University of Moratuwa
Faculty of Engineering
Department of Electronic & Telecommunication Engineering

BM 2210: Biomedical Device Design

Team Eye-can

Project Report

Group Members

Index No	Name
220009H	Abeysinghe G.A.I.N.M.
220169V	Fernando S.R.N.
220700T	Wickramasinghe S.D.

Contents

1 Introduction

1.1 Need Statement

"Improve the fundoscopic examination equipment so that small movements of the patient do not affect the accuracy of the diagnosis and healthcare professionals can perform this procedure keeping a safe distance with the patient."

1.2 Problem Status

Fundoscopic examination, also known as ophthalmoscopy, is a common procedure in routine eye examinations. This procedure allows doctors to scan the back of the eye (retina), which is crucial for diagnosing many human eye diseases. However, the main problem with this examination is that even the slightest movement of the patient can lead to blurred images, resulting in an inaccurate diagnosis. Additionally, due to the need for close contact during the procedure, healthcare professionals encounter a significant risk.

1.3 Initial Concepts

Smartphone-integrated Indirect Fundoscope

Smartphone-integrated indirect fundoscopy improves retinal examinations by enabling healthcare professionals to capture quality images with smartphone attachments. While it enhances portability and accessibility, patient movement can blur images and affect diagnostic accuracy. It is a valuable tool in areas with limited access to advanced equipment.

Wireless Head-mounted Fundoscope

A wireless head-mounted fundoscope allows for hands-free operation, enabling clinicians to stabilize the patient's head better during examinations, which improves image clarity and reduces the risk of infection. Its wireless design also enhances flexibility, eliminating cable constraints and stationary setups.

Wireless Table-mounted Fundoscope with WiFi Connectivity

The wireless table-mounted fundoscope with WiFi connectivity enhances fundoscopic examinations by stabilizing patient movement for clearer retinal images and fewer diagnostic errors. Real-time image sharing and analysis improve efficiency, and the setup maintains a safer distance, reducing the risk of close-contact infections.

1.4 Final Concept

- Wireless table-mounted Fundoscope with WiFi Connectivity

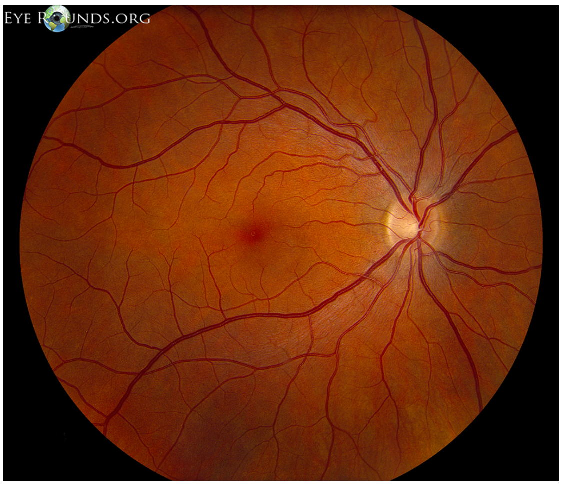


Figure 1: Retina

2 Methodology

2.1 Introduction to fundoscopy

Fundoscopic examination (or ophthalmoscopy) is a procedure used by healthcare professionals to examine the eye's interior surface, particularly the retina, optic disc, macula, and blood vessels. This is done using an ophthalmoscope, which shines a light into the eye to illuminate these structures.

Recognized Diseases Through Fundoscopy

Ophthalmological Conditions

- Diabetic Retinopathy: Retinal hemorrhages, swelling, and microaneurysms.
- Glaucoma: Optic disc cupping and thinning of retinal nerve fibers.
- Hypertensive Retinopathy: Retinal vascular narrowing, hemorrhages, and exudates.
- Retinal Detachment: Retinal elevation or folds.
- Age-Related Macular Degeneration (AMD): Drusen deposits and macular degeneration.

Systemic Illnesses

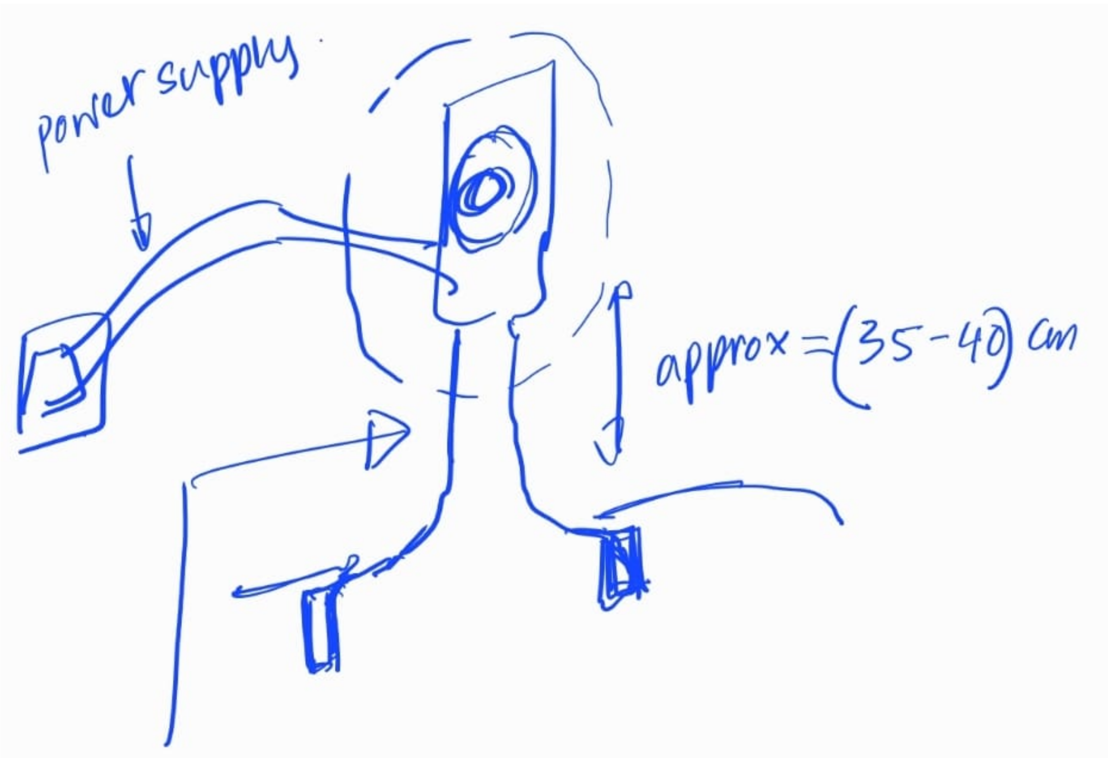
- Papilledema: The optic disc's swelling indicates increased intracranial pressure.
- Retinal Artery Occlusion (RAO): Retinal whitening and "cherry red spot," suggesting cardiovascular issues.

