



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 8, August 2024

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.524

 9940 572 462

 6381 907 438

 [ijirset@gmail.com](mailto:ijirset@gmail.com)

 [www.ijirset.com](http://www.ijirset.com)

# SIGHTSCULPT: AI-Driven Vision Disorder Analysis

Dr.M.Nirmala<sup>1</sup>, Ravi Shankar M V<sup>2</sup>, Ayyanar S<sup>3</sup>

Associate Professor, Department of Master of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, India<sup>1</sup>

II MCA Student, Department of Master of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, India<sup>2</sup>

II MCA Student, Department of Master of Computer Applications, Hindusthan College of Engineering and Technology, Coimbatore, India<sup>3</sup>

**ABSTRACT:** This project focuses on developing a convolutional neural network (CNN) to classify medical images into categories such as normal, diabetic retinopathy, glaucoma, and cataract. Leveraging layers like Conv2D and MaxPooling2D, the CNN extracts and learns features from the input images, followed by dense layers to perform the final classification. The model is trained using a dataset containing labelled images for each condition. During the training process, metrics such as training and validation accuracy are tracked to monitor the model's performance and diagnose potential issues like overfitting or under fitting. The provided code snippet demonstrates how to visualize these metrics using matplotlib, enabling a clear understanding of the model's learning progress over multiple epochs. This visualization is crucial for ensuring the model's robustness and effectiveness in classifying medical images, which can aid in early diagnosis and treatment planning. The project aims to leverage deep learning techniques to improve the accuracy and efficiency of medical image classification, contributing to better healthcare outcomes.

**KEYWORDS:** Vision Disorder, ophthalmic disease, deep learning cataract, glaucoma, diabetic retinopathy

## I. INTRODUCTION

In machine learning, evaluating a model's performance during training is essential for understanding its learning progress and diagnosing issues like overfitting or under fitting. The provided code snippet demonstrates how to plot training and validation accuracy using Matplotlib for visualization and NumPy for generating synthetic data. It sets up the plot with epoch numbers on the x-axis and accuracy values on the y-axis, creating line plots for training and validation accuracy. The plot is titled, labelled, and includes a legend and grid for clarity. To apply this code with real data, replace the synthetic accuracy arrays with actual values from your model's training history, offering an accurate visual representation of your model's performance over time. In this project, we use convolutional neural networks (CNNs) for image classification tasks, leveraging layers such as Conv2D and MaxPooling2D to learn and extract features from images, followed by dense layers for classification.

## II. RELATED WORK

Diverse methods have been proposed for the categorization of eye conditions. Convolutional neural networks (CNN) are used by the authors in [1] to propose a two-stage method for location of optical discs (OD) on fundus images. According to a different study, models for automatically classifying ocular illnesses were developed [2] and rely on the distillation of knowledge. This system incorporates two deep networks, each of which is trained and optimized separately. The deep network ReLayNet is fully convolutional was suggested by Lee et al. [3] to segregate retinal fluids and layers from OCT data.

Researchers [4] developing a technique for the diagnosis of various eye conditions utilizing optical coherence tomography. Employing dilated convolution filters in convolutional neural networks, OCT scans were classified pixel-by-pixel [5]. The performance of the model was assessed using 400 OCT images of individuals with different stages of age-related macular degeneration. Hu et al. [6] proposed a CNN-based method for intraretinal fluid

detection on OCT images. Twelve thousand OCT scans were used to train the CNN model, and following cross-validation, the segmented pictures had a result on the dice of 0.911.

The authors [7] used a special multifunctional architecture with convolution through supervised instruction method. Three tasks possess training into this model simultaneously: lesion identification, segmentation of red lesions, and brilliant lesion segmentation. It did well, as seen by its 0.839 AUC. A neural network with convolutions (CRFs) and fully linked random fields with conditions are the foundation of a proposed retinal vascular segmentation method [8]. The STARE [8] and DRIVE [9] datasets' color fundus images were used to analyse the model's precision and effectiveness.

### III. METHODOLOGY

#### Proposed System

The Convolutional Neural Networks (CNNs) are commonly used in eye image classification tasks due to their ability to effectively capture spatial hierarchies and patterns in images. Here's how CNNs are typically used in eye image classification:

**Convolutional Layers:** The initial layers of a CNN consist of convolutional filters that learn features directly from the input images. In the context of eye image classification, these filters detect basic patterns such as edges, textures, and shapes relevant to different eye conditions.

