AI and health-care technology into a single web portal.

IV. RESULTS AND DISCUSSIONS

Testing of the VisionCare system was carried out for predicting eye disease, healthcare assistance, and the general performance of the system. The prediction algorithm implemented using CNN technology performed accurate classification of uploaded eye pictures into disease types such as Cataracts, Bulging Eyes, Crossed Eyes, Uveitis, and Normal Eyes. Prediction results were obtained from the system in real-time together with the confidence score.

Frontend user interaction for uploading images, chatting with the chatbot, searching doctors, and Amsler grid tests was achieved. API calls by Flask and FastAPI backend services worked effectively. The databases used were also able to store healthcare information and prediction history.

The chatbot module was successfully able to handle queries regarding healthcare topics and increase the engagement level of the users. In the same way, the Nearby Doctor Finder module successfully fetched nearby ophthalmologists and healthcare facilities using APIs provided by OpenStreetMap. The Amsler Grid Testing module also performed efficiently for the analysis of vision distortions at the preliminary stage.

Through the experimental testing process, it was evident that there was accuracy in the predictions made with an efficient response time of the system. The implementation of the modular approach led to the enhanced scalability of the healthcare functionalities within the platform.

Table.III. Sample Prediction Results:

Eye Image Type	*Predicted Disease*	*Confidence Score*
Eye Sample 1	*Cataracts*	*95%*
Eye Sample 2	*Normal Eyes*	*97%*
Eye Sample 3	*Crossed Eyes*	*93%*
Eye Sample 4	*Uveitis*	*91%*

- **Home Page**

There was an easy navigation system on the home page for all the modules including disease prediction, AI chatbot, doctor finding, and Amsler grid test.

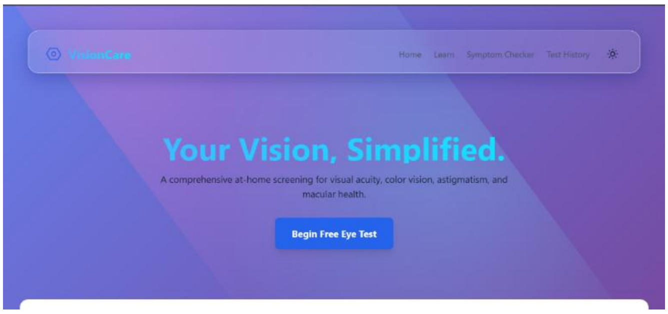

Fig. 2. Home Page of Website

- **Disease Prediction Module**

Users could easily upload eye images, and prediction results would be instantly generated. These predictions were shown with proper disease name and confidence level.

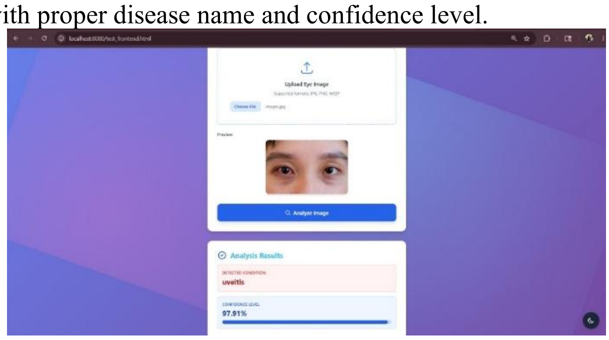

Fig. 3. Checking of eye by uploading the photos of eye

- **AI Chatbot Module**

The AI Chatbot module efficiently answered health care questions of users and assisted them in eye care tips.

- **Doctor Finder Module**

It successfully showed ophthalmologists, hospitals, and clinics nearby the user's geographical location using open source map integration.

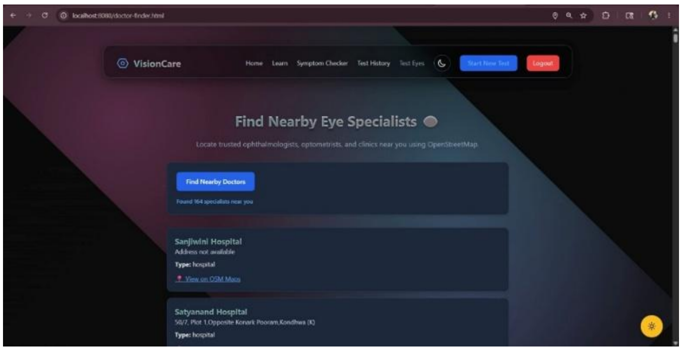

Fig. 4. Doctor Finder Module

- **Amsler Grid Testing**

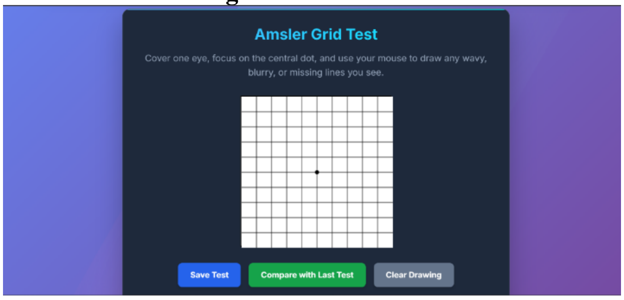

Fig.5. Amsler Grid test interface for early detection of macular degeneration and central vision abnormalities.

