Deep Learning Model For Eye Disease Prediction

INTRODUCTION

1.1 Project Overview

The primary goal of this project is to develop a deep learning model that can accurately predict the presence and progression of various eye diseases using medical images, such as retinal scans or fundus photographs.Gather a diverse dataset of eye images encompassing different eye conditions, including diabetic retinopathy, age-related macular degeneration (AMD), glaucoma, etc.Clean and preprocess the collected images to ensure uniformity, correct anomalies, and standardize image formats.Experiment with various deep learning architectures (e.g., CNNs - Convolutional Neural Networks) suitable for image classification tasks.Train the selected model(s) on the preprocessed dataset, utilizing techniques like transfer learning to improve performance.Evaluate the trained models using metrics such as accuracy, precision, recall, and F1-score to assess their performance on a validation dataset.Optimize hyperparameters to enhance the model's predictive capabilities and generalization on unseen data. Create an interface or platform where clinicians or users can upload eye images, and the trained model can predict the likelihood of different eye diseases.Develop a user-friendly interface to visualize predictions and provide insights to healthcare professionals or users.

1.2 Purpose

- **Early Detection:** Enable early detection of eye diseases like diabetic retinopathy, age-related macular degeneration (AMD), glaucoma, etc., facilitating timely intervention and treatment.
- **Accuracy Enhancement:** Enhance diagnostic accuracy by utilizing deep learning models trained on a diverse dataset, potentially surpassing human capabilities in identifying subtle signs or patterns indicative of eye diseases.
- **Resource Optimization:** Assist healthcare providers in optimizing resources by streamlining diagnosis processes, prioritizing cases that require immediate attention.
- **Improved Patient Outcomes:** Early detection often leads to more effective treatment, thus potentially preventing vision loss or minimizing its impact.
- **Personalized Treatment:** Provide personalized care plans based on the identified disease, its severity, and progression.
- **Long-term Monitoring:** Enable continuous monitoring of eye health, especially for chronic conditions, allowing for proactive management and reducing the risk of complications.

LITERATURE SURVEY

2.1 Existing problem

Patients with vision problems suffer from significantly reduced quality of life. Diabetes retinopathy and glaucoma are examples of eye diseases. Globally, 64.3 million people had glaucoma in 2013 [1]. Early detection of eye diseases can aid in proper treatment or, at the very least, stop these conditions from worsening. However, it is restricted due to the lack of awareness, shortage of ophthalmologists, and high consultation costs., Hence, automated screening is critical.Using eye pictures, numerous research has attempted to anticipate and/or categorize a healthy

eye from one with a disease.Diabetic eye disease (DED) categorization relies heavily on image processing; hence offered a comprehensive study on the topic. Picture quality enrichment, image segmentation, image augmentation (geometric transformation), and classification comprise the suggested automated classification framework for DED.

2.2 References

[1] Tham, Y., Li, X., Wong, T., Quigley, H., Aung, T. & Cheng, C. Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis. Ophthalmology. 121, 2081-2090 (2014)
[2] Chelaramani, S., Gupta, M., Agarwal, V., Gupta, P. & Habash, R. Multi-task knowledge distillation for eye disease
prediction. Proceedings Of The IEEE/CVF Winter Conference On Applications Of Computer Vision. pp. 3983-3993 (2021)
[3] Nazir, T., Nawaz, M., Rashid, J., Mahum, R., Masood, M., Mehmood, A., Ali, F., Kim, J., Kwon, H. & Hussain, A.
Detection of Diabetic Eye Disease from Retinal Images Using a Deep Learning Based CenterNet Model. Sensors. 21 (2021), https://www.mdpi.com/1424-8220/21/16/5283
[4] Smaida, M. & Serhii, Y. Comparative Study of Image Classification Algorithms for Eyes Diseases Diagnostic. International Journal Of Innovative Science And Research Technology. 4 (2019)
[5] Grassmann, F., Mengelkamp, J., Brandl, C., Harsch, S., Zimmermann, M., Linkohr, B., Peters, A., Heid, I., Palm, C. & Weber, B. A deep learning algorithm for prediction of age-related eye disease study severity scale for age-related macular degeneration from color fundus photography. Ophthalmology. 125, 1410-1420 (2018)
[6] Bhatia, K., Arora, S. & Tomar, R. Diagnosis of diabetic retinopathy using machine learning classification algorithm.
2016 2nd International Conference On Next Generation Computing Technologies (NGCT). pp. 347-351 (2016)
[7] Sarki, R., Ahmed, K., Wang, H., Zhang, Y., Ma, J. & Wang, K. Image preprocessing in classification and identification
of diabetic eye diseases. Data Science And Engineering. 6, 455-471 (2021)

2.3 Problem Statement Definition

The early detection and diagnosis of eye diseases is essential for preventing vision loss and blindness. However, traditional methods of eye disease diagnosis, such as eye exams and imaging tests, can be time-consuming and expensive. Deep learning models have the potential to automate and streamline the eye disease diagnosis process, making it more accessible and affordable for everyone.The accurate and early diagnosis of eye diseases is crucial for preventing vision loss and blindness. However, traditional diagnosis methods, such as manual examination by ophthalmologists, are time-consuming, subjective, and prone to errors. Deep learning (DL) has emerged as a promising tool for automating eye disease diagnosis and achieving higher accuracy and efficiency.

