14 JULY 2023

DIABETES RETINOPATHY APPLICATION PROTOTYPE

LESS MONEY, MORE CARE

TEAM LEADER – KHUSHI GUPTA

TEAM MEMBERS – SUDIKSHA BHAT M

VAIBHAV DUTTA

INDEX

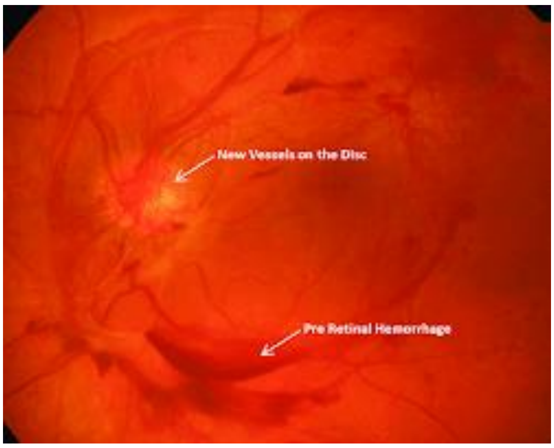
WHAT IS DIABETES RETINOPATHY?

Diabetic retinopathy is a condition that affects the blood vessels in the retina, the light-sensitive tissue at the back of the eye. As a result of diabetes, the blood vessels in the retina can become damaged, leaky, or blocked. This can lead to a number of symptoms, including:

- Microaneurysms: These are small, red dots that appear on the retina. They are caused by bulges in the walls of the blood vessels.

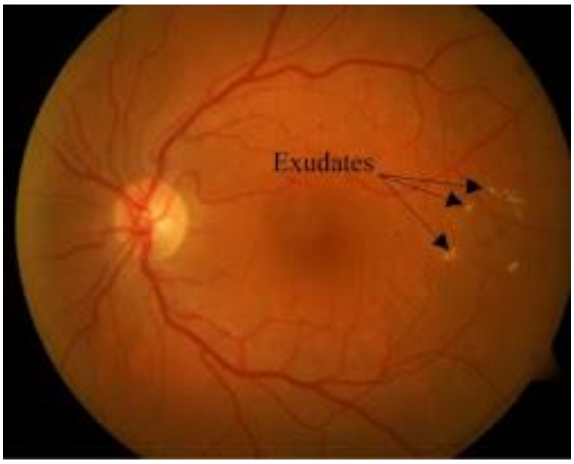
Microaneurysms diabetic retinopathy

- Intraretinal haemorrhage: This is bleeding within the retina. It can appear as small, red spots or as larger, blotchy areas.



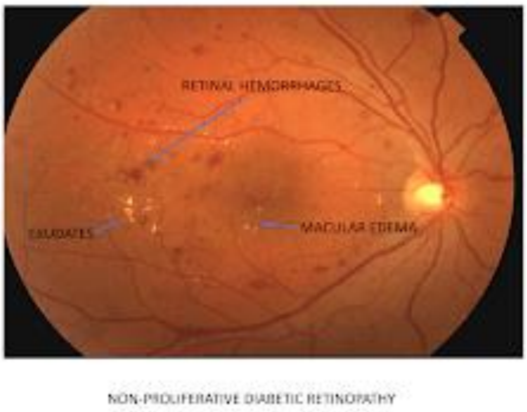
Intraretinal haemorrhage diabetic retinopathy

- Exudates: These are white or yellow spots that appear on the retina. They are caused by the leakage of fluid or proteins from the blood vessels.



Exudates diabetic retinopathy

- Macular edema: This is swelling of the macula, the part of the retina responsible for central vision. It can cause blurred vision or distortion of images.



Macular edema diabetic retinopathy

If you have any of these symptoms, it is important to see an eye doctor immediately. Diabetic retinopathy can be treated, but the earlier it is diagnosed, the better the chances of preventing vision loss.

Our application/website will help users to check if they are having any of the above types of diabetes retinopathy and can take adequate measures to prevent further damage to the eyes.

PROBLEM STATEMENT

The inability to monitor and detect eye disorders early due to a lack of inexpensive and accessible fundus imaging technology results in delayed diagnosis and higher healthcare expenses. Healthcare practitioners will have access to a practical and affordable instrument to take high-quality retinal pictures with the help of a mobile app that is integrated with a fundus camera. This will enable quicker diagnosis and better management of eye problems.

Business Need Assessment

To conduct a business need assessment for a diabetic retinopathy app, you would typically evaluate several key factors:

1. Market Analysis: Assess the current market conditions for diabetic retinopathy apps, including the size of the target audience, competition, and growth potential. Identify any gaps or opportunities that your app can address.

2. User Needs: Understand the specific needs of individuals with diabetic retinopathy and related stakeholders such as healthcare professionals. Conduct surveys, interviews, and user testing to gather insights into their requirements, challenges, and expectations.

3. Regulatory Considerations: Investigate the regulatory requirements and guidelines applicable to medical or health-related apps, especially those related to diabetic retinopathy. Compliance with privacy, security, and data protection regulations is crucial.

4. Technological Feasibility: Evaluate the technical feasibility of developing an app for diabetic retinopathy. Consider factors such as data integration with existing medical systems, compatibility with various devices and platforms, and the availability of necessary resources and expertise.

5. Business Model: Define your app's revenue model, considering options like upfront purchase, subscription, freemium, or in-app purchases. Identify potential revenue streams and assess their viability based on the target market and competition.

6. Cost Analysis: Estimate the development, maintenance, and operational costs associated with the app. Consider factors such as software development, infrastructure, hosting, ongoing updates, customer support, and marketing expenses.

Target Specifications and Characterization of Diabetic Retinopathy App:

Based on the business need assessment, the target specifications and characterization of the diabetic retinopathy app may include the following:

1. User Interface: Design an intuitive and user-friendly interface that accommodates individuals with varying levels of technical expertise and visual impairments. Ensure the app adheres to accessibility guidelines for users with disabilities.

2. Retinal Image Analysis: Incorporate a robust and accurate algorithm for analyzing retinal images to detect signs of diabetic retinopathy. The algorithm should assess the severity of the

condition, identify abnormalities, and provide relevant recommendations for further medical intervention.

3. Integration with Medical Systems: Enable seamless integration with electronic health record (EHR) systems or other medical platforms to exchange patient data securely. This integration should allow healthcare professionals to access and review patient information, track progress, and collaborate effectively.

4. Personalized Recommendations: Provide personalized recommendations based on the user's condition, medical history, and specific needs. These recommendations can include reminders for eye exams, medication schedules, lifestyle modifications, and self-care instructions.

5. Education and Awareness: Offer educational content about diabetic retinopathy, its symptoms, risk factors, preventive measures, and available treatment options. Include visual aids, interactive modules, and frequently asked questions to enhance user knowledge and promote proactive management of the condition.

