Advances in Knowledge-Based Systems, Data Science, and Cybersecurity Received 15-06-2026; Accepted 21-08-2026; Published 28-08-2026

Deep Learning Based Detection of Eye Conditions Using External Eye Images

Ammar Almutawa AHALMUTAWWA@uj.edu.sa
Department of Information Systems and Technology
University of Jeddah
Jeddah, Saudi Arabia

Mustafa Kafyyah MKAFYYAH.stu@uj.edu.sa
Department of Information Systems and Technology
University of Jeddah
Jeddah, Saudi Arabia

Hashim Alattas HALATAS.stu@uj.edu.sa
Department of Information Systems and Technology
University of Jeddah
Jeddah, Saudi Arabia

Osama Alghamdi OALGHAMDI0226.stu@uj.edu.sa
Department of Information Systems and Technology
University of Jeddah
Jeddah, Saudi Arabia

Mohammed Mubarak MMUBARAK0002.stu@uj.edu.sa
Department of Information Systems and Technology
University of Jeddah
Jeddah, Saudi Arabia

Ammar Yahya AYAHYA0024.stu@uj.edu.sa
Department of Information Systems and Technology
University of Jeddah
Jeddah, Saudi Arabia

Corresponding Author: Ammar Almutawa

Abstract

Eye conditions such as anemia, cataract, conjunctivitis and uveitis can often be identified from an external inspection. However, many people in remote or low resource areas lack access to proper healthcare facilities. This project proposes a low-cost, non-invasive system that uses external eye images to provide an initial non-diagnostic assessment of the conditions. A smartphone user uploads a close-up image of the eye, the system verifies that an eye is present and preprocesses the image. The processed image is then passed through a deep learning model that estimates the likelihood of disease presence. Data was sourced from publicly available datasets. The prototype application's outputs can show the likelihood of

Citation: Ammar Almutawa, et al. Deep Learning Based Detection of Eye Conditions Using External Eye Images. **Advances in Knowledge-Based Systems, Data Science, and Cybersecurity** ;Research. 2026;3(2):29.

Page No.- 579-599.

the disease and a simple next-step guide to help the user. This would not replace doctors, but it could provide early guidance for people living far away from healthcare centers and make basic screening more accessible.

Keywords: Machine Learning, Deep Learning, CNN, Eye Diseases, Anemia, Conjunctivitis, Uveitis, Cataract

1. INTRODUCTION

1.1 Background

The eye is one of the few places in the body where serious conditions leave visible signs that can be spotted without specialized equipment. Anemia, conjunctivitis, uveitis and cataract all produce visible changes in the conjunctiva, lens, or surrounding tissues, making them strong candidates for low-cost, image-based screening [1]. The challenge is that acting on those signs still requires a trained specialist and access to one is exactly what remote parts of the world lack [2]. This project aims to close that gap by building a system that takes a single smartphone photograph of the eye and provides an initial assessment of whether any of the four conditions may be present. These diseases were chosen because each produces externally visible signs, each benefits from early detection and together they represent a range of common conditions that occur worldwide.

Anemia occurs when the body lacks sufficient healthy red blood cells to carry oxygen. Common symptoms include fatigue, weakness and feeling short of breath. Anemia affects about a third of the world's population [3].

Conjunctivitis causes inflammation in the conjunctiva, the thin membrane that covers the white of the eye and lines the inside of the eyelids. Infection may spread quickly, but treatment is easy. The disease affects individuals of all ages [4].

Uveitis is an inflammatory condition affecting the uvea, the middle layer of the eye. The inflammation may involve a single region of the uvea or multiple regions and it can affect one eye or both simultaneously. Depending on the location of the inflammation, uveitis may cause pain, redness, irritation, blurred vision, or general vision disruption. The condition may develop suddenly and worsen quickly, or it may progress gradually over time [5].

Cataract is a clouding of the eye's clear lens. Most cataracts develop slowly and do not disturb vision at first, but over time they may cause symptoms such as blurred or dim vision, trouble seeing at night and sensitivity to light and glare. If left untreated, cataracts can severely impair vision, especially in older adults [6].

1.2 Problem Definition

Many people living in the countryside or semi-remote areas do not have access to well-equipped hospitals or health care centers because large hospitals are usually concentrated in cities, far from these areas. Even if there is one, it is usually not up to standard compared to those in big cities. Late

diagnosis for some of these diseases could lead to complications such as permanent loss of sight and make recovery harder and longer [7].

Patients would view a simple appointment with a doctor to check their symptoms as a waste of resources (time, transportation, money), so check-ups are often neglected. These could all be avoided by early examinations. However, diagnostic tests can be costly, physically straining and require specialized equipment and trained staff, which are not accessible in places with fewer resources [8].

However, there are signs in the eye that may indicate the possibility of having one of these diseases. By developing an image-processing algorithm, users could perform an initial self-examination at home using only a smartphone camera. This would not replace a doctor, but it could provide early guidance for people living far away from healthcare centers and make basic screening more accessible.

1.3 Recommended Solution

The suggested solution is to create a low-cost, non-invasive tool that can analyze the eye using a smartphone image. A machine learning model evaluates a patient's case by analyzing an eye image without the need for a medical professional. The system does not require specialized equipment or highly trained staff, only a smartphone camera is needed. The app will provide a likelihood of the detected disease and give guidance on the next appropriate steps.

It is important to emphasize that the system's primary goal is to help users decide whether a visit to a clinic or a specialist is warranted. It is a helpful screening aid and will not replace the need for a doctor's diagnosis or clinical judgment.

2. BACKGROUND AND LITERATURE REVIEW

2.1 Related Work

Technology is a key part of healthcare. It is used every day to help doctors diagnose and treat patients accurately, whether through medical imaging systems, robotics or electronic health records [9]. In recent years, machine learning has become even more important because it can find patterns in data that are difficult for humans to detect. In eye medicine, image-based tools are already available to screen for diseases. External photos of the eye can show useful signs, such as redness, cloudiness or pallor, that can indicate conditions like anemia, conjunctivitis, cataract or uveitis [1]. This study builds on these ideas by using deep learning on external eye photographs to provide an initial, non-diagnostic assessment that can guide users without access to health centers in low-resource areas.