- Autoimmune Diseases: Retinal vasculitis or inflammatory changes (e.g., lupus).
- Infections: Cytomegalovirus (CMV) retinitis with white retinal lesions and hemorrhages.
- Genetic Disorders:
  - Retinitis Pigmentosa: Bone spicule pigmentation and progressive vision loss.
  - Leber's Hereditary Optic Neuropathy (LHON): Optic disc hyperemia and vascular changes.

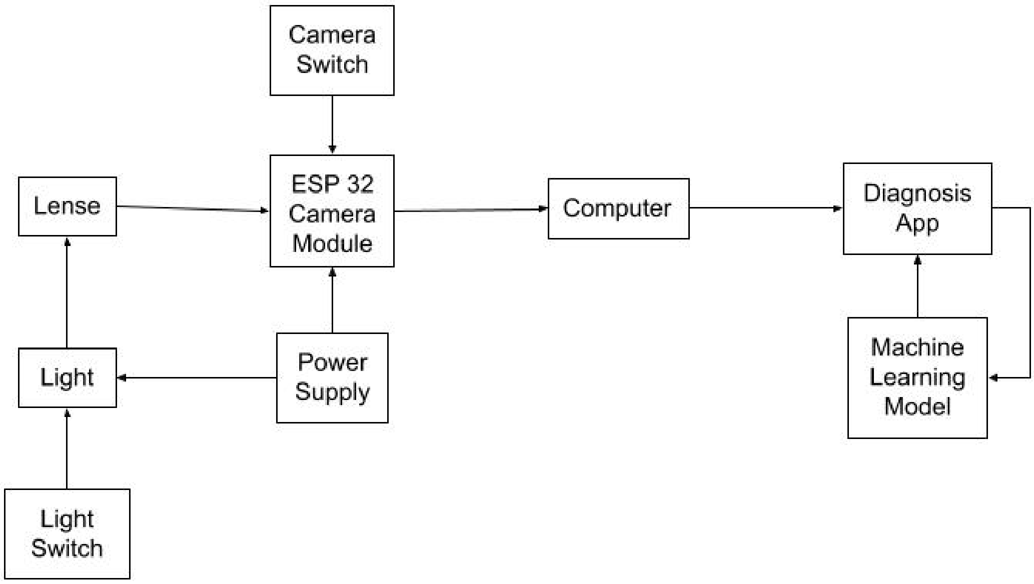
2.2 Our Method

1. A table-mounted fundoscope - No need to move the device, the effect on the image from the movements of the patient and doctor is minimized.
2. WiFi connectivity - Wireless transmission of the retina image to the computer
3. UI integrated Machine Learning system - The doctor can easily upload the image and get a diagnosis of the disease, which can be sent for further analysis later.

2.3 Initial Sketch



2.4 Block Diagram



3 Implementation

3.1 Component Selection

Lens

We tried several lenses for our device.

1. **Macro lens** - Initially, we tried a macro lens used for mobile phones as our lens. Though it is good for taking photos of small objects, it does not have enough power to magnify the retina.

2. **ZnSe laser focus lens** - Our next idea was to use a ZnSe laser focus lens, but it did not suit our purpose either. It is orange in color and we cannot see through properly through it.

3. **Magnifying glass lens** - We then moved on to a magnifying glass lens, which also did not have enough magnification power.

Figure 2: Macro Lens

Figure 3: ZnSe Laser Focus Lens

4. **20D aspheric lens -**

Light

We tried a few lights for our device.

1. **LED ring light** - The ring light we brought online was too big for the device.
2. **Rechargeable 20W light** - This light is also too big for the device.