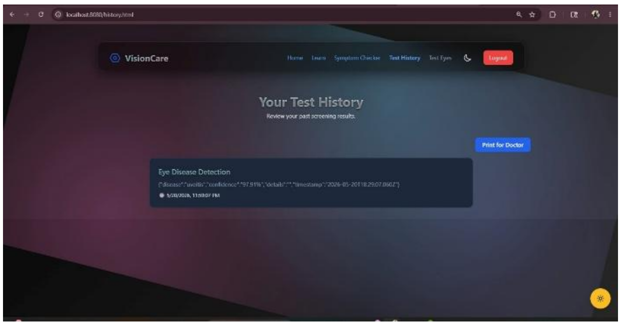

Fig.6. Test History Module of VisionCare System

V. Future Scope and limitations

- **Future Scope**
  1. Integrate more eye-related diseases including glaucoma, diabetes retinopathy, and retinal conditions.
  2. Increase the accuracy in predictions through enhanced deep learning techniques like ResNet and Vision Transformers.
  3. Develop android and iPhone apps to ensure greater accessibility.
  4. Deploy on the cloud to increase flexibility.
  5. Offer multilingual chatbot capabilities to cater to different geographical areas.
  6. Include online booking of appointments along with telemedicine consultations.
  7. Offer an eye disease detection feature based on live video taken via webcam or smartphone cameras.
  8. Connect to EHR systems to provide healthcare data security management.
  9. Incorporate advanced analytics and dashboard monitoring tools for healthcare providers.

- **Limitations**
  1. Currently, the system accommodates a small set of eye diseases.
  2. Predictions depend on the quality of images uploaded under adequate lighting conditions.
  3. The website can only provide preliminary support in healthcare and cannot diagnose patients properly.
  4. Insufficient size of dataset may hamper model accuracy when generalizing over all scenarios in real life.
  5. An internet connection is necessary to facilitate services via a chatbot and API.
  6. Camera quality of different devices may impact the performance of predictions.
  7. OpenStreetMap search for a doctor relies on data available on maps.
  8. At present, there is no mobile application and the system works only on the website.

VI. Conclusion

The successful implementation of VisionCare has demonstrated that Artificial Intelligence, Deep Learning, and web technologies can be effectively integrated into the healthcare sector to diagnose eye problems as well as help patients visualize better. VisionCare was developed as a web-based healthcare application that has the capability to detect eye diseases based on eye image data provided by the patient while at the same time offering other healthcare services like AI chatbot, doctor search, and vision tests.

VisionCare uses a Convolutional Neural Network (CNN) approach that is based on eye image data sets and categorization for disease prediction. VisionCare accurately predicted several diseases like Cataract, Bulging Eyes, Crossed Eyes, Uveitis, and Normal Eyes with high prediction accuracy and confidence scores.

In addition, the project proved to have implemented the effective use of Flask, FastAPI, and Node.js technologies when creating the backend and for REST API communications. The front-end created using HTML, CSS, and JavaScript provided a responsive and easy-to-use interface for interaction with the healthcare system. Furthermore, the AI-powered chatbot module, Nearby Doctor Finder through OpenStreetMap API integration, and the Amsler Grid Test module helped improve access to healthcare and engagement with the application.

Use of SQLite and MongoDB database technologies facilitated effective management and storage of healthcare records, history of predictions, and user data. Through experimental testing, we found that the application was effective in communicating between the front-end and back-end services at relatively fast response times.

In conclusion, the VisionCare platform proves that AI is indeed an effective technology in healthcare software, which can be employed for preliminary diagnosis, healthcare accessibility, and provision of healthcare assistance. The project was successful in accomplishing its goals and objectives.

VII. References

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105, 2012.

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. International Conference on Learning Representations (ICLR), 2015.

[4] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2016. [Online]. Avail-able: https://www.tensorflow.org/

[5] TensorFlow Documentation, “TensorFlow API Documentation,” [Online]. Available: https://www.tensorflow.org/api_docs

[6] Keras Documentation, “Keras Deep Learning API,” [Online]. Available: https://keras.io/

[7] Flask Documentation, “Flask Web Framework,” [Online]. Available: https://flask.palletsprojects.com/

[8] FastAPI Documentation, “FastAPI Framework Documentation,” [Online]. Available: https://fastapi.tiangolo.com/

[9] OpenStreetMap Contributors, “OpenStreetMap,” [Online]. Available: https://www.openstreetmap.org/

[10] MongoDB Documentation, “MongoDB Database Documentation,” [Online]. Available: https://www.mongodbcom/docs/

[11] Nde.js Documentation, “Node.js JavaScript Runtime,” Onlin]. Available: https://nodejs.org/en/docs/

[12] . Kingma, P., & Ba, J. "Adam: A method for stochastic optimization" Proceedings of the International Conference on Learning Representations (ICLR), 2015

[13]R. C. Gonzales and R.E. Woods, Digital Image Processing 4th edition Noida, India: Pearson Education, 2018

[14] S. Russell, & P. Norvig Artificial Intelligence:A Modern Approach, 4th editionPearson Education, 2021

[15] J. Brownlee, Deep Learning for Computer Vision Machine Learning