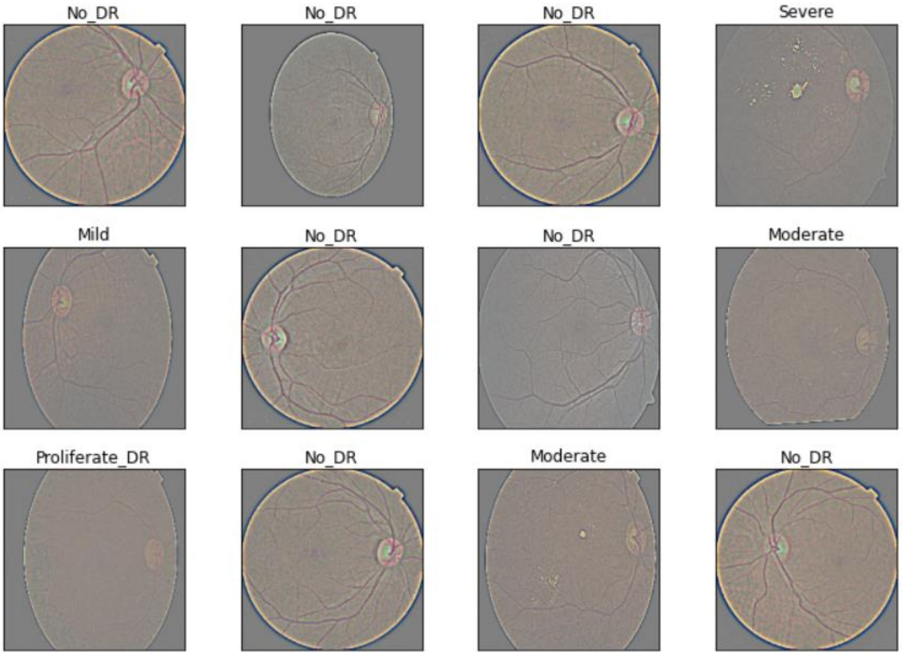
6. Notifications and Alerts: Implement timely notifications and alerts to remind users about medication schedules, upcoming appointments, and recommended activities. Customizable reminders can help individuals adhere to their treatment plans and maintain regular eye care.

7. Data Security and Privacy: Prioritize data security and privacy measures to protect sensitive user information. Adhere to industry best practices for data encryption, secure storage, user consent management, and compliance with applicable regulations, such as GDPR or HIPAA.

8. Collaboration with Healthcare Professionals: Facilitate communication and collaboration between users and their healthcare providers. Enable users to share their retinal images, progress reports, and questions with medical professionals, fostering a collaborative approach to managing diabetic retinopathy.

Remember that these specifications and characteristics are general guidelines, and the final product should be tailored to the specific needs of your target audience and in compliance with relevant regulations and standards.

SAMPLE DATASET:



STEP 1: Importing libraries

MobileNetV2_MP3.ipynb

File Edit View Insert Runtime Tools Help Last edited on February 2

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
from IPython.display import Image, display, Markdown
import matplotlib.cm as cm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from time import perf_counter
import seaborn as sns
import os
import cv2
def printmd(string):
    # Print with Markdowns
    display(Markdown(string))
```

```
image_dir = Path('/content/drive/MyDrive/New_DR/gaussian_filtered_images/gaussian_filtered_images')
# Get filepaths and labels
filepaths = list(image_dir.glob(r'**/*.png'))
labels = list(map(lambda x: os.path.split(os.path.split(x)[0])[1], filepaths))
```

```python
filepaths = pd.Series(filepaths, name='Filepath').astype(str)
labels = pd.Series(labels, name='Label')

# Concatenate filepaths and labels
image_df = pd.concat([filepaths, labels], axis=1)

# Shuffle the DataFrame and reset index
image_df = image_df.sample(frac=1).reset_index(drop = True)

# Show the result
image_df.head()
```

	Filepath	Label
0	/content/drive/MyDrive/New_DR/gaussian_filtere...	No_DR
1	/content/drive/MyDrive/New_DR/gaussian_filtere...	No_DR
2	/content/drive/MyDrive/New_DR/gaussian_filtere...	No_DR
3	/content/drive/MyDrive/New_DR/gaussian_filtere...	Severe
4	/content/drive/MyDrive/New_DR/gaussian_filtere...	Mild

STEP 2: Display some pictures of the dataset with their labels

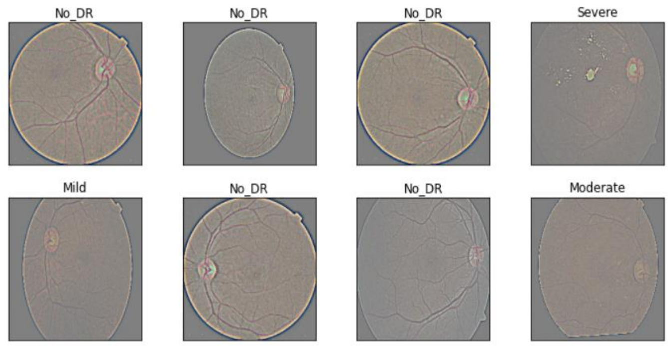
MobileNetV2_MP3.ipynb

File Edit View Insert Runtime Tools Help Last edited on February 2

Comment Share

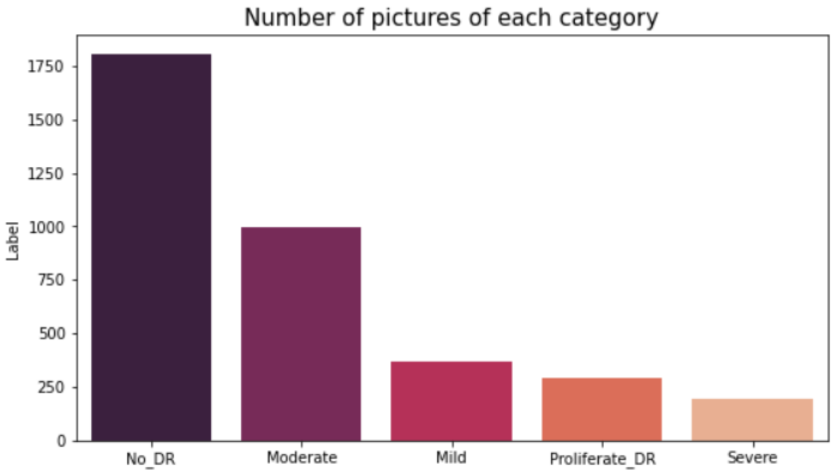
+ Code + Text Connect