Several studies have explored the use of machine learning in the medical field, specifically for eye diseases. In this section, we reviewed related studies that propose and develop various machine learning and deep learning techniques to assist both healthcare professionals and patients. All of the following works focus on models that use image data and/or biodata to support the diagnosis of eye diseases or related conditions.

2.1.1 Detection of anemia using conjunctiva images: A smartphone application approach

Appiahene et al. (2023) [10], developed a smartphone application that detects anemia from conjunctival pallor (pale conjunctiva, lacking redness) using machine learning models, including convolutional neural networks and logistic regression. The dataset they used was from a publicly available set containing 710 images of the conjunctiva from patients in Ghana. They also created a data collection system using the "Kobo Collect" application, where users were able to volunteer their biodata, such as age, gender, Hb value and a photo of their eye and conjunctiva.

The study is relevant to our work in many ways. Both systems use external images of the eye to train and test machine learning algorithms. Another similarity is that both aim to build an automated, non-invasive tool accessible in low-resource settings. Appiahene et al. (2023) [10], focus only on the detection of anemia, while ours aims to detect other conditions, such as uveitis, conjunctivitis and cataract. Another difference is that their system uses detailed patient biodata, while our project uses only the eye image and provides guidance based on visual information.

2.1.2 Application of machine learning approach for iron deficiency anaemia detection in children using conjunctiva images

Asare et al. (2024) [11], developed multiple machine learning frameworks for detecting iron-deficiency anemia. Their work uses approaches such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Naïve Bayes and Decision Trees. To acquire their data, they used the "Kobo Collect" application. Users volunteered their biodata, such as age, gender, Hb value and a photo of their conjunctiva. The dataset for the study was aimed at children under 6 years old, that is, from 6 to 59 months.

This study is similar and relevant to our work because it involves machine learning based anemia detection using images of the eye and it emphasizes the importance of cheaper, more accessible screening tools that do not require professional personnel or expensive equipment. The differences are that Asare et al. (2024) [11], explore many machine learning approaches using various data types, including conjunctiva images. It also only focuses on the detection of iron-deficiency anemia, while ours aims to detect other conditions, such as uveitis, conjunctivitis and cataract.

2.1.3 Deep learning model-based detection of anemia from conjunctiva images

Sehar et al. (2025) [12], proposed a non-invasive technique for detecting iron-deficiency anemia from conjunctiva images. Their work uses machine learning approaches such as Support Vector Machines (SVMs), K-Nearest Neighbors (KNNs), Naïve Bayes and Decision Trees. To acquire their data, they used online databases and photos they captured with smartphones.

Just like our study, Sehar et al.(2025) [12], use machine learning algorithms to detect anemia using external images of the eye and conjunctiva. Moreover, they aim to develop cheaper, more accessible screening tools that do not require professional personnel or expensive equipment. The main differences are that their study focuses purely on anemia and evaluates many alternative

models, while our project extends it to several conditions, including uveitis, conjunctivitis and cataract and their work does not include an app.

2.1.4 Classification and localisation of diabetic-related eye disease

Osareh et al. (2002) [13], focused on the detection of diabetic-related eye diseases using fundus images, photographs of the inside of the eye. They used image processing and machine learning techniques to segment the retina, identify lesions and classify whether the eye showed signs of diabetic retinopathy. Techniques used include K-Nearest Neighbor (KNN), Linear Delta Rule (LDR) and Neural Networks (NN).

This work is relevant to us because it demonstrates how ML models can classify eye diseases using image data. Osareh et al. (2002) [13], and our studies both aim to create algorithms to detect eye diseases. However, this study focuses on the classification of eye disease related to diabetes, which differs from our slightly broader selection of eye conditions. They also used specialized equipment to capture an image of the retina, while ours uses everyday smartphones to capture an image of the outside of the eye.

2.1.5 A deep learning model for novel systemic biomarkers in photographs of the external eye: A retrospective study

Babenko et al. (2023) [14], developed a deep learning system (DLS), specifically trained on data using a CNN, that takes external eye images and predicts a range of biomarkers, indicators of a condition or disease. Their dataset included 215,665 images from 38,398 patients with diabetes.

Both this and our study are based on using external photographs of eyes to detect diseases and conditions using deep learning models, CNNs, to learn from the images and return probabilistic outputs. However, this study focuses on biomarkers for detecting various diseases, not specifically eye conditions. Also, their images include fundus images in addition to external images, whereas our system is limited to external images captured with smartphones.

2.1.6 Data driven approach for eye disease classification with machine learning background knowledge review

Malik et al. (2019) [15], proposed a general framework for recording diagnostic data in an international format and using it to classify eye disease with machine learning. They used algorithms including Decision Trees, Naïve Bayes, Random Forests and Neural Networks to analyze patient data. The study focuses on structured data such as age, symptoms and illness history rather than images.

Both studies develop machine learning algorithms for diagnosing eye-related conditions. However, Malik et al. (2019) [15], do not use images; their inputs are symptoms and general health data, while our project relies on photographs of the eye. Their proposed framework is clinic-centric, meaning it

is designed for trained staff, whereas our project is intended for direct patient use without requiring medical training.

2.1.7 AI workflow, external validation and development in eye disease diagnosis

In this study, Chen et al. (2025) [16], evaluated how AI tools can be used in a clinical workflow for eye disease diagnosis, specifically age-related macular degeneration (AMD). They used approximately 60,000 images from approximately 4500 individuals, conducted tests with 24 clinicians and evaluated their diagnostic assessments with and without AI assistance.

Both this and our work use machine learning models to detect eye disease from image data. However, Chen et al. (2025) [16], focus on fundus images, which are inside the eye, unlike our images, which are external photographs of the patient's eye. The environment is also different; their model is meant to assist clinicians with their diagnostic assessment, while ours is meant to be used by the patient for guidance.