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Template

Deep Learning Model For Eye Disease Prediction

An AI Doctor

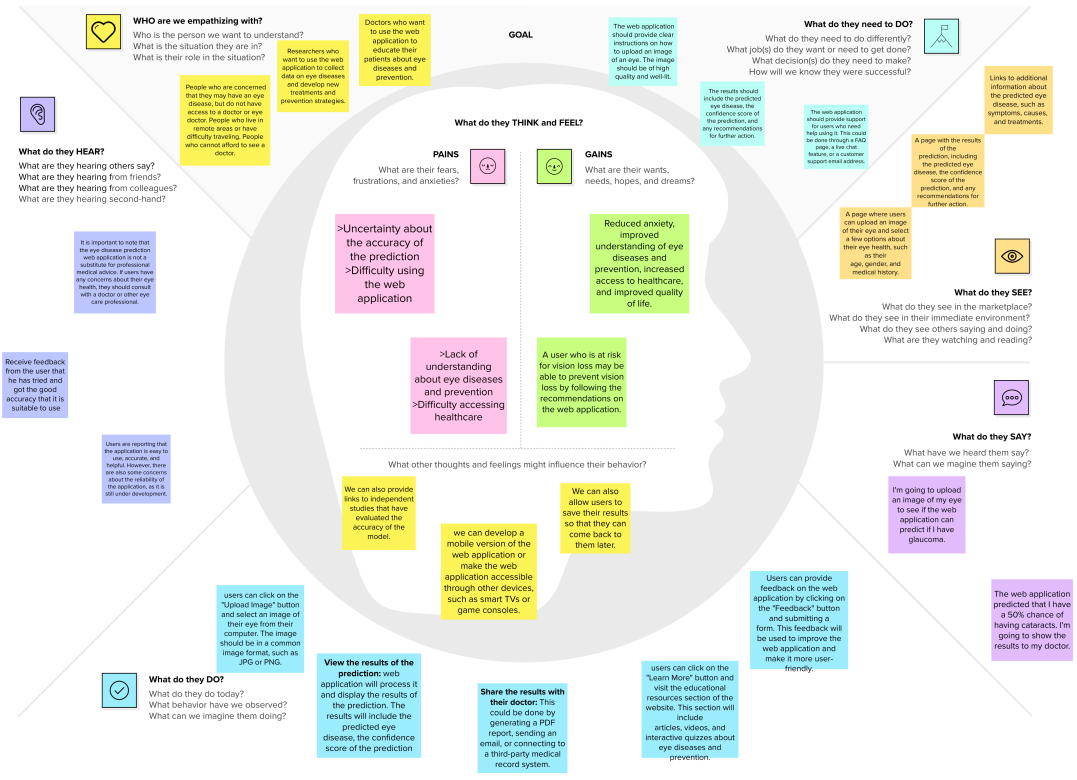
Originally created by Dave Gray at

XPLANE

Share template feedback

Deep Learning Model For Eye Disease Prediction

In this project, we are categorizing different kinds of ocular diseases that individuals can develop for a variety of reasons, such as age, diabetes, etc. The four main groups into which these disorders fall are Normal, Diabetic Retinopathy, Cataract, and Glaucoma. In artificial intelligence (AI), deep learning (DL) techniques are key components of high-performance classifiers used in the image-based diagnosis of eye diseases. One of the most widely used methods that has improved performance in many domains, particularly image analysis and classification, is transfer learning. We employed transfer learning approaches like as Inception V3, VGG19, and Xception V3, which are more popular and very successful in the field of image analysis.



3.2 Ideation & Brainstorming

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

eye diseases, specifically focusing on conditions such as diabetic retinopathy, glaucoma, macular degeneration, and cataracts. Eye diseases are a significant global health concern, and early detection and intervention are crucial to prevent irreversible vision loss. Traditional methods for detecting these diseases often rely on time-consuming and costly clinical examinations by ophthalmologists, leading to delays in diagnosis and limited accessibility to healthcare services.

5 minutes

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Person 2

Person 3

Person 4

Person 5

Person 6

Person 7

Person 8

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

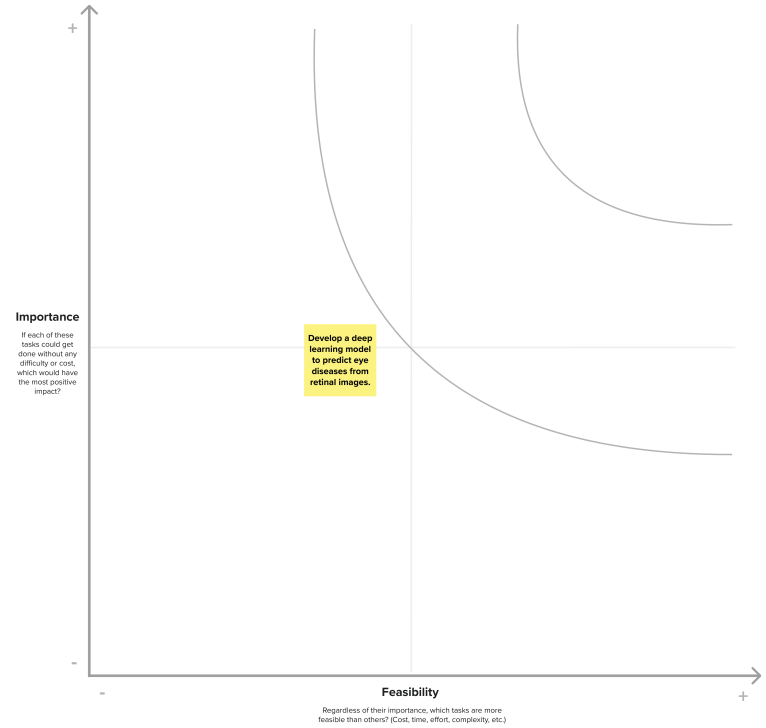
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