**Pooling Layers:** After convolutional layers, pooling layers are often used to downsample the feature maps, reducing the spatial dimensions while retaining the most important features. Max pooling is a common pooling technique used in CNNs, which selects the maximum value from each region of the feature map.

**Flattening and Fully Connected Layers:** Following the convolutional and pooling layers, the feature maps are flattened into a one-dimensional vector and passed through fully connected layers. These layers learn higher-level features by combining the features learned in the earlier layers. In eye image classification, these layers integrate information from different parts of the image to make predictions about the presence of specific eye conditions.

**Output Layer:** The final layer of the CNN is the output layer, which typically uses a softmax activation function to produce probability scores for each class (e.g., normal, glaucoma, diabetic retinopathy, cataract). The class with the highest probability is predicted as the classification result for the input image.

**Training and Optimization:** CNNs are trained using labeled training data through a process called backpropagation, where the network learns to minimize a loss function by adjusting its weights and biases. Common optimization algorithms such as stochastic gradient descent (SGD) or Adam are used to update the network parameters during training.

**Evaluation:** Once trained, the CNN is evaluated on a separate set of validation or test images to assess its performance in classifying eye conditions. Metrics such as accuracy, precision, recall, and F1-score are commonly used to evaluate the model's performance.

Overall, CNNs have demonstrated success in eye image classification tasks by automatically learning relevant features from the input images and accurately predicting the presence of different eye conditions based on these learned features.

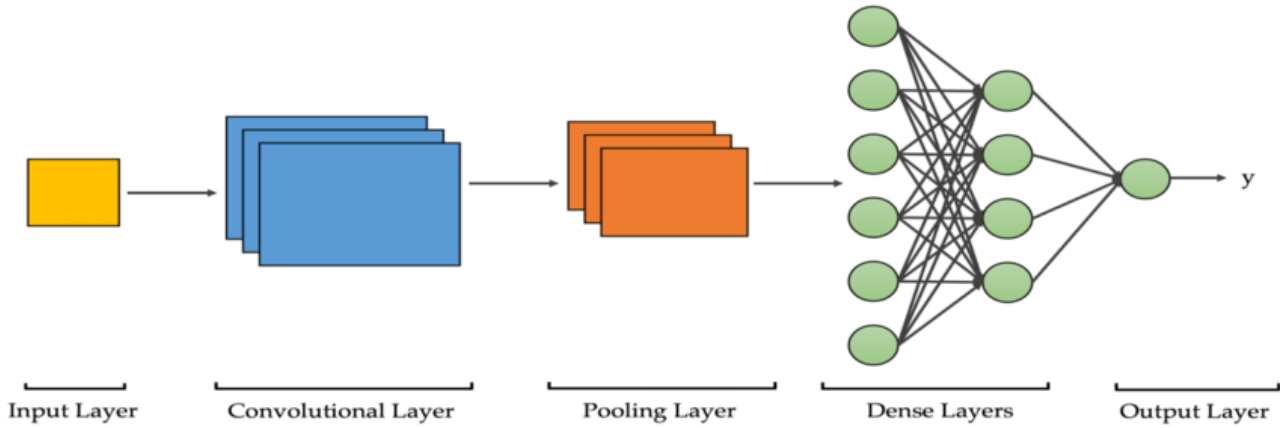


Figure 1: An example of the suggested procedure.

The steps that were done during the research are shown in Figure 1. It illustrates how the previously trained VGG-19 model was used to train each of those images of the left and right eyes independently. Following training, they were classified as either the first class (disease) or the second class (normal).

**Dataset**

The Kaggle dataset titled "Eye Diseases Classification" is designed to help classify various eye diseases using image data. This dataset includes images for conditions such as glaucoma, diabetic retinopathy, and cataracts. Here's a detailed overview:

To download this dataset use: <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification>

Table 1 Displaying the total number of images

Total number of Normal Images	1074
Total number of Diabetic Retinopathy Images	1098
Total number of Glaucoma Images	1007
Total number of Cataract Images	1038
Total images	4217