2.1.8 Comparisons of deep learning algorithms for diagnosing bacterial keratitis via external eye photographs

Kuo et al. (2021) [17], evaluated several deep learning architectures, such as ResNet50, ResNeXt50 and DenseNet121, for diagnosing bacterial keratitis (BK) from external eye photographs of patients. The authors collected eye photos from patients admitted to Chang Gung Memorial Hospital in Taiwan and trained models to distinguish BK from other types of keratitis.

Like our study, Kuo et al. (2021) [17], used external eye photographs of patients to diagnose eye diseases and they also used deep learning models, such as neural networks. Unlike our study, which focuses on multiple eye conditions, this study focuses on patients with BK and differentiates them from other types of keratitis. Also, their images are obtained from clinical environments using specialized imaging equipment. This project is relevant to us because it shows that deep learning applied to external eye images can help detect and diagnose eye diseases.

2.1.9 Multiple ocular disease diagnosis using fundus images based on multi-label deep learning classification

Ouda et al. (2022) [18], designed a computer-aided diagnosis (CAD) system that automatically detects ocular diseases. They created a multi-label deep learning system, based on CNNs, to diagnose diseases from fundus images. Their dataset contains images with 45 different conditions.

This project shares the idea of multi-disease classification using images and implements deep learning models, such as CNNs, to classify diseases. However, their models focus on fundus images, whereas ours use external images of the eye. They also focused on a wide range of diseases, whereas we focused on only four.

2.1.10 Deep learning system for the auxiliary diagnosis of thyroid eye disease: Evaluation of ocular inflammation, eyelid retraction, and eye movement disorder

Han et al. (2025) [19], proposed a deep learning system to aid in diagnosing thyroid eye disease from photographs. They also focused on symptoms involving eyelid retraction, eye movement disorders and ocular inflammation. The data included 1,839 eye images, which were used to train and evaluate the models.

Both projects focus on using external eye images and deep learning techniques to diagnose eye diseases. This project focuses on only thyroid eye disease, unlike ours, which focuses on four conditions, including uveitis, conjunctivitis, cataract and anemia and their work does not include an app.

2.2 Comparison of Literature

TABLE 1, presents a comparative study of the related works discussed above, summarizing the algorithms used, dataset characteristics, target diseases, image type and the best reported results.

TABLE 1: Comparative study of related works.

Related Works	Algorithm Used	Image Aug.	Dataset Size	Disease	Image Type	Results (Best Model)
Appiahene et al. [10]	CNN, Logistic Regression	Yes	710 images	Anemia	External	Accuracy: 92.5%; Sensitivity: 90%; Specificity: 95%
Asare et al. [11]	CNN, KNN, Naïve Bayes, Decision Tree, SVM	Yes	527 images	Anemia	External	Accuracy: 98.45%; F1: 97.63%; Precision: 97.64%; Recall: 97.63%
Sehar et al. [12]	KNN, Naïve Bayes, Decision Tree, SVM	Yes	764 images	Anemia	External	Accuracy: 89.48%; F1: 91%; Precision: 88%; Recall: 95%
Osareh et al. [13]	SCG*, BP*, LDR*, KNN, QG*	No	60 images resulted in 4037 segmented regions	Diabetic-Related (DR**)	Fundus	Accuracy: 90.1%; Sensitivity: 93.4%; Specificity: 92.7%
Babenko et al. [14]	CNN	No	215,665 images	Not specific to any disease	External	AUC: 87.7%
Malik et al. [15]	Decision Trees, Naïve Bayes, Random Forest, Neural Networks	No	3025 instances	52 different diagnoses	—	Accuracy: 86.63%; F1 Measure: 86.1%; Precision: 88.9%; Recall: 86.6%
Chen et al. [16]	CNN (DeepSeeNet)	No	~60,000 images	AMD**	Fundus	F1 Score: 81.63%
Kuo et al. [17]	CNN (Various Frameworks)	Yes	3,497 images	Bacterial keratitis, non-bacterial keratitis	External	Accuracy: 71.7%; Sensitivity: 82.4%; Specificity: 54.7%
Ouda et al. [18]	CNN (Various Frameworks)	Yes	3,200 images	45 different diseases	Fundus	Accuracy: 94%; Sensitivity: 80%; Precision: 92%; AUC: 97%
Han et al. [19]	CNN (Various Frameworks)	Yes	1,839 images	Thyroid Eye Disease	External	IoU: 95.0%; Dice: 96.9%

*Scaled Conjugate Gradient (SCG), Back-Propagation (BP), Linear Delta Rule (LDR), Quadratic Gaussian (QG).
**Diabetic Retinopathy (DR), Age-related Macular Degeneration (AMD).

2.3 Research Gap

In summary, the literature review showed that existing systems typically focus on a single disease or on internal image types, such as fundus photographs and often require medical professionals, clinical equipment and/or detailed patient data. Unlike this project, which aims to extend these ideas to a multi-condition setting using only external eye images captured with a smartphone and is targeted at users in low-resource settings. The literature review identifies three gaps. No existing smartphone-based system screens for all four target conditions (anemia, conjunctivitis, cataract and uveitis). Prior multi-disease systems rely on fundus images, which require specialized equipment, rather than external photographs. Patient-facing tools that return non-diagnostic guidance without requiring clinical input are lacking. This project develops a multi-condition classifier that operates on external eye images from an ordinary smartphone camera, for use by individuals in low-resource settings.

3. METHODOLOGY

3.1 System Workflow

Proper organization and understanding of the datasets are essential, we achieve this by following several steps; the first of which is data collection. The goal is to ensure that all collected images adequately represent the target groups and are diverse enough to achieve effective learning.

Exploratory data analysis is then performed using descriptive statistics and visualizations to identify patterns, trends and outliers in the data. This analysis also reveals four dataset-level problems that must be addressed before training: class imbalance, irrelevant images, inconsistent image dimensions and irregular file naming. The preprocessing steps in the following section are designed to resolve each of these in turn.

In data cleaning, the irrelevant images were removed for consistency. Image standardization was performed to make the entire dataset uniform. Pixel normalization ensures numerical stability throughout the training. Image augmentation was used to enlarge the sample size for the classes. Data splitting: the data was split into training, validation and test sets. All three partitions contained images from all five classes. By solving these problems, we can ensure the dataset is ready for model training.