Figure 4: Magnifying Glass Lens

Figure 5: LED Ring Light

Figure 6: Rechargeable 20W light

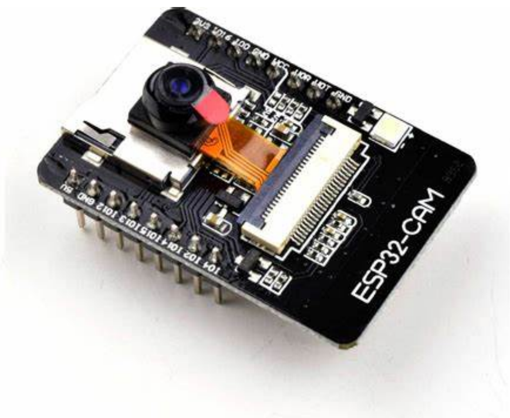
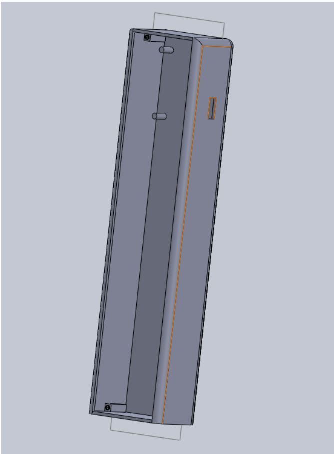
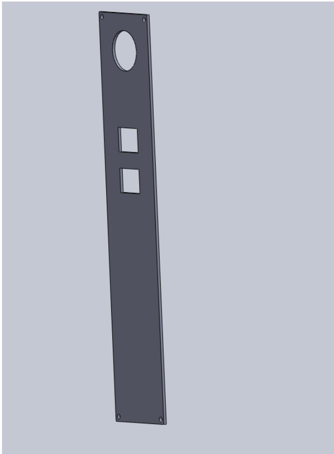
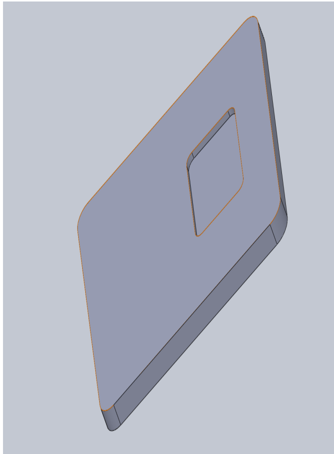
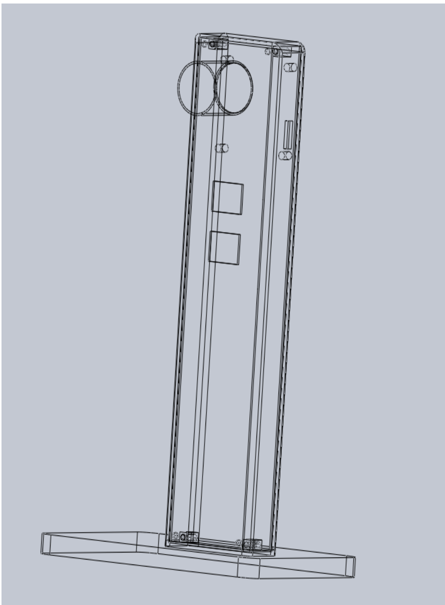
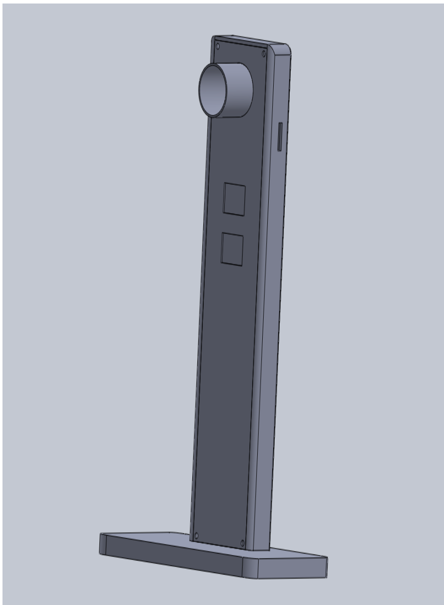


Figure 7: ESP32 Camera Module

Camera Module

Our device uses the ESP32-CAM module. Its compact size and support for microSD storage make it ideal for real-time data streaming and remote monitoring. While the module has limitations, such as limited GPIOs and no built-in USB interface, its wireless features enable seamless connectivity for smart applications.

3.2 Enclosure Design

3.3 PCB Design

One layer PCB is designed using Altium software and printed by Duino Electronics in Sri Lanka.

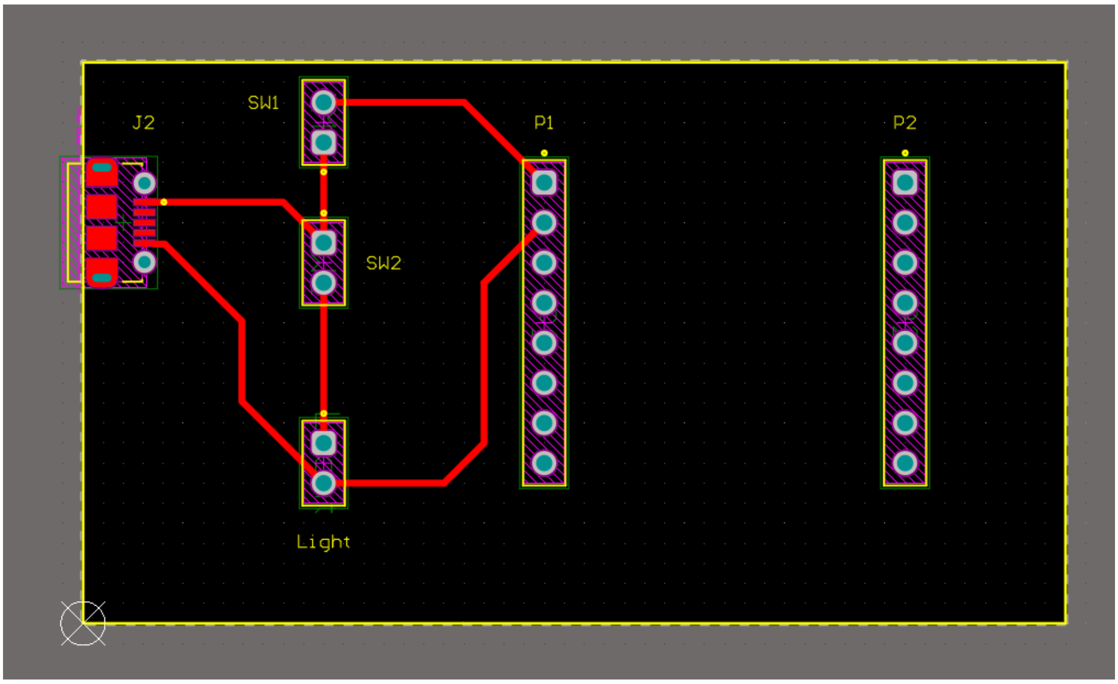


Figure 8: Altium Design

3.4 Machine Learning Integration

Key Points

- The model will self-train using a pre-existing dataset consisting of approximately 20000 images of various levels of the eye fundus.
- There is no live training involved.
- Images are represented using tensors or NumPy arrays in TensorFlow, which is based on neural networks.(CNN)
- The diabetic level of the patient is recognized through this.
  - Class 0: Normal Fundus - The eye is healthy with no visible signs of disease or abnormalities.
  - Class 1: Mild Disease - Early signs of eye conditions are present, such as slight discoloration or minimal lesions. Regular monitoring is recommended.
  - Class 2: Moderate Disease - Noticeable changes in the fundus that indicate the progression of an eye condition. Immediate consultation with a specialist is advised.
  - Class 3: Severe Disease - Advanced eye disease with significant damage. Requires urgent medical intervention.