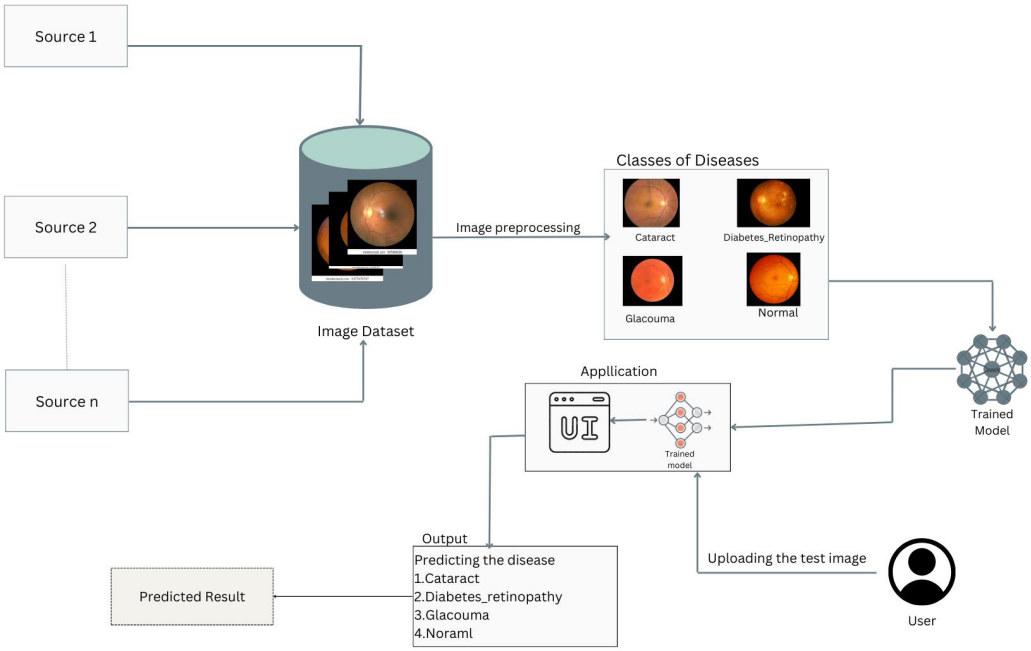
1.Project setup & Infrastructure
2.Development Environment
3.Data collection
4.Data preprocessing
5.Model Development
6.Training the model
7.Model deployment and Integration
8.Testing and quality assurance

4.2 Non-Functional requirements

1.Security Implementations.
2.Scalable Architecture
3.Availability
4.Performance

5. PROJECT DESIGN

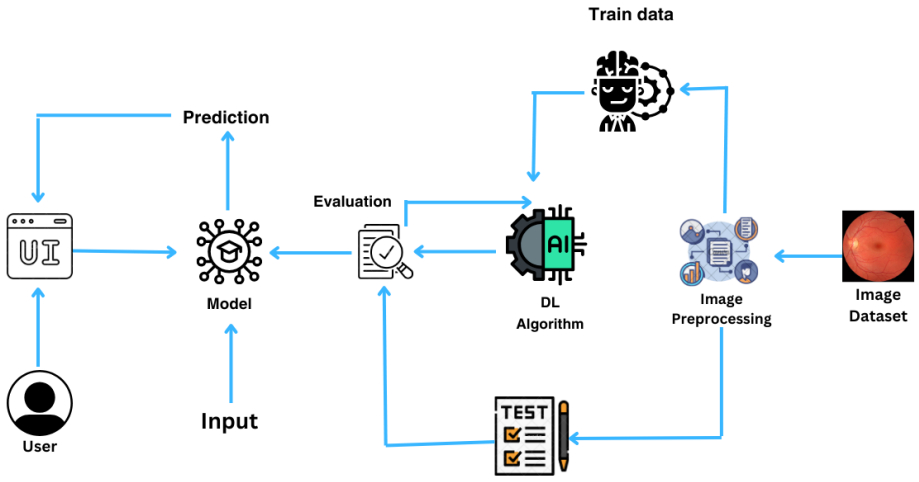
5.1 Data Flow Diagrams & User Stories



5.2 Solution Architecture

The solution architecture is designed to be scalable and reliable. The cloud-based services would allow the solution to handle large volumes of data and traffic. The solution would also be highly available, with multiple redundant components to ensure that it is always available to users.
The solution architecture is also designed to be flexible. The different components of the solution can be customized to meet the specific needs of the users. For example, the user interface can be customized to match the branding of the organization that is using the solution.
Overall, the solution architecture for the deep learning model for eye disease prediction is designed to provide a scalable, reliable, and flexible solution for eye disease diagnosis.