```python
# Display some pictures of the dataset with their labels
fig, axes = plt.subplots(nrows=3, ncols=4, figsize=(10, 7),
                        subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(image_df.Filepath[i]))
    ax.set_title(image_df.Label[i])
plt.tight_layout()
plt.show()
```



STEP3: Display the number of pictures of each category

```
# Display the number of pictures of each category
vc = image_df['Label'].value_counts()
plt.figure(figsize=(9,5))
sns.barplot(x = vc.index, y = vc, palette = "rocket")
plt.title("Number of pictures of each category", fontsize = 15)
plt.show()
```



STEP 4: Loading the Images with a Generator and Data Augmentation

```
def create_gen():
    # Load the Images with a generator and Data Augmentation
    train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input,
        validation_split=0.1
    )

    test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
        preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
    )

    train_images = train_generator.flow_from_dataframe(
        dataframe=train_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=True,
        seed=0,
        subset='training',
        rotation_range=30, # Uncomment to use data augmentation
        zoom_range=0.15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.15,
```

```
        horizontal_flip=True,
        fill_mode="nearest"
    )

    test_images = test_generator.flow_from_dataframe(
        dataframe=test_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=False
    )

    return train_generator,test_generator,train_images,val_images,test_images
```

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

+ Code + Text

```
def get_model(model):
# Load the pretained model
    kwargs =    {'input_shape':(224, 224, 3),
                'include_top':False,
                'weights':'imagenet',
                'pooling':'avg'}

    pretrained_model = model(**kwargs)
    pretrained_model.trainable = False

    inputs = pretrained_model.input

    x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
    x = tf.keras.layers.Dense(128, activation='relu')(x)

    outputs = tf.keras.layers.Dense(5, activation='softmax')(x)

    model = tf.keras.Model(inputs=inputs, outputs=outputs)

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

    return model
```

STEP 5: Separating into train and test data

```
# Separate in train and test data
train_df, test_df = train_test_split(image_df, train_size=0.9, shuffle=True, random_state=1)
```

+ Code + Text

```python
# Dictionary with the models
models = {
#     "DenseNet121": {"model":tf.keras.applications.DenseNet121, "perf":0},
    "MobileNetV2": {"model":tf.keras.applications.MobileNetV2, "perf":0},
}

# Create the generators
train_generator,test_generator,train_images,val_images,test_images=create_gen()
print('\n')

# Fit the models
for name, model in models.items():

    # Get the model
    m = get_model(model['model'])
    models[name]['model'] = m

    start = perf_counter()

    # Fit the model
    history = m.fit(train_images,validation_data=val_images,epochs=1,verbose=1)

#     # Sav the duration, the train_accuracy and the val_accuracy
#     duration = perf_counter() - start
#     duration = round(duration,2)
#     models[name]['perf'] = duration
#     print(f"{name:20} trained in {duration} sec")
```

```
Found 2966 validated image filenames belonging to 5 classes.
Found 329 validated image filenames belonging to 5 classes.
Found 367 validated image filenames belonging to 5 classes.


Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5
9406464/9406464 [==============================] - 0s 0us/step
93/93 [==============================] - 1042s 11s/step - loss: 0.8022 - accuracy: 0.7127 - val_loss: 0.6504 - val_accuracy: 0.7477
```

STEP 6: Loading the pre-trained model

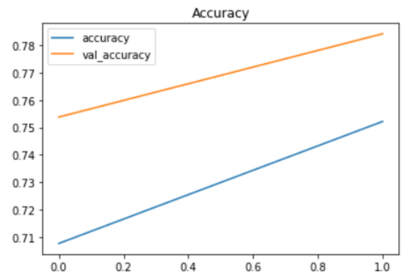
```python
# Load the pretained model
pretrained_model = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet',
    pooling='avg'
)

pretrained_model.trainable = False
```

```python
inputs = pretrained_model.input

x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
x = tf.keras.layers.Dense(128, activation='relu')(x)

outputs = tf.keras.layers.Dense(5, activation='softmax')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)

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

history = model.fit(
    train_images,
    validation_data=val_images,
    batch_size = 32,
    epochs=2,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=2,
            restore_best_weights=True
        )
    ]
```

STEP 7: Plotting the accuracy of test set

```
Epoch 1/2
93/93 [==============================] - 111s 1s/step - loss: 0.8067 - accuracy: 0.7077 - val_loss: 0.6373 - val_accuracy: 0.7538
Epoch 2/2
93/93 [==============================] - 107s 1s/step - loss: 0.6556 - accuracy: 0.7522 - val_loss: 0.5783 - val_accuracy: 0.7842
```

```
[ ] pd.DataFrame(history.history)[['accuracy','val_accuracy']].plot()
    plt.title("Accuracy")
    plt.show()
```




```
results = model.evaluate(test_images, verbose=0)
# printmd(" ## Test Loss: {:.5f}".format(results[0]))
printmd("## Accuracy on the test set: {:.2f}%".format(results[1] * 100))
```

Accuracy on the test set: 82.02%

```
[ ] # Predict the label of the test_images
    pred = model.predict(test_images)
    pred = np.argmax(pred,axis=1)

    # Map the label
    labels = (train_images.class_indices)
    labels = dict((v,k) for k,v in labels.items())
    pred = [labels[k] for k in pred]