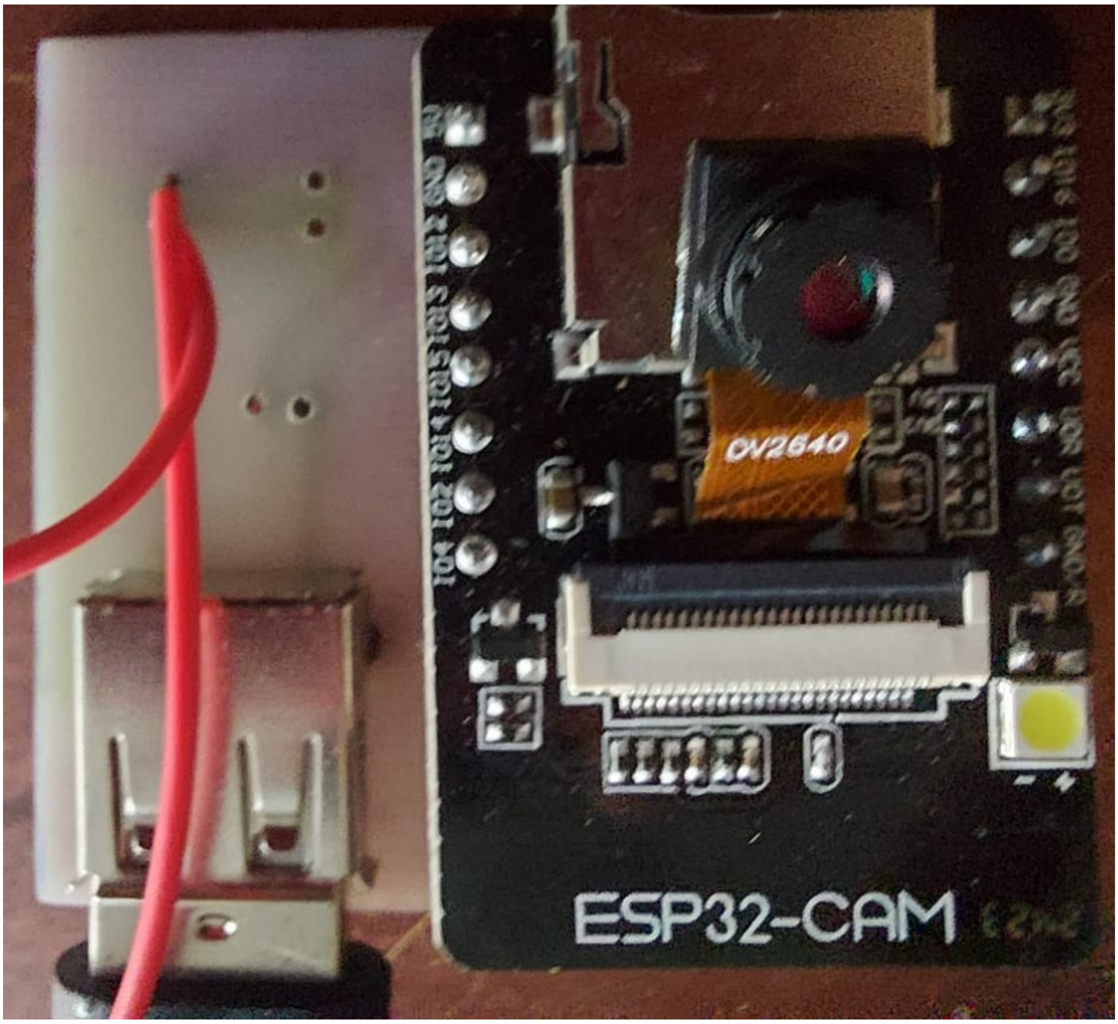


Figure 9: Printed PCB

– Class 4: Proliferative Disease - Critical condition with severe abnormalities, such as neovascularization. This can lead to vision loss without prompt treatment

Codes

Listed below are the codes we used to train and test the machine-learning model.

```
import os # file handleing
import cv2 # for image processing
import pandas as pd # iterating through rows
import numpy as np # to store arrays

image_folder = "C:\\Users\\Imesh Abeysinghe\\train_data\\
    ↪ train_data\\needed_train"
csv_file = "C:\\Users\\Imesh Abeysinghe\\trainLabels.csv\\
    ↪ trainLabels.csv"
# Load CSV
df = pd.read_csv(csv_file)

X = []  # List for images
y = []  # List for labels
missing_images = []

# Load images
for index, row in df.iterrows():
    image_path = os.path.join(image_folder, f"{row['image']}.jpeg"
        ↪ )
    if os.path.exists(image_path):
        image = cv2.imread(image_path)
        if image is not None:
            image = cv2.resize(image, (128, 128)) #resizing
                ↪ according to the UI requirement
            X.append(image)
            y.append(row['level'])
        else:
            print(f"Failed to read {image_path}")
    else:
        missing_images.append(row['image'])


if missing_images:
    print(f"Missing {len(missing_images)} images: {missing_images
        ↪ [:10]}")

# Convert lists to NumPy arrays
X = np.array(X)
y = np.array(y)