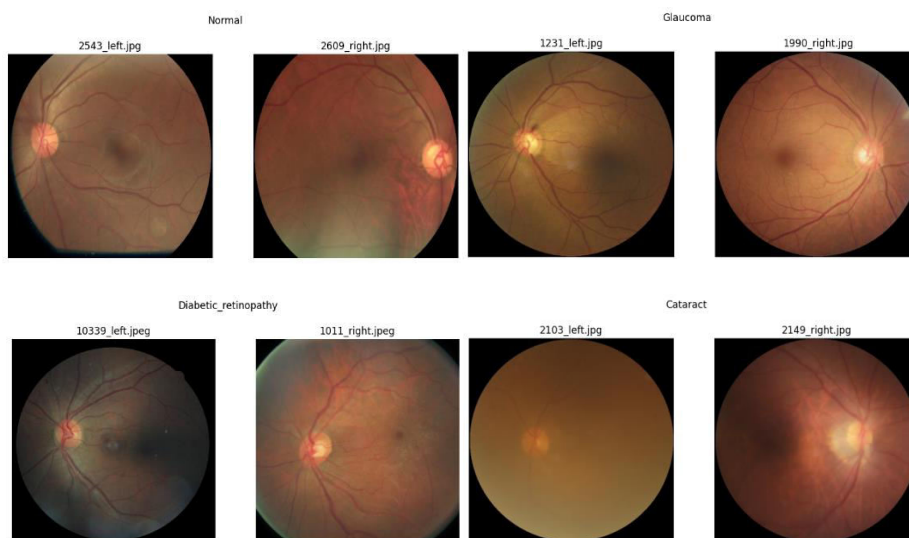


Figure 2: Sample Images from the Dataset

### 3.1 Convolution Neural Network

In applications such as object identification and picture classification, the neural network with convolution performs remarkably well. One machine-learning technique for creating multilayer perceptron's intended to in-order to process two-dimensional input is convolutional neural networks. CNNs are one sort in terms of deep neural networks since they can combine picture data in a network in numerous ways. This is what defines CNNs as deep neural networks. Convolution serves as the foundation for the CNN algorithm. Activation layers, pooling layers, flattening layers, and operations involving convolution are the layers that make up the CNN algorithm.

$$\begin{aligned} \text{Convolution Operation, } C(i, j) &= (I * K)(i, j) \\ &= \sum_m \sum_n I(m, n) K(i - m, j - n), \\ \text{Activation Layer, Sigmoid } S(x) &= \frac{1}{1 + e^{-x}}. \end{aligned}$$

### 3.2 Research Steps

#### Processing of images in detail

For each image, it reads the image using OpenCV, converts the color format from BGR to RGB, and resizes the image to a specified dimension. The processed image data is then appended to a list X, while the corresponding label based on a predefined dictionary is appended to another list y. This process prepares the image data (X) and labels (y) for training a machine learning model to classify eye conditions based on retinal images.

first normalizes the pixel values of the input images in the dataset X by dividing them by 255, effectively scaling the pixel values to the range [0, 1]. Then, it splits the dataset into training and testing sets using the train\_test\_split function from scikit-learn, where 20% of the data is allocated for testing (test\_size=0.2). The pixel values of the training set (X\_train) are converted to the float32 data type, which is a common practice for compatibility with certain machine learning libraries and operations. Similarly, the target labels (y\_train) are also converted to the float32 data type. These preprocessing steps prepare the data for training a machine learning model, ensuring that the input features and target labels are in the appropriate format for the training process.

#### Keras Sequential

Keras Sequential model named data\_argumentation for data augmentation, a technique commonly used to increase the diversity of training data by applying various transformations. The model comprises several layers from the keras.layers.experimental.preprocessing module, including RandomRotation, RandomContrast, and two instances of RandomZoom. Each layer represents a different data augmentation operation: RandomRotation randomly rotates input images by a maximum angle of 0.2 radians, RandomContrast randomly adjusts the contrast of images by a factor of 0.3, and RandomZoom randomly zooms into images by factors of 0.3 and 0.7, respectively. These augmentation layers help improve the model's robustness and generalization ability by introducing variations in the training data.