For data modeling, we used convolutional neural networks (CNNs). CNNs are a type of deep learning model that works well with images. CNNs have the ability to learn important visual features and distinguish normal eyes from diseased eyes, without manual intervention.

3.2 Data Collection

The data was collected from publicly available repositories. For the eye dataset:

1. Mendeley “Image Dataset on Eye Diseases Classification” [20],

2. Mendeley "CP-AnemiC" conjunctival pallor dataset from Ghana [21],
3. IEEE DataPort "EYES-DEFY-ANEMIA" dataset [22],
4. Kaggle "Eye Disease Dataset" [23],
5. Media Research Lab "MRL Eye Dataset" [24].

For the non-eye dataset, used to train the eye-validation classifier:

6. Animal faces from the dataset released by Choi et al. as part of their StarGAN v2 study [25],
7. Kaggle "Human Faces" [26],
8. Kaggle "Body Parts" [27],
9. Kaggle "Nature" [28],
10. Kaggle "House Rooms" [29].
11. GitHub "AdversarialEyeImages" [30].

These sources were chosen because they are openly licensed, contain authentic images and are organized by class, which makes them suitable for supervised classification.

3.3 Data Description

TABLE 2, presents the data dictionary of the eye disease dataset, listing each class, the number of images and the visible disease attributes. Representative examples of each class are shown in FIGURE 1 through 5.

TABLE 2: Data dictionary of the eye disease dataset.

Disease	No. of images	Disease Attributes
Anemia	218	Pale conjunctiva.
Conjunctivitis	357	Redness, itching, tearing, discharge and crusting of the eyelids.
Cataract	603	Cloudy or blurred vision, difficulty seeing at night, sensitivity to glare.
Uveitis	215	Eye redness, pain, blurred vision, sensitivity to light and floating spots.
Normal	1523	No abnormalities, clear vision, no redness or swelling.

In addition to the five eye classes shown in TABLE 2, a non-eye class of 1,700 images was compiled from multiple publicly available sources mentioned previously. These images include unrelated content such as body parts, scenery, house rooms, human and animal faces, which are used only by the eye-validation stage of the pipeline.

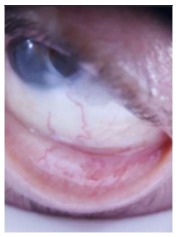

FIGURE 1: Eye with anemia.

FIGURE 2: Eye with conjunctivitis.

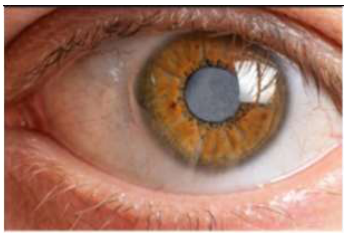

FIGURE 3: Eye with cataract.

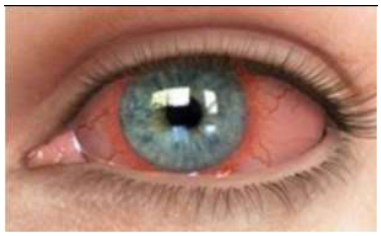

FIGURE 4: Eye with uveitis.

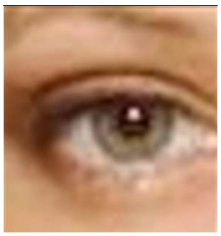

FIGURE 5: Normal eye.

3.4 Data Preprocessing

3.4.1 Challenges with datasets

During the dataset preparation, several challenges were encountered:

Class Imbalance. After counting the number of images in each class, we noticed a significant imbalance between them: some classes are large, some are medium and some are small. This imbalance may lead to problems, such as bias towards larger classes and lower accuracy in smaller classes.

Irrelevant Images. Some of the data resources included irrelevant images, such as drawings, images of animals, images with text and artificially generated images.

Image Dimensions. Because we used different data sources, we encountered inconsistencies in image dimensions; some images had larger widths than others.

File Naming. Because the data sources used different naming conventions, we employed a script to rename all images in each folder to a unified style.

3.4.2 Data cleaning

The data preparation process involved several key steps to ensure the dataset was ready for model training:

Image Cleaning. The irrelevant images were removed for consistency. As previously stated, some of the images consisted of drawings, images of animals, text in images and artificially generated images.

Image Standardization. The images are resized to 224×224 pixels to make the dataset uniform. Since the images were taken from different places, the dimensions vary. This step is important because it reduces memory consumption and speeds up training. Images are renamed to Anemic_001, Anemic_002, Conjunctivitis_001, Conjunctivitis_002, etc., to simplify reading and processing.

Pixel Normalization. The values are rescaled from the 0–255 interval to the 0–1 interval. This step is necessary because it ensures numerical stability throughout training, prevents high values from affecting convergence, drastically accelerates convergence and improves the model's behavior across training cycles.

Image Augmentation and Oversampling. Due to data imbalance, some classes had fewer images than others; data augmentation was used to increase the sample sizes for these classes. The difference between anemia and other classes, such as conjunctivitis, was very prominent. To solve this problem, several methods were employed, including horizontal flip, rotation and zooming. Oversampling is another strategy for addressing image imbalance. It works by increasing the number of samples of the minority classes, making them equal to the number of samples from the larger classes.

Data Splitting. The data was split into training, validation and test sets. All three partitions contained images from all five classes. To ensure a fully held-out evaluation of the two-stage pipeline, both stages share a single unified split. Each eye image is assigned once to the training, validation, or test partition, and the binary eye-validation gate reuses that exact assignment for the eye images, splitting only the non-eye images separately. No eye image is used to train one stage and test the other, removing the leakage risk of independently drawn splits.

3.5 Model Architecture

This section explains the model development process used in the eye disease detection system. Deep learning models were used to analyze uploaded eye images and provide automated screening results. Several convolutional neural network architectures were tested to compare their accuracy, reliability and stability for multiclass eye disease classification.