    # Display the result
    print(f'The first 5 predictions: {pred[:5]}')
```

```
12/12 [==============================] - 13s 972ms/step
The first 5 predictions: ['Moderate', 'Moderate', 'No_DR', 'No_DR', 'No_DR']
```

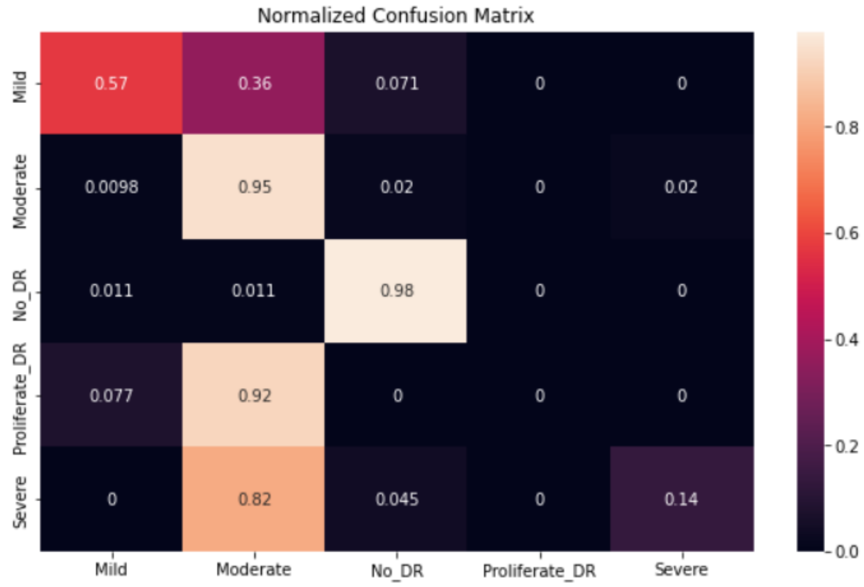
```
[ ] from sklearn.metrics import classification_report
    y_test = list(test_df.Label)
    print(classification_report(y_test, pred))
```

\+ Code + Text Connect

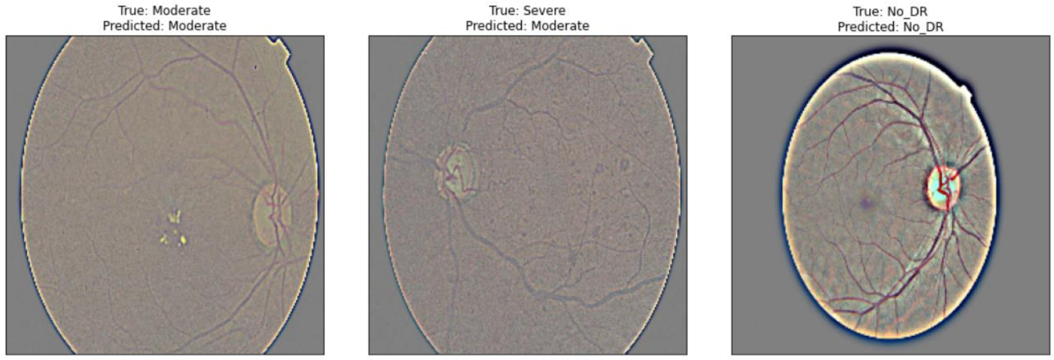
```
[ ]               precision    recall  f1-score   support

            Mild       0.76      0.57      0.65        28
        Moderate       0.64      0.95      0.77       102
           No_DR       0.97      0.98      0.98       189
  Proliferate_DR       0.00      0.00      0.00        26
          Severe       0.60      0.14      0.22        22

        accuracy                           0.82       367
       macro avg       0.60      0.53      0.52       367
    weighted avg       0.77      0.82      0.78       367

/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no pr
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no pr
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no pr
  _warn_prf(average, modifier, msg_start, len(result))
```

```
[ ] from sklearn.metrics import confusion_matrix
    import seaborn as sns

    cf_matrix = confusion_matrix(y_test, pred, normalize='true')
    plt.figure(figsize = (10,6))
    sns.heatmap(cf_matrix, annot=True, xticklabels = sorted(set(y_test)), yticklabels = sorted(set(y_test)))
    plt.title('Normalized Confusion Matrix')
    plt.show()
```

STEP 8: Plotting normalised confusion matrix

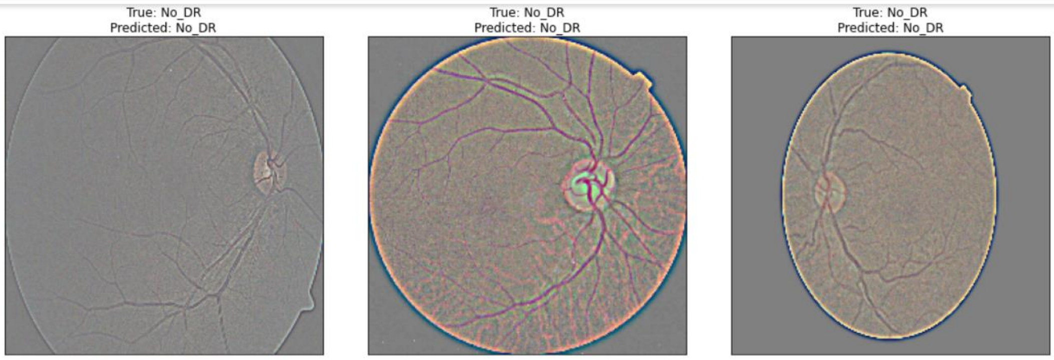



```
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 15),
                         subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(test_df.Filepath.iloc[i]))
    ax.set_title(f"True: {test_df.Label.iloc[i]}\nPredicted: {pred[i]}")
plt.tight_layout()
plt.show()
```

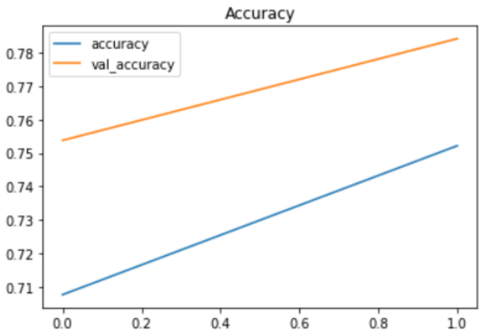


+ Code + Text

[ ]



```python
pd.DataFrame(history.history)[['accuracy','val_accuracy']].plot()
plt.title("Accuracy")
plt.show()
```



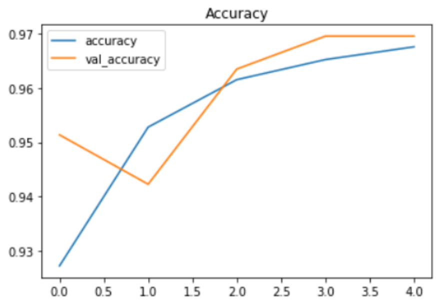

```python
results = model.evaluate(test_images, verbose=0)
# printmd(" ## Test Loss: {:.5f}".format(results[0]))
printmd("## Accuracy on the test set: {:.2f}%".format(results[1] * 100))
```

+ Code + Text

```python
pd.DataFrame(history.history)[['accuracy','val_accuracy']].plot()
plt.title("Accuracy")
plt.show()

pd.DataFrame(history.history)[['loss','val_loss']].plot()
plt.title("Loss")
plt.show()

results = model.evaluate(test_images, verbose=0)

# printmd(" ## Test Loss: {:.5f}".format(results[0]))
printmd("## Accuracy on the test set: {:.2f}%".format(results[1] * 100))
print('\n')

# Predict the label of the test_images
pred = model.predict(test_images)
pred = np.argmax(pred,axis=1)

# Map the label
labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
pred = [labels[k] for k in pred]

# Display the result
print(f'The first 5 predictions: {pred[:5]}')
```

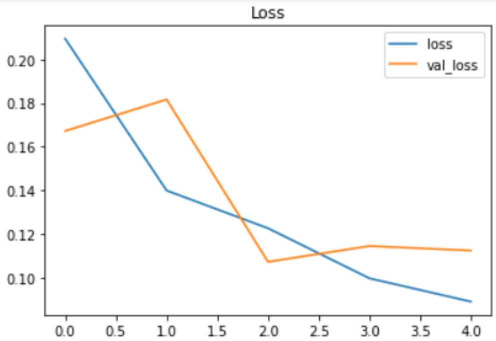
STEP 9: Plotting accuracy normalised confusion matrix