- Data Storage
- Data Preprocessing
- Model Building
- Disease Prediction
- Model deployment
- User Interface
- Security



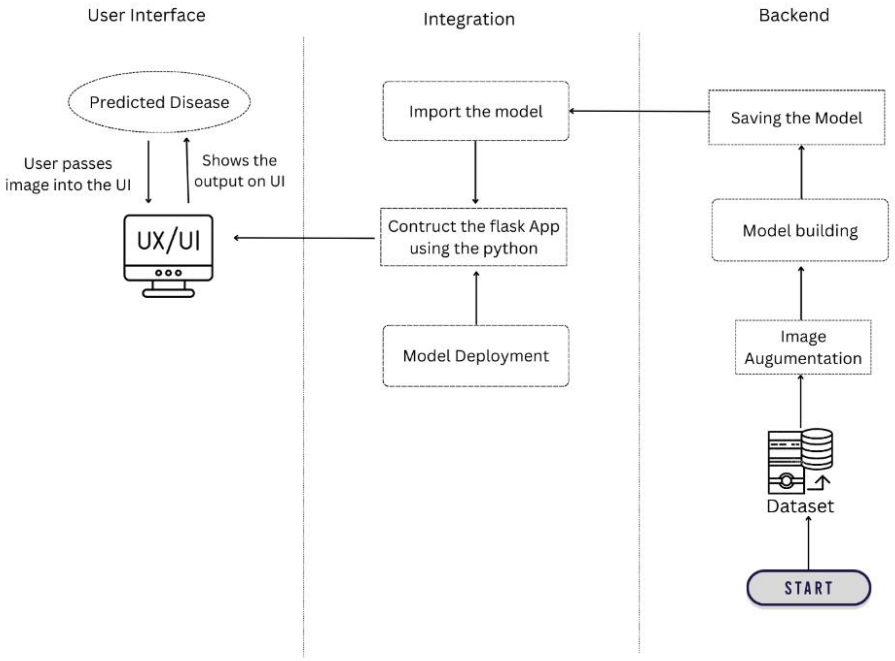
6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Components and Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Database	Collect the Dataset Based on the ProblemStatement	File Manager, Google drive
4.	File Storage/ Data	File storage requirements for Storing the dataset	Google Drive
5.	Frame Work	Used to Createa web Application, IntegratingFrontend and Back End	Python Flask
6.	Deep Learning Model	Purpose of Model	CNN, Transfer Learning
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal ServerConfiguration:5000 port Cloud Server Configuration:	pythonanywhere platform

Thechnical Architecture:



6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Project setup& Infrastructure	USN-1	Set up the development environment with the required tools and frameworks to predict the eye disease.	1	High	Varsha
Sprint-1	development environment	USN-2	Gather a diverse dataset of images containing different types of eye diseases for training the deep learning model.	2	High	Varsha
Sprint-2	Data collection	USN-3	Preprocess the collected datasetby resizing images,normalizing pixel values, and splitting it into training and validation sets.	2	High	Sushma
Sprint-2	data preprocessing	USN-4	Explore and evaluate different deep learning architectures (e.g., CNNs, Transfer learning) to select the most suitable model for garbage classification.	3	High	Sushma
Sprint-3	model development	USN-5	train the selected deep learning modelusing the preprocessed dataset and monitorits performance on the validation set.	4	High	Vivek
Sprint-3	Training	USN-6	implement data augmentation techniques (e.g., rotation, flipping)to improve the model's robustness and accuracy.	6	medium	Vivek
Sprint-4	model deployment & Integration	USN-7	deploy the trained deep learning model as an API or web service to make it accessible for common people. integrate the model's API into a user-friendly web interface for users to upload images and get the accurate results.	1	medium	Vivek

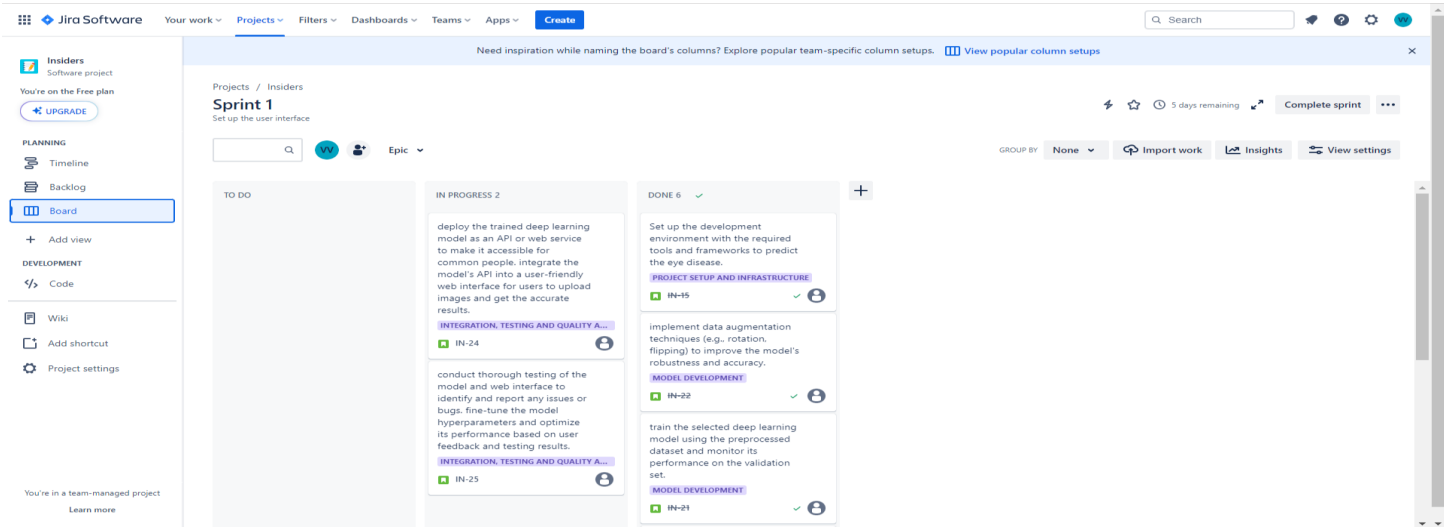
Sprint-5	Testing & quality assurance	USN-8	conduct thorough testing of the model and web interface to identify and report any issues or bugs. fine-tune the model hyperparameters and optimize its performance basedon user feedback and testing results.	1	medium	Roopesh