The system was designed as a two-stage workflow. In the first stage, the uploaded image is passed through a binary eye-validation classifier that determines whether it is a valid eye image; if it is not, the system stops and returns a message to the user. This binary classifier uses the same EfficientNetB0 architecture selected for the disease classifier, trained on the 2,916 eye images and 1,700 non-eye images. Although presented first in the user-facing pipeline, the binary classifier was developed after the disease classifier, reusing the architecture and settings identified as optimal. The disease classifier was trained and evaluated first across the full experimental sweep and the configuration that produced the best results in those experiments (EfficientNetB0) was then reused to train the binary classifier in a single trial. Repeating the full 48-trial sweep for the binary task was

unnecessary, since the architecture and training-setting search had already been completed and the binary problem is substantially easier than the five-class one. In the second stage, the valid image is passed to the disease classification model, which classifies it into one of five categories: Uveitis, Cataract, Conjunctivitis, Anemia and Normal. The classification results are then used to display non-diagnostic guidance to the user.

3.6 Experimental Setup

The model experiments used the parameters listed in TABLE 3.

TABLE 3: Parameter setup.

Setting	Value
Image size	224×224
Batch size	32
Maximum epochs	15
Dataset splits	70/15/15; 80/10/10; 5-fold cross validation
Augmentation	Four levels of augmentation
Oversampling	Oversampling the minority class

After cleaning, the full dataset contained 2,916 eye images across the five classes, used to train the disease classifier. A separate set of 1,700 non-eye images was used for the binary eye-validation classifier. The binary classifier was trained once using the configuration identified as best in the disease-classifier experiments, so the discussion below refers only to the five-class disease classifier.

Depending on the split technique, the training set ranged from 1,894 to 2,332 images and the validation and testing sets ranged from 292 to 438 images. After oversampling, every training set was balanced so that each class matched the size of the largest class in that specific split.

A single set of training parameters was used that was constant across every trial so that any differences in performance could be attributed to the model architecture, the data split or the augmentation level, rather than to inconsistent training settings. Input images were resized to 224 by 224 pixels, batch size is 32 and each model is trained for a maximum of 15 epochs.

The models were evaluated under three different dataset splitting techniques. The first is a 70/15/15 holdout (Training/Validation/Testing), the second is an 80/10/10 holdout (Training/Validation/Testing) and the third is a stratified five-fold cross-validation.

Three splitting techniques were used because model results can change depending on how the data is split, so we wanted to cover all our bases. The 70/15/15 and 80/10/10 holdouts allowed us to observe how performance shifted as the training and test sizes changed, while five-fold cross-validation took it a step further by rotating each image through the test set, removing the effect of any single split. Testing all three gave us confidence that our results accurately reflect the model's performance. Image augmentation (horizontal flips, rotation, zooming and color change) was tested during experiment runs as one approach to improve model generalization. Four different combinations of augmentation were tested:

- Level 0 (none): None
- Level 1 (A1): Flip, Rotation (5%), Zoom (5%)
- Level 2 (A2): Flip, Rotation (10%), Zoom (10%)
- Level 3 (A3): Flip, Rotation (10%), Zoom (10%), Color change

The models compared in the experiment were EfficientNetB0, MobileNetV2, ResNet18 and ResNet50 because they are popular Convolutional Neural Network (CNN) architectures used in computer vision tasks, particularly for image classification.

Every training set is balanced by oversampling each minority class to match the count of the largest class in that split. Oversampling is applied only to the training split; validation and test splits are not oversampled.

The full experimental design consists of four models, three split techniques and four augmentation combinations, which gives 48 training trials in total. For each trial, we record the model's name, split, augmentation level, number of epochs trained, training time and many evaluation metrics. Overall, the training time for all 48 trials took approximately twelve hours. The selection criteria for the final model are the macro F1 score, the balanced accuracy and the training time. We prioritized macro-F1 and balanced accuracy over plain accuracy because these metrics are not biased by class prevalence, which matters when there are fewer images per class.

4. RESULTS

4.1 Evaluation Metrics

The models were evaluated using two main metrics: macro F1-score and balanced accuracy. Other metrics were also recorded for reference but not used as part of the main selection criterion. Macro F1-score and balanced accuracy provide a fairer summary for this multiclass problem. Macro F1-score is the unweighted average of the F1-score of each class, where F1 combines precision and recall into a single number. Because the average is unweighted, every class contributes equally regardless of how many samples it contains. This is important for our project because the dataset is imbalanced.

Balanced accuracy is the average per-class recall, with each class weighing equally. Recall measures the proportion of true cases that the model correctly identifies. Plain accuracy was not used as the main selection metric because it can be misleading on imbalanced datasets. A model that performs well only in the largest class can still achieve a high overall accuracy while performing poorly on the smaller classes. Macro F1-score and balanced accuracy avoid this problem and provide a fairer summary of model behavior.

4.2 Disease Classifier Results

The results below refer to the five-class classifier. They are divided by each model and the best trial per model has been highlighted in bold.

TABLE 4: Results of EfficientNetB0.

Split	Augmentation	Time (s)	Macro F1	Bal. Acc.
80-10-10	none	359.388	**0.975**	**0.979**
80-10-10	A1_light	432.149	0.968	0.975
80-10-10	A2_moderate	430.843	0.962	0.967
80-10-10	A3_heavy	639.346	0.947	0.953
70-15-15	none	321.823	0.960	0.970
70-15-15	A1_light	312.678	0.957	0.960
70-15-15	A2_moderate	418.801	0.951	0.960
70-15-15	A3_heavy	667.750	0.972	0.978
5-fold-cv	none	1585.899	0.968	0.969
5-fold-cv	A1_light	1537.665	0.962	0.965
5-fold-cv	A2_moderate	1641.052	0.957	0.960
5-fold-cv	A3_heavy	2819.102	0.951	0.954

TABLE 5: Evaluation of MobileNetV2.