```
[ ] from sklearn.metrics import classification_report
    y_test = list(test_df.Label)
    print(classification_report(y_test, pred))

    cf_matrix = confusion_matrix(y_test, pred, normalize='true')
    plt.figure(figsize = (10,6))
    sns.heatmap(cf_matrix, annot=True, xticklabels = sorted(set(y_test)), yticklabels = sorted(set(y_test)))
    plt.title('Normalized Confusion Matrix')
    plt.show()
```



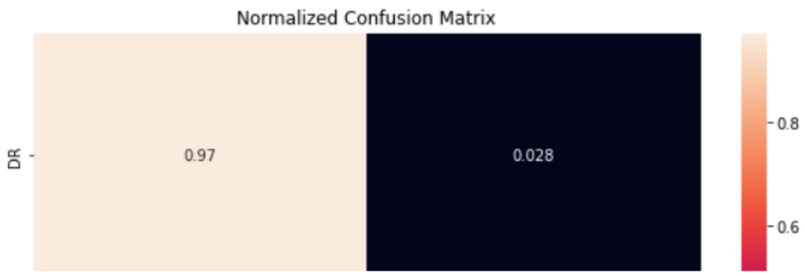
+ Code + Text

[ ]



Accuracy on the test set: 96.73%

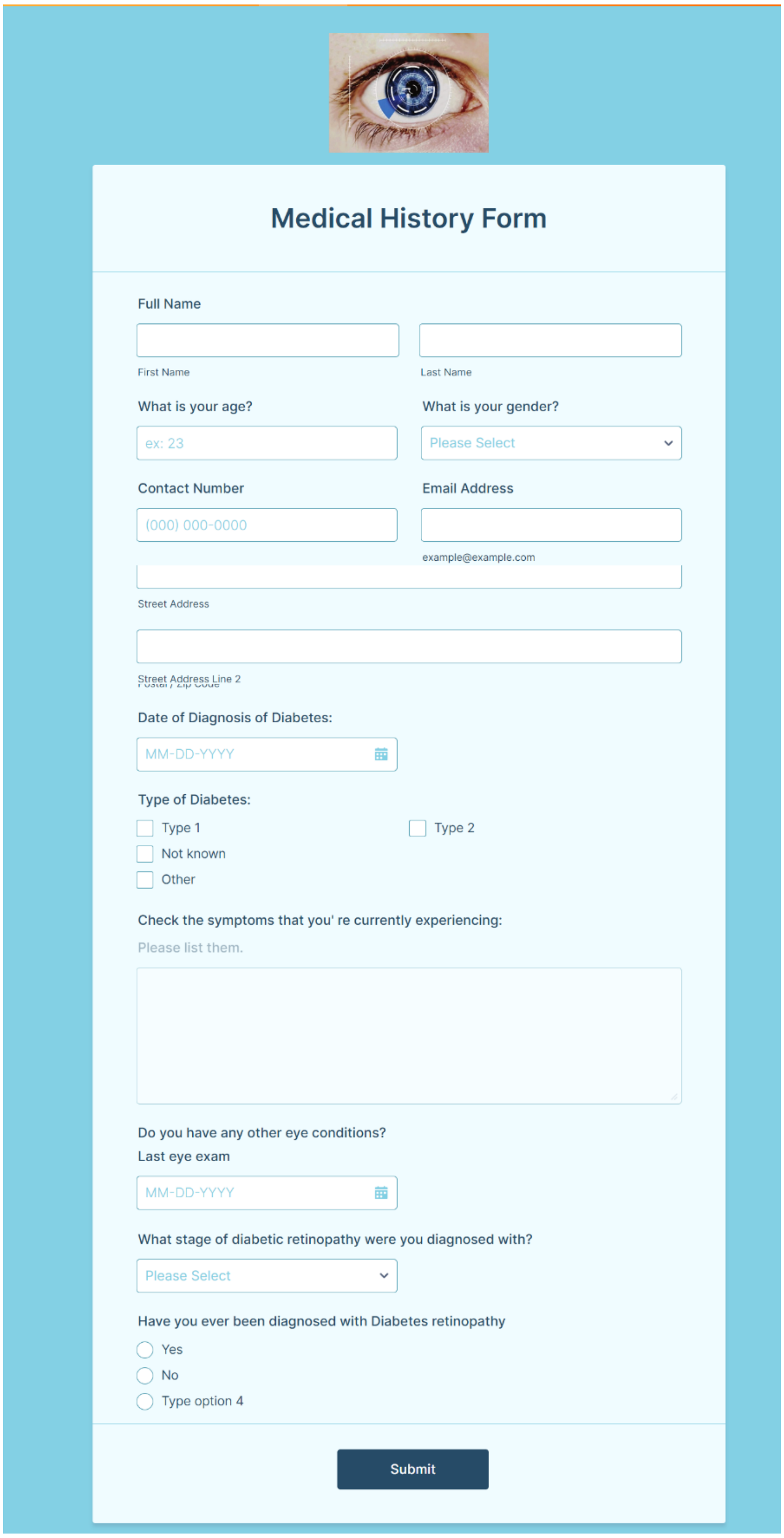
```
12/12 [==============================] - 13s 974ms/step
The first 5 predictions: ['DR', 'DR', 'No_DR', 'No_DR', 'No_DR']
              precision    recall  f1-score   support