```

```python
print(f"Loaded {len(X)} images and {len(y)} labels")
print("Shape of X:", X.shape)
print("Shape of y:", y.shape)
```

```python
    from tensorflow.keras.models import Sequential # sends the output of one layer to the next sequentially
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout #  layers to train the CNN
from tensorflow.keras.utils import to_categorical
import numpy as np

# One-hot encode the labels
X = np.array(X)
y = np.array(y)
num_classes = 5
y = to_categorical(y, num_classes=num_classes)

# Create the model
def create_model(input_shape, num_classes):
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=input_shape), # how the filters are added
        MaxPooling2D((2, 2)), # reducing the outputs in a single layer
        Dropout(0.25),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D((2, 2)),
        Flatten(), # multidimension to 1D
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model

# Parameters
input_shape = (128, 128, 3)
num_classes = 5

# Create and train the model
model = create_model(input_shape, num_classes)
history = model.fit(X, y, epochs=10, validation_split=0.2, batch_size=32)

# Save the model
model.save("eye_disease_detection.keras")
```

```python
    from tensorflow.keras.models import load_model

# Load the model
model = load_model("eye_disease_detection_model.h5")

# Predict function
def predict_image(img_path, model):
    img = cv2.imread(img_path)
    img = cv2.resize(img, (128, 128)) / 255.0
    img_array = np.expand_dims(img, axis=0)
    prediction = model.predict(img_array)
    predicted_class = np.argmax(prediction)
    confidence = np.max(prediction) * 100
    return predicted_class, confidence

# Test prediction
test_image_path = "path_to_test_image.jpg"
predicted_class, confidence = predict_image(test_image_path, model
    ↪ )
print(f"Predicted Class: {predicted_class}, Confidence: {
    ↪ confidence:.2f}%")
```

3.5 User Interface

Key Points

- Streamlit is used to create the user interface.
- It predicts the levels of diabetes in the eye from stages 0 up to 4 with a certain confidence level
- Accepts images of order 128 x 128 and processes the image converting it to gray scale.

Code

```python
    import streamlit as st
from PIL import Image
import numpy as np
import tensorflow as tf
import base64


def get_base64(image_file):
    with open(image_file, "rb") as file:
        data = file.read()
    return base64.b64encode(data).decode()

```

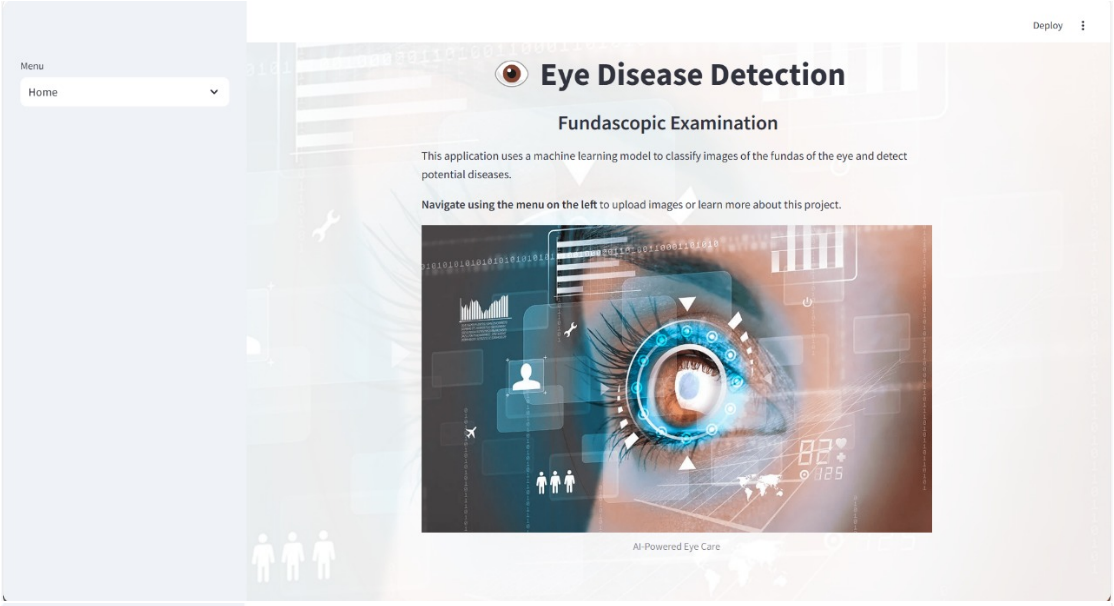
```
# Function to set the background image
def set_background(image_file):
    encoded_image = get_base64(image_file)
    css_code = f"""
    <style>
    body {{
        background-image: url("data:image/png;base64,{encoded_image}");
        background-size: cover;
        background-repeat: no-repeat;
        background-attachment: fixed;
    }}
    .stApp {{
        background: rgba(255, 255, 255, 0.9);
        padding: 20px;
        border-radius: 15px;
    }}
    h1, h2, h3 {{
        font-family: 'Arial Black', sans-serif;
        color: #ff4c4b;
        text-align: center;
    }}
    </style>
    """
    st.markdown(css_code, unsafe_allow_html=True)


set_background("C:\\Users\\Imesh Abeysinghe\\snimok-ekrana-2020-12-03-150858.png")

# Load the trained ML model
@st.cache_resource
def load_model():
    model = tf.keras.models.load_model(
        "C:\\Users\\Imesh Abeysinghe\\Downloads\\archive\\eye_disease_detection.keras"
    )
    return model

# Function to preprocess the image
def preprocess_image(uploaded_image):
    image = Image.open(uploaded_image).convert("RGB")
    resized_image = image.resize((128, 128))
    normalized_image = np.array(resized_image, dtype=np.float32) / 255.0
    input_image = np.expand_dims(normalized_image, axis=0)
    return input_image

# Main function
```

```python
def main():
    # Sidebar menu
    menu = ["Home", "Upload & Classify", "Information", "About"]
    choice = st.sidebar.selectbox("Menu", menu)

    if choice == "Home":
        st.title("\U0001F441 Eye Disease Detection")
        st.markdown(
            """
            ### Fundascopic Examination
            This application uses a machine learning model to classify images of the fundas of the eye and detect potential diseases.

            **Navigate using the menu on the left** to upload images or learn more about this project.
            """
        )
        st.image("C:\\Users\\Imesh Abeysinghe\\snimok-ekrana-2020-12-03-150858.png", caption="AI-Powered Eye Care", use_container_width=True)

    elif choice == "Upload & Classify":
        st.title("\U0001F4E4 Upload & Classify Eye Image")

        # Load the model
        model = load_model()

        # File uploader
        uploaded_image = st.file_uploader("Upload an Eye Image", type=["jpg", "jpeg", "png"])

        if uploaded_image is not None:
            # Display the original uploaded image
            st.image(Image.open(uploaded_image), caption="Uploaded Image", use_container_width=True)

            # Process the image
            st.write("\U0001F504 **Processing the image...**")
            input_image = preprocess_image(uploaded_image)

            # Check input shape compatibility
            if input_image.shape[1:] == model.input_shape[1:]:
                # Display processed image
                processed_image = (input_image[0] * 255).astype('uint8')
                st.image(processed_image, caption="Processed Image (128x128, Normalized)", use_container_width=True)
```