Split	Augmentation	Time (s)	Macro F1	Bal. Acc.
80-10-10	none	258.726	0.944	0.949
80-10-10	A1_light	353.075	0.966	0.976
80-10-10	A2_moderate	354.740	0.951	0.954
80-10-10	A3_heavy	663.836	0.932	0.946
70-15-15	none	257.407	**0.973**	**0.976**
70-15-15	A1_light	314.763	0.947	0.960
70-15-15	A2_moderate	314.625	0.941	0.945
70-15-15	A3_heavy	552.279	0.959	0.966
5-fold-cv	none	1227.353	0.954	0.955
5-fold-cv	A1_light	1345.738	0.949	0.953
5-fold-cv	A2_moderate	1474.298	0.952	0.957
5-fold-cv	A3_heavy	2577.369	0.945	0.953

TABLE 6: Evaluation of ResNet18.

Split	Augmentation	Time (s)	Macro F1	Bal. Acc.
80-10-10	none	406.814	0.883	0.885
80-10-10	A1_light	270.301	0.692	0.711
80-10-10	A2_moderate	376.735	0.853	0.885
80-10-10	A3_heavy	676.237	0.824	0.876
70-15-15	none	238.781	0.900	0.910
70-15-15	A1_light	333.493	**0.913**	**0.930**
70-15-15	A2_moderate	334.676	0.885	0.907
70-15-15	A3_heavy	597.204	0.688	0.717
5-fold-cv	none	1365.734	0.841	0.834
5-fold-cv	A1_light	1563.872	0.852	0.871
5-fold-cv	A2_moderate	1562.168	0.867	0.867
5-fold-cv	A3_heavy	2780.617	0.812	0.843

TABLE 7: Evaluation of ResNet50.

Split	Augmentation	Time (s)	Macro F1	Bal. Acc.
80-10-10	none	414.484	0.954	0.952
80-10-10	A1_light	532.634	0.955	0.957
80-10-10	A2_moderate	341.019	0.897	0.918
80-10-10	A3_heavy	824.986	0.944	0.945
70-15-15	none	325.967	0.923	0.939
70-15-15	A1_light	484.997	0.962	0.972
70-15-15	A2_moderate	496.533	0.947	0.955
70-15-15	A3_heavy	482.990	0.936	0.947
5-fold-cv	none	2141.773	**0.965**	**0.966**
5-fold-cv	A1_light	2143.834	0.954	0.958
5-fold-cv	A2_moderate	2177.567	0.956	0.959
5-fold-cv	A3_heavy	2977.480	0.947	0.947

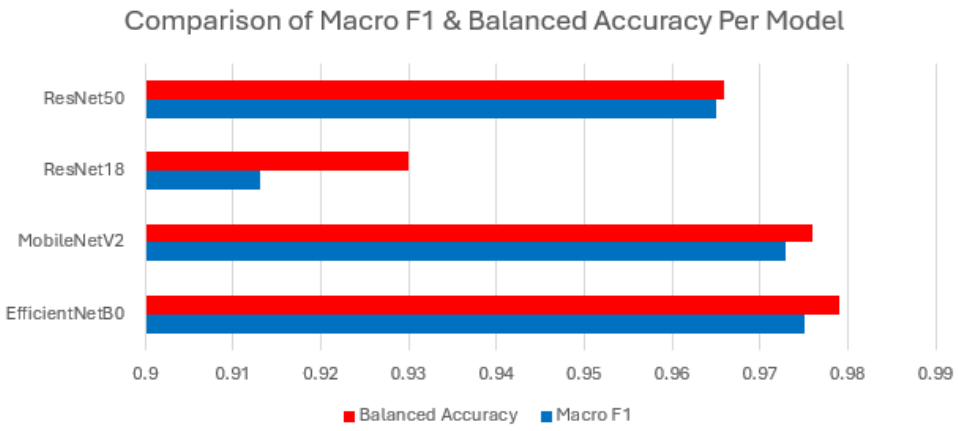

FIGURE 6: Chart comparing metrics per model.

To examine where errors occur, FIGURE 7, shows the confusion matrix of the highest-scoring configuration. Misclassifications are almost entirely between Conjunctivitis and Uveitis, both characterized by ocular redness; the model rarely confuses these with the visually distinct Anemia, Cataract, or Normal classes, consistent with the shared-sign overlap expected for red-eye conditions.

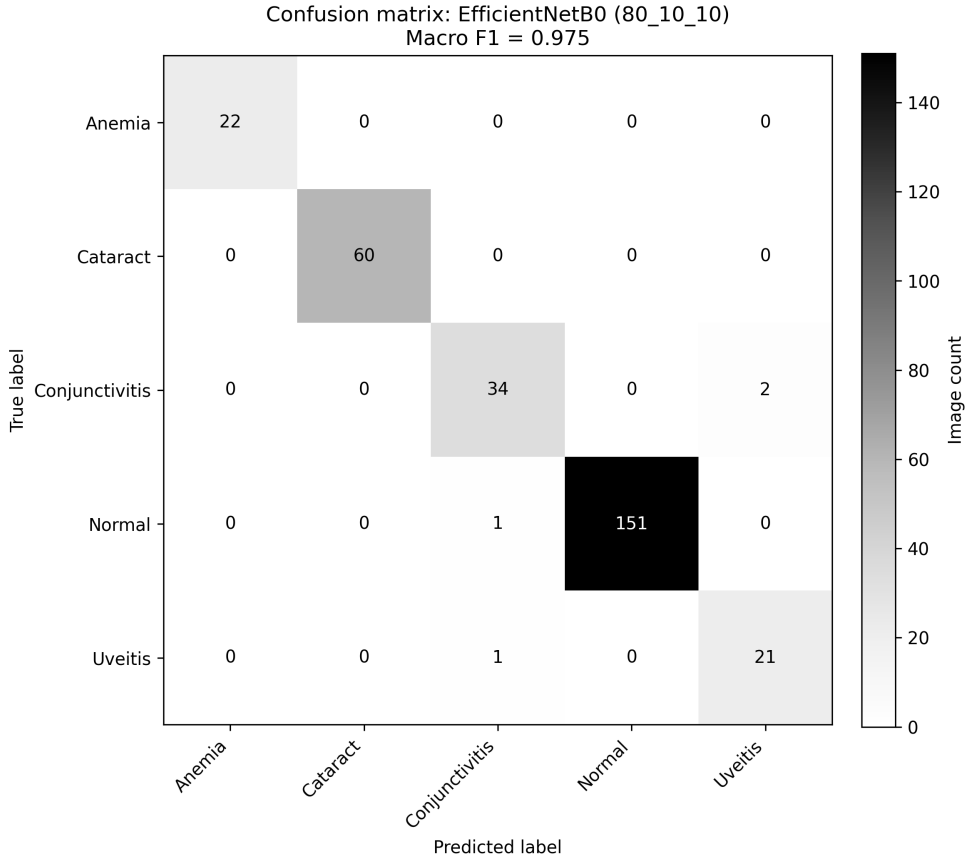

FIGURE 7: Confusion matrix of the highest-scoring configuration (EfficientNetB0, 80/10/10 split); rows are the true class, columns the predicted class.