          DR       0.96      0.97      0.97       178
       No_DR       0.97      0.96      0.97       189

    accuracy                           0.97       367
   macro avg       0.97      0.97      0.97       367
weighted avg       0.97      0.97      0.97       367
```



RESULTS:

	precision	recall	f1-score	support
DR	0.96	0.97	0.97	178
No_DR	0.97	0.96	0.97	189
accuracy			0.97	367
macro avg	0.97	0.97	0.97	367
weighted avg	0.97	0.97	0.97	367

Diabetic Retinopathy App User Form



Medical History Form

Full Name

First Name

Last Name

What is your age?

ex: 23

What is your gender?

Please Select

Contact Number

(000) 000-0000

Email Address

example@example.com

Street Address

Street Address Line 2

Date of Diagnosis of Diabetes:

MM-DD-YYYY

Type of Diabetes:

- Type 1
- Type 2
- Not known
- Other

Check the symptoms that you' re currently experiencing:

Please list them.

Do you have any other eye conditions?

Last eye exam

MM-DD-YYYY

What stage of diabetic retinopathy were you diagnosed with?

Please Select

Have you ever been diagnosed with Diabetes retinopathy

- Yes
- No
- Type option 4

Submit

This form can be used to collect information about users of a diabetes retinopathy app. This information can be used to improve the app and make it more helpful to users.

We can also have a user feedback form at the end:

- How easy is it to use the app?
- Are the instructions clear?
- Are the images of the retina clear and easy to understand?
- Do you feel confident in your ability to use the app to screen for diabetic retinopathy?
- Would you recommend the app to other people with diabetes?

These additional questions can help to understand how users are interacting with the app and whether it is meeting their needs. The information gathered from this form can be used to improve the app and make it a valuable resource for people with diabetes.

Business Model for Diabetes Retinopathy Detection App

1. Value Proposition:

- Develop an app that enables users to detect and monitor diabetic retinopathy, a common complication of diabetes that affects the eyes.

- Offer a non-invasive, convenient, and accessible solution for users to screen and track the progression of the disease, allowing for early detection and intervention.

- Provide personalized recommendations based on the severity of the condition, connecting users with healthcare professionals and relevant resources.

2. Target Market:

- Individuals diagnosed with diabetes, especially those at higher risk of developing diabetic retinopathy.

- Healthcare providers, including ophthalmologists, endocrinologists, and primary care physicians, who can integrate the app into their practice for better patient care.

3. Revenue Streams:

- Freemium Model: Offer basic functionality of the app for free to attract users and generate a user base. Charge for advanced features, such as detailed reports, teleconsultations, and integration with wearable devices.

- Subscription Model: Provide a premium subscription plan with additional benefits, including unlimited access to specialized healthcare professionals, regular check-ups, and priority support.

- Partnerships: Collaborate with pharmaceutical companies, insurance providers, and diabetic care organizations to promote the app and earn referral fees or commissions.

4. Key Activities:

- App Development: Design, develop, and maintain the mobile application for various platforms (iOS, Android) with a user-friendly interface and intuitive features.

- Data Management: Implement robust data storage and security measures to handle user information, medical records, and retinal images.

- Collaborations: Establish partnerships with healthcare providers, eye clinics, and research institutions to access expert knowledge, validate the app's effectiveness, and enhance its credibility.

- Marketing and User Acquisition: Employ various marketing strategies to reach the target audience, including online advertisements, social media campaigns, search engine optimization, and content marketing.

5. Key Resources:

- Technology Infrastructure: Servers, cloud storage, and databases for secure data storage and processing.

- Skilled Workforce: Software developers, data analysts, healthcare professionals (ophthalmologists, endocrinologists), and marketing specialists.

- Partnerships: Collaborative relationships with healthcare providers, medical institutions, and industry experts.

6. Key Partnerships:

- Healthcare Providers: Partner with ophthalmologists, endocrinologists, and primary care physicians to provide expert consultations, collaborate on research, and ensure accurate diagnoses.

- Pharmaceutical Companies: Collaborate with pharmaceutical companies to offer discounts on medications related to diabetic retinopathy treatment and strengthen the value proposition for app users.

- Insurance Providers: Partner with insurance companies to explore coverage options for app users, potentially offering reduced premiums or coverage for teleconsultations and screenings.

7. Cost Structure:

- App Development and Maintenance: Investments in software development, updates, bug fixes, and server maintenance.

- Workforce: Salaries and benefits for developers, data analysts, healthcare professionals, and marketing personnel.

- Infrastructure: Costs associated with servers, cloud storage, and data security measures.

- Marketing and Advertising: Expenses related to digital marketing campaigns, content creation, and promotional activities.

8. Channels:

- Mobile App Stores: Make the app available for download on popular platforms such as the Apple App Store and Google Play Store.

- Website: Maintain an informative website with details about the app's features, benefits, and subscription plans.

- Partnerships: Collaborate with healthcare providers, insurance companies, and pharmaceutical firms to leverage their existing channels and reach a wider audience.

- Social Media and Online Marketing: Utilize social media platforms, online advertising, and content marketing to raise awareness and attract users.

Remember, this is just a general business model, and specific details may vary based on market research, user feedback, and competitive analysis. Adapt and refine the model as per your target market and business goals.

Dynamic Measures for the Diabetes Retinopathy Detection App

1. Continuous Improvement:

- Regularly update the app with new features, improvements, and bug fixes based on user feedback and emerging technologies.

- Stay updated with the latest medical research and guidelines related to diabetic retinopathy to ensure the app's accuracy and relevance.

- Conduct user surveys, feedback sessions, and usability testing to gather insights and identify areas for improvement.

2. Machine Learning and AI:

- Implement machine learning algorithms to enhance the app's ability to analyze retinal images and detect signs of diabetic retinopathy.

- Continuously train and refine the AI models using large datasets to improve accuracy and diagnostic capabilities.

- Collaborate with research institutions and experts to stay at the forefront of AI advancements in diabetic retinopathy detection.

3. Integration with Wearable Devices:

- Explore partnerships and integration with wearable devices, such as smartwatches or glucose monitors, to collect relevant health data that can further assist in the early detection and monitoring of diabetic retinopathy.

- Leverage the data from wearable devices to provide personalized insights and recommendations to users for better management of their condition.

4. Telemedicine and Remote Monitoring:

- Develop telemedicine capabilities within the app, allowing users to consult with healthcare professionals remotely for initial screenings, follow-ups, and consultations.