6.3 Sprint Delivery Schedule

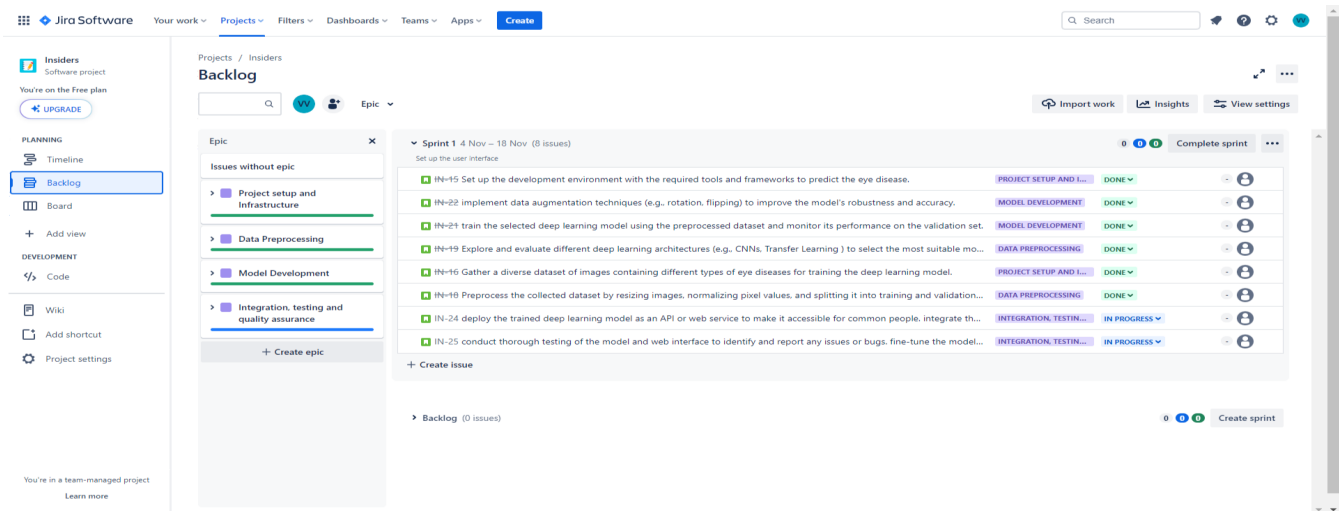
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	3	3 Days	25 oct 2023	27 oct 2023	3	27 oct 2023
Sprint-2	5	3 Days	27 oct 2023	01 Nov 2023	5	
Sprint-3	10	5 Days	01 Nov 2023	05 Nov 2023	10	
Sprint-4	1	7 Days	05 Nov 2023	12 Nov 2023	1	
Sprint-5	1	7 Days	13 Nov 2023	18 Nov 2023	1	

Board section.

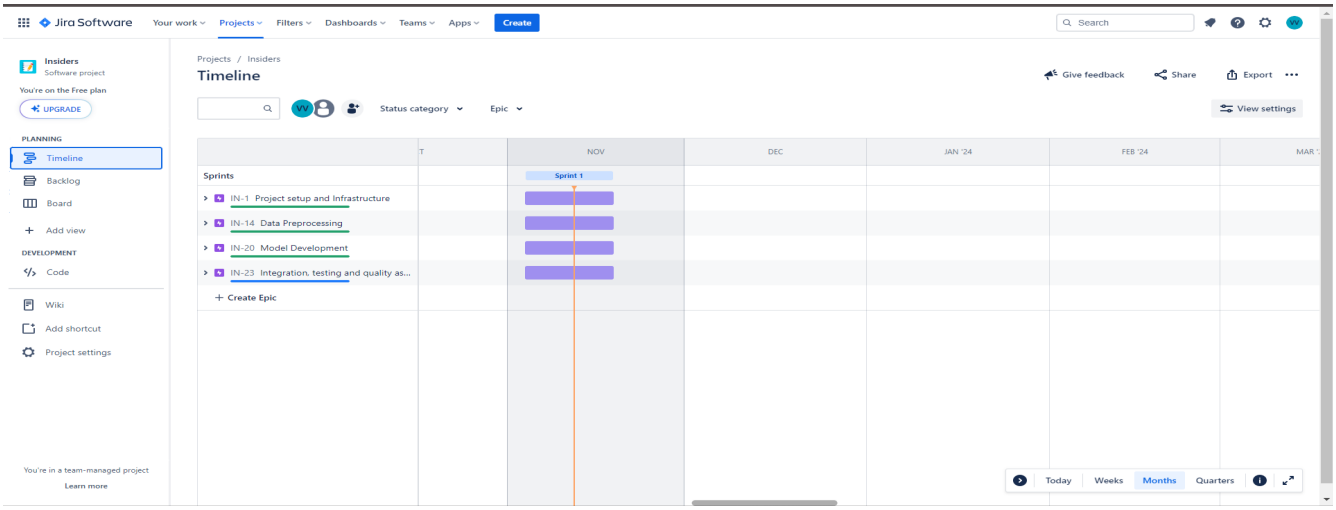
We have completed sprint 1, 2 and 3. So we can see the remaining tasks on board.