4.3 Binary Eye-Validation Results

The binary eye-validation classifier was evaluated using the EfficientNetB0 architecture and the highest-scoring configuration identified in the disease-classifier experiments. The results are summarized in TABLE 8.

TABLE 8: Binary eye-validation results.

Metric	Value
Macro F1	1.00
Balanced Accuracy	1.00
Training Time (seconds)	326.1

Training time was approximately 326.1 seconds. The classifier achieved very high performance on the held-out test set, which is consistent with the visual gap between the two classes: eye images form a relatively narrow visual category, while the non-eye images are drawn from highly distinct domains such as scenery, faces and household objects. This score should therefore be interpreted as an indication that the binary task is substantially easier than the five-class one, rather than as a claim of perfect real-world reliability.

4.3.1 Adversarial stress test

We stress-tested the trained gate against images that resemble eyes but are not valid eyes, in three categories; animal eye close ups, drawings of eyes and round object close ups. The gate rejected only 35.8%, confirming it is not robust to eye-like inputs. After hardening the gate by adding half of the adversarial images to the training examples and holding out the other half for evaluation, the rejection of eye-like inputs rose to 91.4% (TABLE 9), with no change to the eye/non-eye result.

TABLE 9: Adversarial stress test.

Baseline Rejection Rate	Hardened Rejection Rate	Delta
35.8%	91.4%	55.6%

5. DISCUSSION

5.1 Model Performance Comparison

Across all 48 trials, EfficientNetB0, MobileNetV2 and ResNet50 performed strongly. As shown in FIGURE 6, the smallest Macro F1 observed was 68.8%, recorded for ResNet18 and the largest Macro F1 score was 97.5%, achieved with EfficientNetB0.

The most important evaluation metric was Macro F1 because it balances precision and recall equally across all five classes. This is suitable for a medical screening system where the model should perform well across all classes rather than only the most common one.

5.2 Effect of Augmentation and Oversampling

One of the most interesting findings of this study is the unexpected behavior of image augmentation. In most deep learning setups, augmentation is considered a standard tool for improving generalization and it is usually expected to raise performance on the validation and test sets. In our experiments, the opposite happened: across nearly every model and every split, the no-augmentation and light-augmentation runs produced higher Macro F1 and higher balanced accuracy than the augmented runs.

There could be many factors to explain this unusual result. One is oversampling. The main reason augmentation is typically helpful is that it increases the model's exposure to a wider range of

examples, especially for minority classes. In our pipeline, class imbalance was already addressed by oversampling each minority class to match the size of the largest class in the training split. Once the training set was balanced this way, additional augmentation no longer added useful information. Another factor could be the data's visual nature. Eye images for these conditions rely on visual cues, such as the redness of the conjunctiva, the cloudiness of the lens or the paleness of the eyelids. Aggressive rotations, zooms and color changes can distort these cues, which can explain why heavier augmentations did not improve performance.

It should be noted that this finding does not mean that augmentation is harmful in general. It only means that, in our setup, oversampling was sufficient to resolve the data imbalance.

Oversampling helped address class imbalance by increasing the representation of minority classes during training. Because oversampling changes the training distribution, balanced accuracy and macro-averaged metrics were especially important. These metrics evaluate each class equally and therefore provide a more trustworthy picture of model behavior than accuracy alone.

5.3 Limitations

Although the project produced good results, there were some limitations:

1. The dataset size was relatively small compared with real medical datasets.
2. Some diseases share similar visual signs, which can lead to confusion during classification.
3. The quality of prediction depends on image clarity, lighting, angle and focus.
4. Oversampling helped during training, but it cannot replace collecting more real images.

6. CONCLUSION

6.1 Summary

The project successfully achieved all its objectives: to develop a deep learning based system for eye disease detection, achieve at least 75% accuracy and develop a prototype app for users. The system works in two stages. First, a binary EfficientNetB0 classifier verifies that the uploaded image is a valid eye image; if not, the user is asked to re-capture the photograph. If the image is accepted, it is passed to a second EfficientNetB0 classifier that assigns it to one of the five target categories. This process can help users get quick initial screening results and increase awareness of possible eye problems.

Several CNN models were trained and tested under the same conditions. The results showed that the top three architectures (EfficientNetB0, MobileNetV2 and ResNet50) performed within roughly one percentage point of one another, with EfficientNetB0 achieving the highest Macro F1-score of 97.5%; ResNet18 was clearly the weakest. The EfficientNetB0 architecture was reused for the binary eye-validation stage, achieving a very high score on the held-out test set. When stress-tested

against eye-like inputs the gate initially rejected only a minority of them, but after hardening with hard-negative examples its rejection rate rose substantially, with no loss on the standard eye/non-eye test. Oversampling improved class balance and led to fairer model performance. In general, the final system achieved strong results and can be considered a useful tool for preliminary screening of eye disease.

6.2 Future Work and Recommendations

Several improvements can be made in the future to improve the system. Increasing the dataset size, especially for minority classes, can improve accuracy and reduce bias. Using real clinical data can help evaluate real-world performance. Training with more diverse images under different conditions can improve robustness. Expanding the scope of conditions the model covers can increase the system's usefulness. Integrating the application with a large language model that has access to the web and medical information can greatly expand the scope of eye conditions detected and provide more personalized guidance to the user. These improvements can address current limitations by improving generalization, reliability, transparency and practical usability.