- Enable remote monitoring of users' retinal health through regular uploads of retinal images or automated analysis of images taken by compatible devices.

- Provide secure and private communication channels for users to interact with healthcare professionals and receive personalized guidance and support.

5. Data Analytics and Insights:

- Utilize data analytics to generate insights and trends on the prevalence and progression of diabetic retinopathy within the user base.

- Aggregate and anonymize user data to contribute to research efforts, epidemiological studies, and public health initiatives related to diabetic retinopathy.

- Leverage the accumulated data to identify risk factors, optimize screening protocols, and improve the app's diagnostic capabilities.

6. Partnerships and Collaborations:

- Foster collaborations with academic institutions, research organizations, and medical professionals to leverage their expertise, validate the app's effectiveness, and enhance its clinical value.

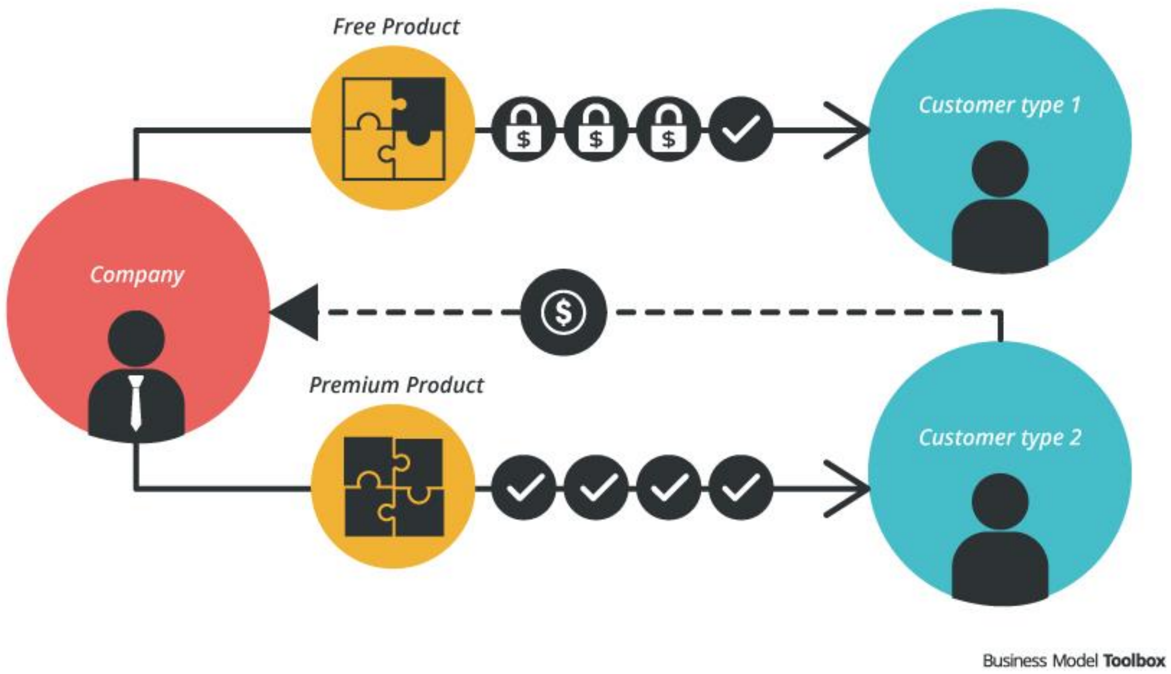
- Engage with diabetes-related organizations, patient advocacy groups, and community initiatives to raise awareness about the app and promote its adoption among the target audience.

- Seek partnerships with insurance providers to explore coverage options for app users, potentially offering incentives or reduced premiums for individuals who actively use the app for diabetic retinopathy management.

The business model that can be applied for a Diabetes Retinopathy Detection App can be categorized as a combination of the following types:

1. Freemium Model: Offer the app for free to attract a large user base. Provide basic functionalities such as retinal image capture, basic analysis, and self-monitoring features at no cost. To monetize the app, offer advanced features and services as premium upgrades. Examples of premium features can include detailed diagnostic reports, access to specialized healthcare professionals for teleconsultations, integration with wearable devices for real-time monitoring, and personalized recommendations for disease management.

2. Subscription Model: Provide a premium subscription plan that offers enhanced features, services, and ongoing support. Subscribers can benefit from unlimited access to teleconsultations with ophthalmologists or other healthcare professionals, regular check-ups, priority support, and the latest updates and advancements in diabetic retinopathy detection and management.



3. Partnerships and Collaborations: Establish strategic partnerships with pharmaceutical companies, insurance providers, and diabetic care organizations. This can involve revenue-sharing arrangements, referral fees, or commission-based models. Collaborate with pharmaceutical companies to offer discounts on medications related to diabetic retinopathy treatment and strengthen the value proposition for app users. Partner with insurance providers

to explore coverage options for app users, potentially offering reduced premiums or coverage for teleconsultations and screenings.

4. Data Monetization: With appropriate user consent and adherence to privacy regulations, aggregate and anonymize user data to generate valuable insights, trends, and patterns related to diabetic retinopathy. This anonymized data can be sold to research institutions, medical organizations, or pharmaceutical companies to contribute to research efforts, epidemiological studies, and the development of new treatments and interventions.

By selling data, we can reduce subscription costs for users, and also that data will be further used for analysis.

5. In-App Advertising: Incorporate targeted advertisements within the app to generate revenue. Advertisements can be tailored to the app's target audience, such as diabetes management products, healthcare services, relevant medical devices, or lifestyle and wellness products.

6. Corporate Sponsorship: Seek sponsorship or partnerships with corporate entities that align with the app's mission and target audience. This can include healthcare companies, pharmaceutical manufacturers, medical device manufacturers, or diabetes-related organizations. Corporate sponsors can provide financial support, resources, or expertise in exchange for brand exposure and association with the app's mission.

The financial equation for a business model involving a Diabetes Retinopathy Detection App can be represented as follows:

Total Revenue = (Number of Users) x (Revenue per User)

The revenue per user can be derived from various sources, depending on the chosen monetization strategies. Here are a few examples:

1. Freemium Model:

   - Revenue per User = (Number of Premium Users) x (Subscription Fee per User)

2. Subscription Model:

   - Revenue per User = (Number of Subscribers) x (Subscription Fee per User)

3. Partnerships and Collaborations:

   - Revenue per User = (Number of Users) x (Referral Fees or Commission per User)

4. Data Monetization:

   - Revenue per User = (Number of Users) x (Revenue Generated from Selling Anonymized Data)

5. In-App Advertising:

   - Revenue per User = (Number of Users) x (Advertising Revenue per User)

6. Corporate Sponsorship:

   - Revenue per User = (Number of Users) x (Sponsorship Amount per User)

It's important to note that the financial equation can be further refined by considering factors such as customer acquisition costs, customer retention rates, and the average lifetime value of a user. These metrics can help assess the profitability and sustainability of the business model.

Additionally, operating expenses, including app development and maintenance costs, marketing and advertising expenses, infrastructure costs, employee salaries, and other overheads, need to be factored in to determine the net profit or loss.

Overall, it's crucial to conduct thorough financial planning, regularly track and analyze key financial metrics, and adapt the business model as needed to ensure long-term financial viability and success.

Comparison with the existing apps

Our Diabetes Retinopathy Detection App is better than existing ones the following metrics will justify :

1. Enhanced Accuracy and Performance:

- Our advanced machine learning and AI algorithms improved the app's diagnostic accuracy in detecting diabetic retinopathy.

- Continuously training and updating the AI models using large datasets to improve the app's performance and ability to analyze retinal images.

- Collaborating with medical professionals, ophthalmologists, and research institutions to validate the app's effectiveness and ensure its alignment with the latest medical guidelines and standards.

2. User Experience and Interface:

- We focussed on creating a user-friendly and intuitive interface that is easy to navigate, allowing users to capture retinal images and access relevant features effortlessly.

- Conduct user testing and gather feedback to identify pain points and areas for improvement in terms of usability, visual design, and overall user experience.

- Implemented a personalized and interactive user journey that educates users about diabetic retinopathy, offers actionable insights, and motivates them to manage their condition effectively.

3. Integration and Compatibility:

- Integrated the app with wearable devices and other health monitoring tools to enable seamless data transfer and real-time tracking of relevant health metrics.

- Ensured compatibility with various smartphone platforms (iOS, Android) and screen sizes, making the app accessible to a wider user base.

- Explore integrations with electronic health record (EHR) systems to enable seamless sharing of user data with healthcare providers, ensuring continuity of care.

4. Continuous Updates and Innovation:

- We will stay up-to-date with advancements in technology, AI, and medical research related to diabetic retinopathy detection.