### 3.3 Model

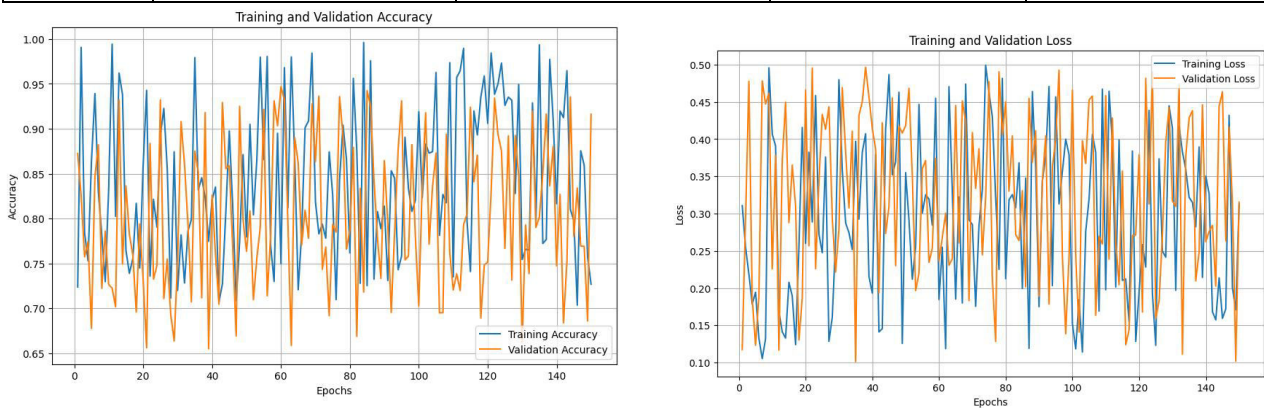
A convolutional neural network (CNN) model using TensorFlow's Keras API. The model comprises several layers organized sequentially within a Sequential model container. The first layer is the data\_argumentation model, which performs data augmentation operations on input images to increase the diversity of the training dataset. Subsequent layers consist of convolutional (Conv2D) and max-pooling (MaxPooling2D) layers, with varying filter sizes and activation functions such as softmax and relu. The dropout layer (Dropout) is added to prevent overfitting by randomly dropping 20% of the units during training. Finally, the model includes a flatten layer (Flatten) to prepare the feature maps for the fully connected layer (Dense) with 50 units and a sigmoid activation function, which serves as the output layer. The model is compiled with the Adam optimizer, sparse categorical cross-entropy loss function, and accuracy as the evaluation metric. After compilation, the model is saved as model.h5.

**Accuracy and loss of training and validation**

There are only 207 instances of glaucoma in the sample. We included 206 normal instances to counterbalance this and prevent the model from overfitting. We obtained a training loss alongside precision after passing the dataset through the pre-trained VGG-19 following a sample of data.

**Table 3.3.1 Accuracy and loss of training and validation values for first 5 epoch.**

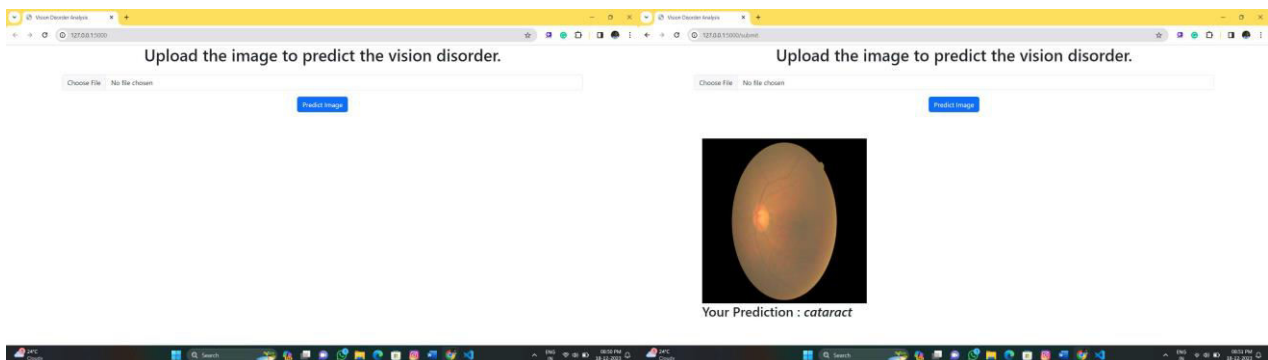
Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	0.842426	0.707203	0.166613	0.322946
2	0.827947	0.664900	0.204244	0.187314
3	0.887549	0.890128	0.188263	0.165655
4	0.880799	0.811233	0.279148	0.438088
5	0.965035	0.704596	0.489818	0.396212



**Figure 3: Models Training and validation of accuracy and loss of each epoch.**

**3.4 Flask Web Application Using this model**

Flask web application for predicting eye diseases from uploaded images. It imports necessary libraries including Flask for web development, Keras for loading the pre-trained CNN model, and PIL for image processing. The predict\_label function preprocesses the uploaded image, makes predictions using the loaded model, and returns the predicted eye disease category. The Flask routes handle HTTP requests: '/' renders the index page, and '/submit' processes the uploaded image, saves it, and returns the prediction result to the index page. The application runs locally on debug mode when executed directly. The template file 'index.html' contains HTML code for the user interface, allowing users to upload images and view prediction results.