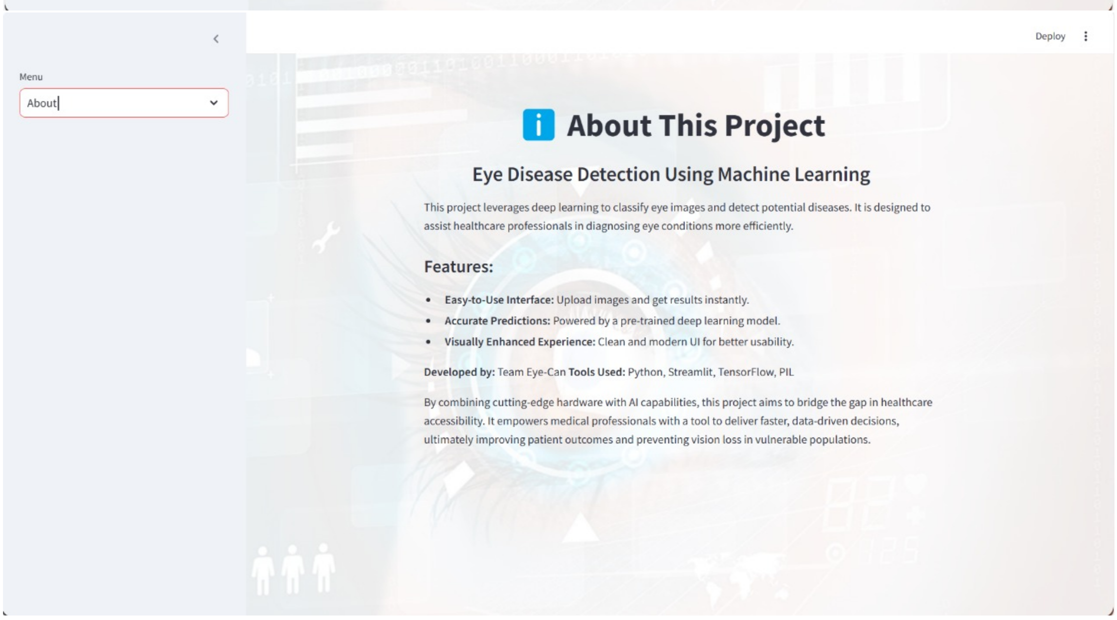
```python

                # Predict and display results
                st.write("\U0001F916 **Classifying the image...**")
                predictions = model.predict(input_image)
                predicted_class = np.argmax(predictions, axis=1)[0]
                confidence = np.max(predictions)

                # Display predictions
                st.success(f"\u2705 **Predicted Class:** {predicted_class}")
                st.info(f"\U0001F4CA **Confidence Level:** {confidence:.2f}")
            else:
                st.error("\u274C Input image size does not match model requirements.")
        else:
            st.warning("\u26A0\uFE0F Please upload an image to proceed.")

    elif choice == "Information":
        st.title("\u2139\uFE0F Information")
        st.markdown(
            """
            ### Eye Disease Levels

            #### Class 0: Normal Fundus
            The eye is healthy with no visible signs of disease or abnormalities.

            #### Class 1: Mild Disease
            Early signs of eye conditions are present, such as slight discoloration or minimal lesions. Regular monitoring is recommended.

            #### Class 2: Moderate Disease
            Noticeable changes in the fundus that indicate progression of an eye condition. Immediate consultation with a specialist is advised.

            #### Class 3: Severe Disease
            Advanced eye disease with significant damage. Requires urgent medical intervention.

            #### Class 4: Proliferative Disease
            Critical condition with severe abnormalities, such as neovascularization. Can lead to vision loss without prompt treatment.
```

```
            """
        )

    elif choice == "About":
        st.title("\u2139\uFE0F About This Project")
        st.markdown(
            """
            ### Eye Disease Detection Using Machine Learning
            This project leverages deep learning to classify eye images and detect potential diseases.
            It is designed to assist healthcare professionals in diagnosing eye conditions more efficiently.

            **Developed by:** Team Eye-Can
            **Tools Used:** Python, Streamlit, TensorFlow, PIL

            By combining cutting-edge hardware with AI capabilities, this project aims to bridge the gap in healthcare accessibility.
            It empowers medical professionals with a tool to deliver faster, data-driven decisions, ultimately improving patient outcomes and preventing vision loss in vulnerable populations.
            """
        )

if __name__ == "__main__":
    main()
```