Backlog section



Timeline



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Data collection

```
[ ] !pip install -q kaggle
```

```
[ ] !mkdir ~/.kaggle
```

```
[ ] !cp kaggle.json ~/.kaggle/
```

```
[ ] !kaggle datasets download -d gunavenkatdoddi/eye-diseases-classification
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Downloading eye-diseases-classification.zip to /content
 99% 729M/736M [00:06<00:00, 160MB/s]
100% 736M/736M [00:06<00:00, 117MB/s]
```

```
[ ] !pip install split-folders
```

```
Collecting split-folders
  Downloading split_folders-0.5.1-py3-none-any.whl (8.4 kB)
Installing collected packages: split-folders
Successfully installed split-folders-0.5.1
```

```
[ ] import splitfolders
```

```
[ ] splitfolders.ratio('/content/dataset', output="output",seed=1337, ratio=(0.8,0.2))
```

```
Copying files: 4217 files [00:02, 1627.08 files/s]
```

Assigning the input image size

```
[ ] IMAGE_SIZE = [224,224]
```

Importing the necessary libraries

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import matplotlib.image as mpimg
    %matplotlib inline
    from tensorflow.keras.preprocessing.image import ImageDataGenerator
    from tensorflow.keras.preprocessing import image
    from PIL import ImageFile
    ImageFile.LOAD_TRUNCATED_IMAGES = True
```

Configuring the ImageDataGenerator

```
[ ] train_datagen = ImageDataGenerator(rescale=1./255,
                                       shear_range = 0.2, zoom_range = 0.2,
                                       horizontal_flip = True)

    test_datagen = ImageDataGenerator(rescale = 1./255)
```

Splitting the data into train and test set

```
train_data = train_datagen.flow_from_directory(
    '/content/output/train',
    target_size = (224,224),
    class_mode = 'categorical'
)

test_data = test_datagen.flow_from_directory('/content/output/val',
                          target_size = (224,224),
                          batch_size = 64,
                          class_mode = 'categorical')
```

```
Found 3372 images belonging to 4 classes.
Found 845 images belonging to 4 classes.
```

Model building

```
from tensorflow.keras.applications.vgg19 import VGG19, preprocess_input
from tensorflow.keras.layers import Flatten, Dense
from tensorflow.keras.models import Model
from tensorflow.keras.models import load_model
```

```
VGG19 = VGG19(input_shape=IMAGE_SIZE + [3], weights='imagenet',include_top=False)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
80134624/80134624 [==============================] - 0s 0us/step
```

```
for layers in VGG19.layers:
  layers.trainable = False
```

```
x = Flatten()(VGG19.output)
```

```
prediction = Dense(4, activation='softmax')(x)
```

```
model = Model(inputs=VGG19.inputs, outputs=prediction)
```

```
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

```
model.fit(
    train_data,
    validation_data=test_data,
    epochs=60,
    steps_per_epoch=len(train_data),
    validation_steps=len(test_data)
)#fitting the model
```

Save the model

```
model.save('/content/drive/MyDrive/Eye Disease Detection/model.h5')
```

7.2 Feature 2

Testing the model

```
[1] from tensorflow.keras.models import load_model
    model = load_model('/content/drive/MyDrive/Eye Disease Detection/model.h5')
```

```
import numpy as np
from tensorflow.keras.preprocessing import image
img = image.load_img(r"/content/drive/MyDrive/Eye Disease Detection/Test images/Glaucoma 4.jpeg",target_size = (224,224)) #loading the model
x = image.img_to_array(img) #image to array
x = np.expand_dims(x,axis = 0)#changing the shape
preds = model.predict(x)
pred=np.argmax(preds,axis=1)
index=['cataract','diabetic_retinopathy','glaucoma','normal']
result=str(index[pred[0]])
confidence_score=max(preds[0][pred])
print("Predicted result is ",result)
print("Confidence Score : ",confidence_score)
```

```
1/1 [==============================] - 0s 26ms/step
Predicted result is  glaucoma
Confidence Score :  1.0
```