**Figure 3.4.1 Screenshots of flask application.**

Flow chart diagram

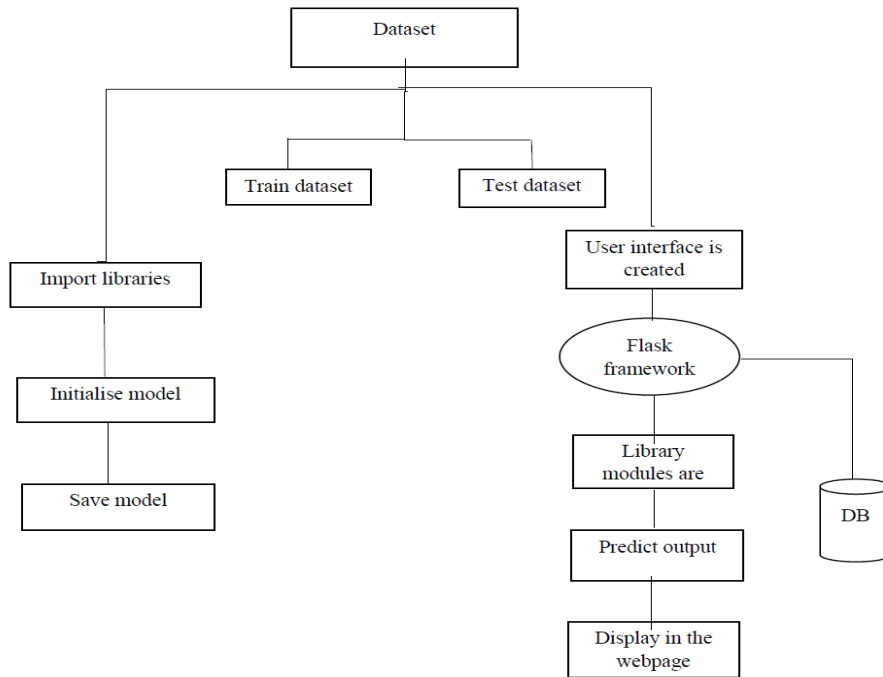


Figure 3.4.2 Flow chart diagram for this project

REFERENCES

1. K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” 2014, <https://arxiv.org/abs/1409.1556>. View at: Google Scholar
2. X. Meng, X. Xi, L. Yang, G. Zhang, Y. Yin, and X. Chen, “Fast and effective optic disk localization based on convolutional neural network,” *Neurocomputing*, vol. 312, pp. 285–295, 2018. View at: Publisher Site | Google Scholar
3. C. S. Lee, A. J. Tying, N. P. Deruyter, Y. Wu, A. Rokem, and A. Y. Lee, “Deep-learning based, automated segmentation of macular edema in optical coherence tomography,” *Biomedical Optics Express*, vol. 8, no. 7, pp. 3440–3448, 2017. View at: Publisher Site | Google Scholar
4. C. Ployout, R. Duval, and F. Chriet, “A Novel Weakly Supervised Multitask Architecture for Retinal Lesions Segmentation on Fundus,” *IEEE Trans Med Imaging*, vol. 38, 2019. View at: Google Scholar
5. H.-C. Shin, H. R. Roth, M. Gao et al., “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016. View at: Publisher Site | Google Scholar
6. K. Hu, Z. Zhang, X. Niu et al., “Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function,” *Neurocomputing*, vol. 309, pp. 179–191, 2018.
7. View at: Publisher Site | Google Scholar
8. A. W. Hoover, “STARE (structured analysis of the retina),” <https://paperswithcode.com/dataset/stare>. View at: Google Scholar
9. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, “Ridge-based vessel segmentation in color images of the retina,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, April 2004. View at: Publisher Site | Google Scholar
10. Z. Li, Y. He, S. Keel, W. Meng, R. T. Chang, and M. He, “Efficacy of a deep learning system for detecting glaucomatous optic neuropathy based on color fundus photographs,” *Ophthalmology*, vol. 125, no. 8, pp. 1199–1206, 2018. View at: Publisher Site | Google Scholar



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details