Based on the results and the limitations observed during this project, we offer the following recommendations for anyone planning to extend or deploy a similar system. First, before adding augmentation to the training pipeline, decide whether class imbalance has already been handled. Our experiments showed that combining oversampling with heavy augmentation actually hurt performance, so these techniques should be tuned together rather than stacked blindly. Second, we recommend treating this kind of system strictly as a screening aid, not a diagnostic tool. The output should always direct the user toward a qualified specialist when a possible condition is detected. Third, when building the dataset for future versions, priority should be given to collecting real clinical images from the target population, especially for minority classes such as anemia and uveitis, since publicly available datasets tend to be small and visually inconsistent. Finally, for real-world deployment, the application should include simple guidance for users on how to capture a usable eye image (good lighting, a steady camera, the correct distance), as image quality was one of the strongest factors affecting prediction reliability.

References

[1] Bell FC. Chapter 114 The external eye examination. Clinical Methods: The History, Physical, and Laboratory Examinations. 3rd ed. Butterworths. 1990.

[2] Resnikoff S, Lansingh VC, Washburn L, Felch W, Gauthier TM, et al. Estimated Number of Ophthalmologists Worldwide (International Council of Ophthalmology Update): Will We Meet the Needs? Br J Ophthalmol. 2020;104:588-592.

[3] https://my.clevelandclinic.org/health/diseases/3929-anemia.

[4] https://www.moh.gov.sa/en/awarenessplateform/VariousTopics/Pages/Conjunctivitis.aspx.

[5] https://www.mayoclinic.org/diseases-conditions/uveitis/symptoms-causes/syc-20378734.

[6] https://www.mayoclinic.org/diseases-conditions/cataracts/symptoms-causes/syc-20353790.

[7] https://www.aao.org/education/current-insight/preventable-blindness-avoiding-cumulative-damage-i.

[8] Truong PN, Merlinsky EA, Maamari NC, Weng CY, Lee AG. Tools and Methods for Cataract Recognition in Low-Resource Settings: A Narrative Review. PLOS Glob Public Health. 2025;5:e0004619.

[9] Thacharodi A, Singh P, Meenatchi R, Tawfeeq Ahmed ZH, Kumar RR, et al. Revolutionizing Healthcare and Medicine: The Impact of Modern Technologies for a Healthier Future—A Comprehensive Review. Health Care Sci. 2024;3:329-349.

[10] Appiahene P, Arthur EJ, Korankye S, Afrifa S, Asare JW, et al. Detection of Anemia Using Conjunctiva Images: A Smartphone Application Approach. Med Nov Technol Devices. Jun. 2023;18:100237.

[11] Asare JW, Brown-Acquaye WL, Ujakpa MM, Freeman E, Appiahene P. Application of Machine Learning Approach for Iron Deficiency Anaemia Detection in Children Using Conjunctiva Images. Inform Med Unlocked. 2024;45:101451.

[12] Sehar N, Krishnamoorthi N, Vinoth Kumar CV. Deep Learning Model-Based Detection of Anemia From Conjunctiva Images. Healthc Inform Res. 2025;31:57-65.

[13] Osareh A, Mirmehdi M, Thomas B, Markham R. Classification and Localisation of Diabetic-Related Eye Disease. In European Conference on Computer Vision. Berlin, Heidelberg. 2002:502-516.

[14] Babenko B, Traynis I, Chen C, Singh P, Uddin A, et al. A Deep Learning Model for Novel Systemic Biomarkers in Photographs of the External Eye: A Retrospective Study. Lancet Digit Health. 2023;5:e257-e264.

[15] Malik S, Kanwal N, Asghar MN, Sadiq MA, Karamat I, et al. Data Driven Approach for Eye Disease Classification With Machine Learning. Appl Sci. 2019;9:2789.

[16] Chen Q, Keenan TD, Agron E, Allot A, Guan E, et al. AI Workflow, External Validation, and Development in Eye Disease Diagnosis. JAMA Netw Open. 2025;8:e2517204.

[17] Kuo MT, Hsu BW, Lin YS, Fang PC, Yu HJ, et al. Comparisons of Deep Learning Algorithms for Diagnosing Bacterial Keratitis via External Eye Photographs. Sci Rep. 2021;11:24227.

[18] Ouda O, AbdelMaksoud E, Abd El-Aziz AA, Elmogy M. MUltiple Ocular Disease Diagnosis Using Fundus Images Based on Multi-Label Deep Learning Classification. Electronics. 2022;11:1966.

[19] Han Y, Xie J, Li X, Xu X, Sun B, et al. Deep Learning System for the Auxiliary Diagnosis of Thyroid Eye Disease: Evaluation of Ocular Inflammation, Eyelid Retraction, and Eye Movement Disorder. Front Cell Dev Biol. 2025;13:1609231.

[20] Bitto AK, Ahmed M. Image Dataset on Eye Diseases Classification (Uveitis, Conjunctivitis, Cataract, Eyelid) With Symptoms and Smote Validation. Mendeley Data. 2024;1. 10.17632/n9zp473wfw.1.

[21] https://data.mendeley.com/datasets/m53vz6b7fx/1. 10.17632/m53vz6b7fx.1.

[22] https://ieee-dataport.org/documents/eyes-defy-anemia.

[23] https://www.kaggle.com/datasets/kondwani/eye-disease-dataset/data.

[24] Sojka E, Gaura J, Fabián T, Fusek R, Krumnikl M, et al. MRL Eye Dataset. Media Research Laboratory. 2020.

[25] https://github.com/clovaai/stargan-v2.

[26] https://www.kaggle.com/datasets/ashwingupta3012/human-faces.

[27] https://www.kaggle.com/datasets/linkanjarad/body-parts-dataset.

[28] https://www.kaggle.com/datasets/heyitsfahd/nature.

[29] https://www.kaggle.com/datasets/annielu21/house-rooms.

[30] https://github.com/MustafaKafyyah/AdversarialEyeImages.