8. PERFORMANCE TESTING

8.1 Performace Metrics

Training Accuracy -91.58

Validation Accuracy – 86.15

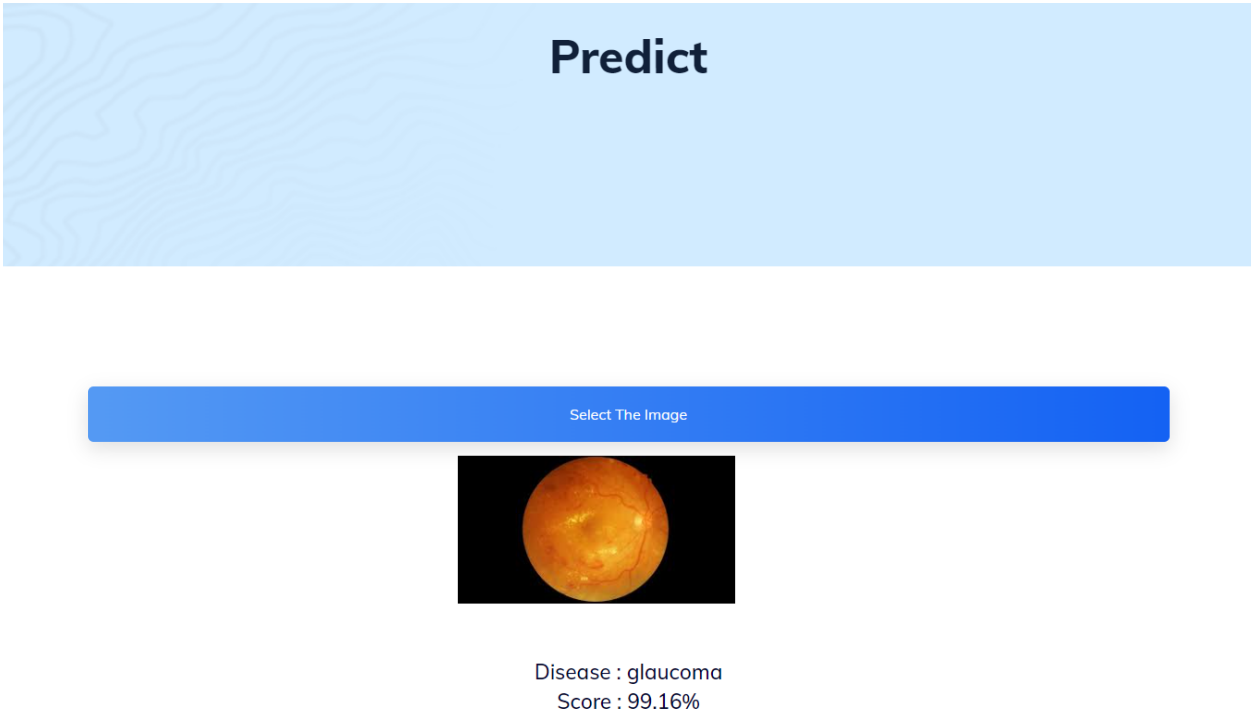
```
Epoch 40/60
106/106 [==============================] - 72s 675ms/step - loss: 0.2465 - accuracy: 0.8998 - val_loss: 0.3450 - val_accuracy: 0.8947
Epoch 41/60
106/106 [==============================] - 71s 667ms/step - loss: 0.2295 - accuracy: 0.9081 - val_loss: 0.4681 - val_accuracy: 0.8426
Epoch 42/60
106/106 [==============================] - 71s 669ms/step - loss: 0.2542 - accuracy: 0.8989 - val_loss: 0.5440 - val_accuracy: 0.8284
Epoch 43/60
106/106 [==============================] - 74s 702ms/step - loss: 0.2366 - accuracy: 0.9084 - val_loss: 0.7972 - val_accuracy: 0.7704
Epoch 44/60
106/106 [==============================] - 71s 665ms/step - loss: 0.2451 - accuracy: 0.9012 - val_loss: 0.7485 - val_accuracy: 0.7740
Epoch 45/60
106/106 [==============================] - 74s 700ms/step - loss: 0.2457 - accuracy: 0.9095 - val_loss: 0.3772 - val_accuracy: 0.8734
Epoch 46/60
106/106 [==============================] - 75s 702ms/step - loss: 0.2658 - accuracy: 0.8968 - val_loss: 0.5034 - val_accuracy: 0.8391
Epoch 47/60
106/106 [==============================] - 73s 687ms/step - loss: 0.2424 - accuracy: 0.9039 - val_loss: 0.5629 - val_accuracy: 0.8166
Epoch 48/60
106/106 [==============================] - 71s 674ms/step - loss: 0.2973 - accuracy: 0.8923 - val_loss: 0.4298 - val_accuracy: 0.8793
Epoch 49/60
106/106 [==============================] - 70s 659ms/step - loss: 0.2643 - accuracy: 0.9007 - val_loss: 0.3938 - val_accuracy: 0.8840
Epoch 50/60
106/106 [==============================] - 72s 678ms/step - loss: 0.2386 - accuracy: 0.9119 - val_loss: 0.5152 - val_accuracy: 0.8414
Epoch 51/60
106/106 [==============================] - 73s 691ms/step - loss: 0.2625 - accuracy: 0.8950 - val_loss: 0.3649 - val_accuracy: 0.8935
Epoch 52/60
106/106 [==============================] - 70s 658ms/step - loss: 0.2606 - accuracy: 0.9012 - val_loss: 0.4582 - val_accuracy: 0.8355
Epoch 53/60
106/106 [==============================] - 70s 664ms/step - loss: 0.2500 - accuracy: 0.9015 - val_loss: 0.4878 - val_accuracy: 0.8580
Epoch 54/60
106/106 [==============================] - 71s 670ms/step - loss: 0.2929 - accuracy: 0.8870 - val_loss: 0.4236 - val_accuracy: 0.8722
Epoch 55/60
106/106 [==============================] - 72s 676ms/step - loss: 0.2344 - accuracy: 0.9125 - val_loss: 0.5679 - val_accuracy: 0.8272
Epoch 56/60
106/106 [==============================] - 70s 663ms/step - loss: 0.2506 - accuracy: 0.9018 - val_loss: 0.4533 - val_accuracy: 0.8402
Epoch 57/60
106/106 [==============================] - 71s 673ms/step - loss: 0.2330 - accuracy: 0.9116 - val_loss: 0.3510 - val_accuracy: 0.8852
Epoch 58/60
106/106 [==============================] - 91s 862ms/step - loss: 0.2471 - accuracy: 0.9048 - val_loss: 0.5656 - val_accuracy: 0.8189
Epoch 59/60
106/106 [==============================] - 71s 663ms/step - loss: 0.2429 - accuracy: 0.9057 - val_loss: 0.4903 - val_accuracy: 0.8663
Epoch 60/60
106/106 [==============================] - 76s 719ms/step - loss: 0.2220 - accuracy: 0.9158 - val_loss: 0.5024 - val_accuracy: 0.8615
<keras.src.callbacks.History at 0x78ece16cb100>
```