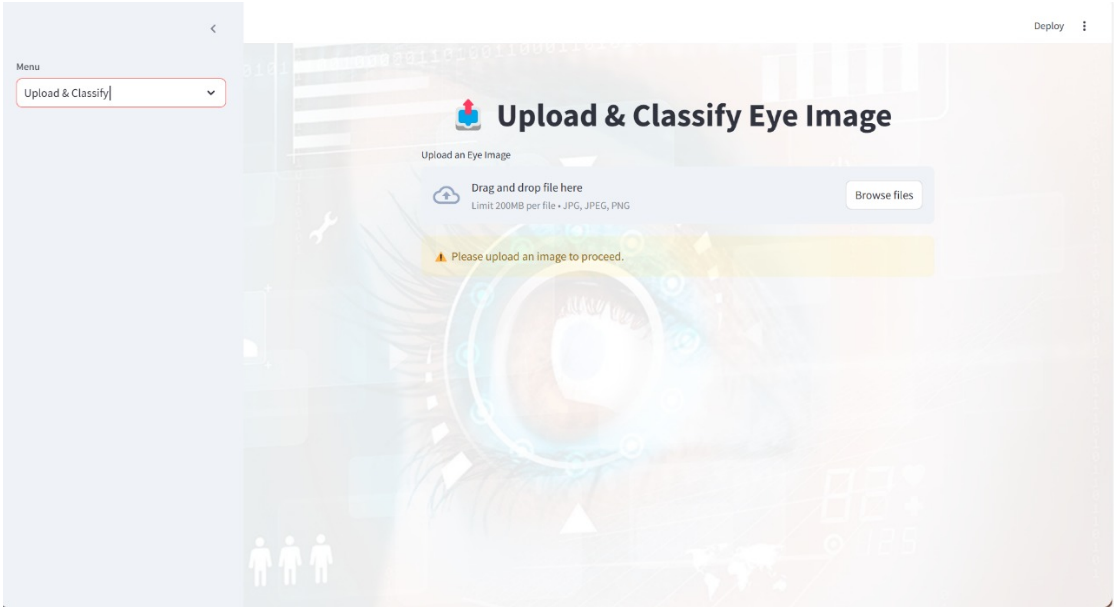
Images

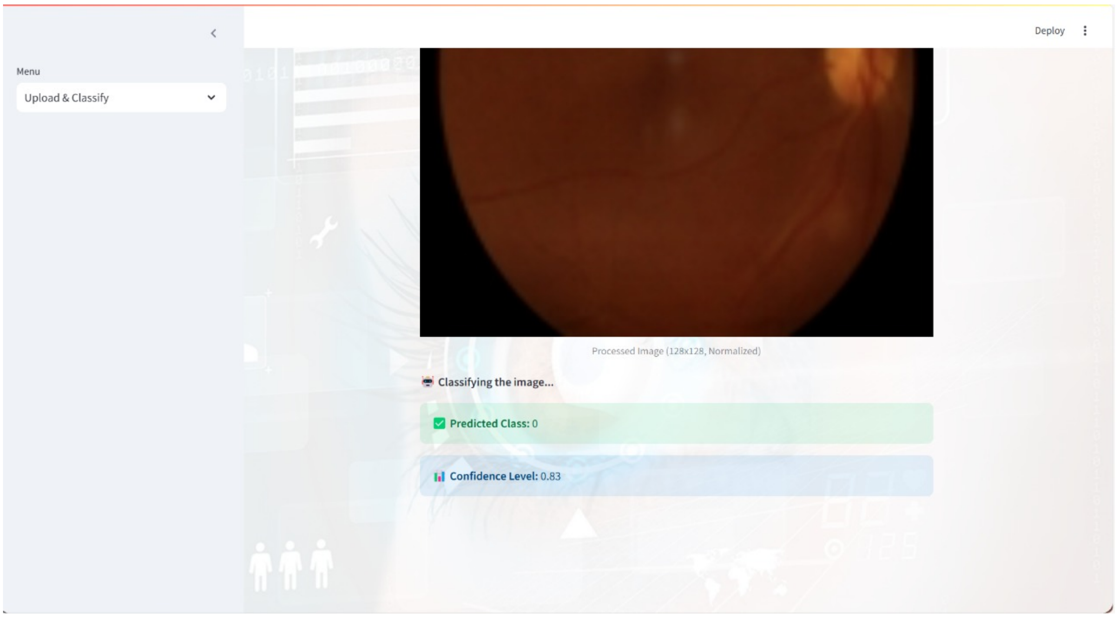


Deploy
Menu
Home
Eye Disease Detection
Fundascopic Examination
This application uses a machine learning model to classify images of the fundas of the eye and detect potential diseases.
Navigate using the menu on the left to upload images or learn more about this project.
AI-Powered Eye Care



Deploy
Menu
About
About This Project
Eye Disease Detection Using Machine Learning
This project leverages deep learning to classify eye images and detect potential diseases. It is designed to assist healthcare professionals in diagnosing eye conditions more efficiently.
Features:
Easy-to-Use Interface: Upload images and get results instantly.
Accurate Predictions: Powered by a pre-trained deep learning model.
Visually Enhanced Experience: Clean and modern UI for better usability.
Developed by: Team Eye-Can Tools Used: Python, Streamlit, TensorFlow, PIL
By combining cutting-edge hardware with AI capabilities, this project aims to bridge the gap in healthcare accessibility. It empowers medical professionals with a tool to deliver faster, data-driven decisions, ultimately improving patient outcomes and preventing vision loss in vulnerable populations.


Menu
Upload & Classify
Deploy
Upload & Classify Eye Image
Upload an Eye Image
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
Please upload an image to proceed.


Menu
Upload & Classify
Deploy
Processed Image (128x128, Normalized)
Classifying the image...
Predicted Class: 0
Confidence Level: 0.83

3.6 Final Product

4 Results and Conclusions

4.1 Results

- We could not find a suitable lens with sufficient magnification on time.
- Our model can predict the patient's diabetic level with around 70% confidence.

4.2 Conclusions

- We need to select a lens with a power greater than 20D.
  - 20D is most commonly used for indirect ophthalmoscopy.
  - Choice of lens power changes with the specific use case.
    * Lower power lenses (e.g., 15D): Provide higher magnification but a narrower field of view. These are suitable for examining specific details in smaller regions of the retina.
    * Higher power lenses (e.g., 28D, 30D): Provide a wider field of view but lower magnification. These are ideal for a broader view of the retina, such as during general screenings or examining peripheral areas.
- We should choose an improved lighting system.
  - The intensity of light suitable for a fundoscope typically ranges between 1 to 5 lumens. (To ensure that it is bright enough to illuminate the retina without causing discomfort to the eye)
  - a light source around 0.5 to 1 mW/cm^2 is generally recommended,(It minimizes retinal exposure risks while maintaining effective visualization.)
  - Adjustable intensity is ideal to accommodate individual patient needs and lighting conditions.
- The height of the column needs to be adjustable to align with eye level fitting patients of varying heights comfortably.
  - Mechanisms like telescopic poles, sliding tracks, or hydraulic systems can facilitate smooth height changes.
  - Reduces strain for both the patient
- It is essential to include a stand to stabilize the patient's head.
  - Minimizes movement that could distort images
  - Can include adjustable components like a chin rest and forehead support to accommodate different heights and postures.
- We need to select a larger data set for training the machine learning model to improve its confidence level.