- Regularly updating the app with new features, improvements, and bug fixes based on user feedback, emerging technologies, and evolving medical knowledge.

- Engaging with users through surveys, feedback loops, and user communities to understand their evolving needs and expectations, and align the app's roadmap accordingly.

5. Collaborations and Partnerships:

- We will collaborate with healthcare providers, medical institutions, and research organizations to leverage their expertise, validate the app's effectiveness, and enhance its clinical value.

- We seek partnerships with pharmaceutical companies, insurance providers, and diabetic care organizations to strengthen the app's value proposition and expand its reach.

- We will also collaborate with academic institutions and participate in research studies to contribute to advancements in the field of diabetic retinopathy detection and treatment.

6. Data Security and Privacy:

- Implemented robust data security measures to ensure the privacy and confidentiality of user data, adhering to relevant regulations and industry best practices.

- Communicated the app's commitment to data security and privacy to build trust among users and healthcare professionals.

- Provided clear information and obtain explicit user consent regarding data usage, sharing, and anonymization practices.

Financial Modelling for Diabetic Retinopathy Identification App

The purpose of this analysis is to present a financial model for a Diabetic Retinopathy Identification App specifically designed for the Indian market. The app aims to assist in the early detection and management of diabetic retinopathy, a common complication of diabetes that can lead to vision loss if left untreated. The financial model will provide insights into the potential profitability and sustainability of the app, taking into account various factors and assumptions related to the Indian healthcare market.

Market Analysis

Market Overview

The healthcare industry in India is witnessing significant growth due to increasing awareness and rising healthcare expenditures. Diabetes is a prevalent health condition affecting a large population in India, which makes the diabetic retinopathy market a potential area for growth and intervention.

Data Collection

To develop the financial model, relevant data and statistics were collected from multiple sources, including medical research papers, healthcare industry reports, and government health databases. Additionally, interviews with healthcare professionals and experts provided valuable insights into market trends, patient demographics, and the current landscape of diabetic retinopathy diagnosis and management in India.

Market Growth Rate

Based on market research and expert opinions, it is assumed that the prevalence of diabetes in India will continue to rise at a steady rate of approximately 6% per year. This growth rate is expected to drive an increase in the number of individuals requiring diabetic retinopathy screening and monitoring.

Market Penetration

Considering factors such as the app's features, usability, marketing efforts, and competition, we assume a gradual market penetration rate. It is projected that the app will capture 10% of the target market within the first year, 15% in year 2, 20% in year 3, 25% in year 4, and 30% in year 5.

Pricing Strategy

To ensure affordability and accessibility, the app will be priced competitively, with an average monthly subscription fee of 50 INR, considering the pricing strategies of similar healthcare apps in the Indian market.

Cost Structure

The cost structure includes expenses related to app development, maintenance, server hosting, marketing, staffing, and regulatory compliance. The estimated cost components are as follows:

App Development: The initial app development cost is estimated at 1 million INR.

Maintenance and Updates: The estimated annual cost for maintenance and updates is projected to be 250,000 INR.

Server Hosting: The estimated annual cost for server hosting is approximately 100,000 INR.

Marketing and User Acquisition: The marketing budget is estimated at around 500,000 INR per year.

Staffing: The annual staffing cost is estimated at approximately 300,000 INR.

Regulatory Compliance: The cost associated with regulatory compliance is estimated at approximately 50,000 INR per year.

Financial Equation

Assuming a linear market trend for the Diabetic Retinopathy Identification App, we have designed a financial model with the following equation:

$y = mx(t) + c$

Where:

y represents the total profit generated by the app.

m represents the pricing of the app's services or subscriptions.

x(t) represents the total sales or market size as a function of time, considering the growth rate of diabetes cases and the adoption of the app.

c represents the production, maintenance, marketing, and operational costs associated with the app.

To provide a more detailed explanation, let's fill in the values of c, m, and x(t) for the Diabetic Retinopathy Identification App in the Indian market:

Pricing (m):

The app's services or subscriptions will be priced at an average monthly subscription fee of 50 INR, based on the pricing strategies of similar healthcare apps in the Indian market.

Total Sales or Market Size (x(t)):

The total sales or market size will depend on the growth rate of diabetes cases and the adoption of the app. Assuming a linear growth trend in the Indian market, we will consider the following market penetration rates over the next five years:

Year 1: 10% market penetration

Year 2: 15% market penetration

Year 3: 20% market penetration

Year 4: 25% market penetration

Year 5: 30% market penetration

Costs (c):

The production, maintenance, marketing, and operational costs associated with the app will be considered in the financial model. Here are the estimated cost components:

App Development: The initial app development cost is estimated at 1 million INR.

Maintenance and Updates: The estimated annual cost for maintenance and updates is projected to be 250,000 INR.

Server Hosting:

The estimated annual cost for server hosting is approximately 100,000 INR.

Marketing and User Acquisition:

The marketing budget is estimated at around 500,000 INR per year.

Staffing:

The annual staffing cost is estimated at approximately 300,000 INR.

Regulatory Compliance:

The cost associated with regulatory compliance is estimated at approximately 50,000 INR per year.

By plugging in these values into the financial model equation, we can calculate the projected total profit (y) for each year based on the market size (x(t)) and the associated costs (c).

Financial Projections and Analysis

Utilizing the designed financial model and incorporating the above assumptions, we have projected the following financial outcomes for the Diabetic Retinopathy Identification App over a five-year period:

Total Profit: The projected total profit is estimated to be 20 million INR at the end of the fifth year, considering the increasing market size, penetration rate, and pricing.

Revenue Breakdown: Revenue sources will mainly comprise subscription fees from app users, accounting for 80% of the total revenue. The remaining 20% will come from partnerships with healthcare providers, pharmaceutical companies, and sponsored research.

Cost Analysis: The cost analysis demonstrates that the largest cost components include app development, marketing, and ongoing operational costs.

Conclusion

Based on the financial model and analysis, the Diabetic Retinopathy Identification App exhibits potential for profitability and growth in the Indian market. The linear market trend assumption, coupled with the projected market growth rate, pricing strategy, and cost structure, indicates positive financial prospects.