9. RESULTS

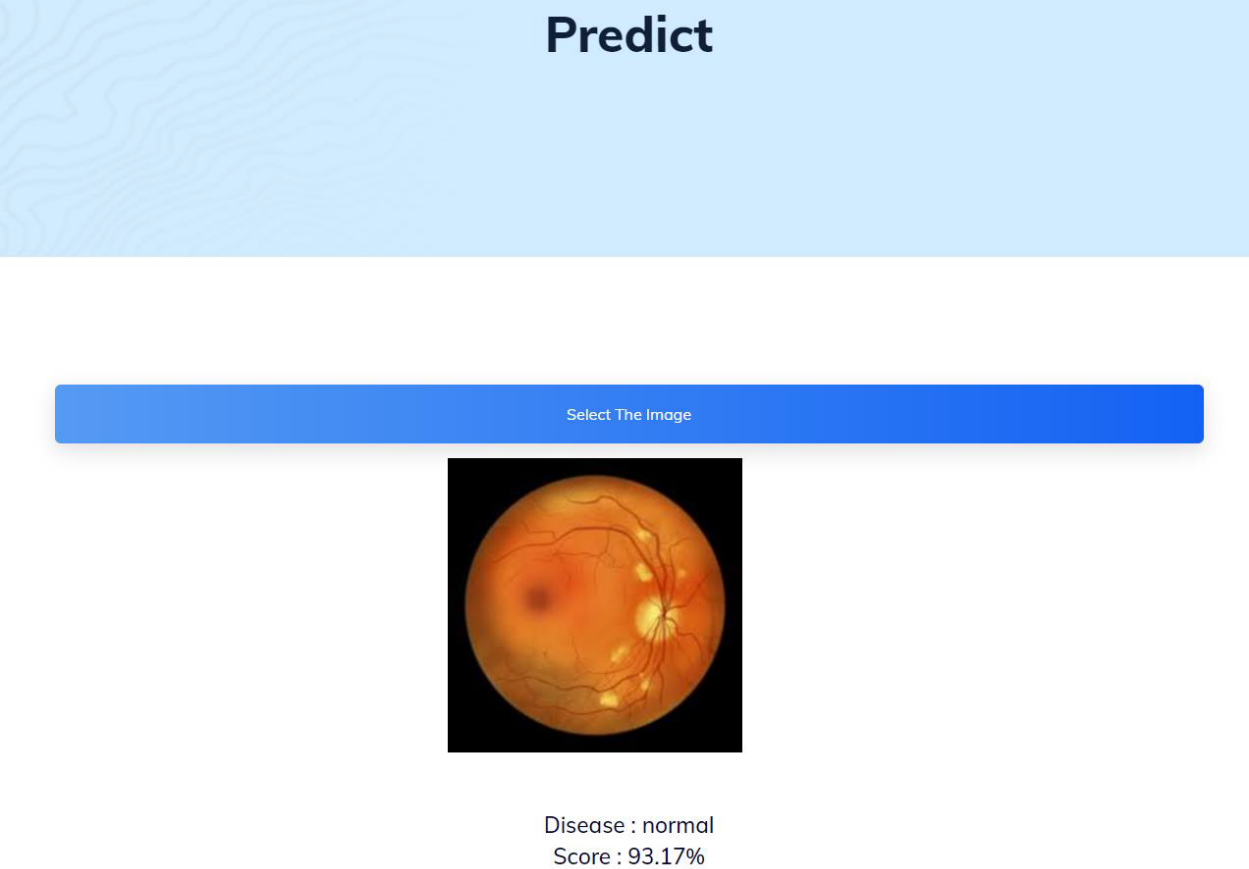
9.1 Output Screenshots

```
1/1 [==============================] - 0s 18ms/step
Predicted result is  diabetic_retinopathy
```

Output 1:



Output 2:



10. ADVANTAGES & DISADVANTAGES

Advantages of Deep Learning Models for Eye Disease Prediction

- Automated Feature Learning
- Handling Large and Complex Data
- Improved Performance
- Early Disease Detection
- Reducing Human Error
- Accessibility in Remote Areas

Disadvantages of Deep Learning Models for Eye Disease Prediction

- Data Dependency
- Explainability
- Bias and Generalizability
- Computational Requirements
- Overfitting
- Regulatory Approval

11. CONCLUSION

The eye is one of the most important sense organs in human anatomy, and losing vision would significantly affect the quality of life. Besides, the eye could also indicate some serious health issues. Unfortunately, many people may not be aware of these health problems due to the lack of ophthalmologists and access to eye care. Nevertheless, the advent of deep learning and image processing could immensely help in detecting and diagnosing these diseases. This study used CNN and transfer learning to detect three eye diseases: cataracts, diabetic retinopathy, and glaucoma. While CNN alone is not good enough for classifying eye diseases, its performance significantly increased with transfer learning. The pre-trained model of EfficientNet is transferred to the CNN model using transfer learning. For future research, we plan to investigate the efficacy of transfer learning on a more diverse dataset that includes other types of eye diseases. And explore other applications of transfer learning in disease detection tasks.

12. FUTURE SCOPE

For future research, we plan to investigate the efficacy of transfer learning on a more diverse dataset that includes other types of eye diseases. And explore other applications of transfer learning in disease detection tasks.

13. APPENDIX

Source Code

https://drive.google.com/drive/folders/1UNOOzQpv0_PjGX4tMexo0Fkn-BeVKBvt?usp=sharing

GitHub & Project Demo Link

https://github.com/smartinternz02/SI-GuidedProject-612305-1698644147