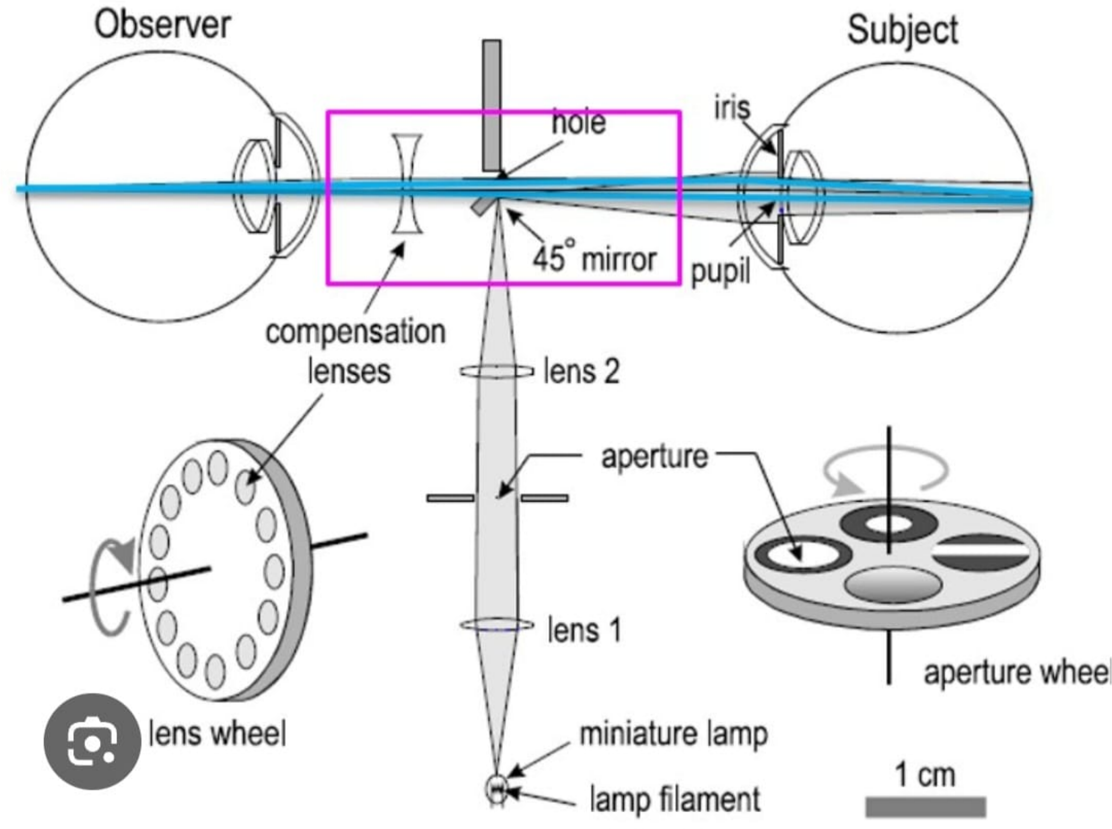


Figure 10: Lightning mechanism used in current fundoscopes

- Use a data set containing at least around 75000 train images.
- Use of intermediate classes of the fundus since its class cannot be exactly discrete as mentioned.

5 Final Budget

Component	**Price(Rs.)**
Macro Lens	835
LED ring light	699
ESP 32 camera module	2230
Rechargable 20w light	2750
ZnSe Laser Focus Lens	11114
2 switches	50
Printed PCB	2750
Enclosure	4237
Total Cost	**24,665**

6 Task Allocation

Name	**Task**
Abeysinghe G.A.I.N.M.	Machine learning integration, User interface design
Fernando S.R.N.	PCB design
Wickramasinghe S.D.	Enclosure design, Documentation

7 Possible Regulatory Pathway

1. **Device Classification**
   - Likely classified as a Class II device under 21 CFR 886.1120 (Ophthalmoscope), requiring a 510(k) Premarket Notification.
   - Identify predicate devices with similar intended use and technology (e.g., handheld fundoscopes).
2. **Preclinical Testing**
   - Performance Testing: Validate optical quality, field of view, magnification, and illumination.
   - Electrical Safety and EMC: Test compliance with IEC 60601-1 and IEC 60601-1-2.
   - Biocompatibility: Evaluate any patient-contacting materials using ISO 10993.
   - Usability: Conduct human factors studies per FDA guidelines.
3. **Software and Cybersecurity**
   - Comply with IEC 62304 for software development and include testing results in the submission.

- Address cybersecurity risks with FDA's cybersecurity guidance, such as data encryption and risk mitigation.

4. **Premarket Submission (510(k))**
   Submit:

   - Device description and intended use.
   - Performance and safety test results.
   - Risk analysis per ISO 14971.
   - Labeling (e.g., "Rx only").

5. **Manufacturing and Post-market Compliance**

   - Implement a QMS compliant with 21 CFR Part 820 (Quality System Regulation).
   - Register the device and manufacturing facility with the FDA.
   - Monitor post-market performance and report adverse events via the Medical Device Reporting (MDR) system.

8 Future Improvements

1. High-Resolution Images

   - Upgrade the camera or optics to capture higher-resolution fundus images for better disease detection.
   - Use advanced image stabilization to reduce blurring from patient or operator

2. Multimodal Imaging

   - Integrate additional imaging modalities like fluorescein angiography or optical coherence tomography (OCT) for a more comprehensive diagnostic tool.

3. Larger Dataset Training

   - Collaborate with hospitals to collect and annotate diverse datasets, improving the AI's generalization and robustness.

4. Integration with Wearables

   - Combine the funduscope with wearable devices for continuous monitoring and early warning systems for retinal issues