HUBBLE
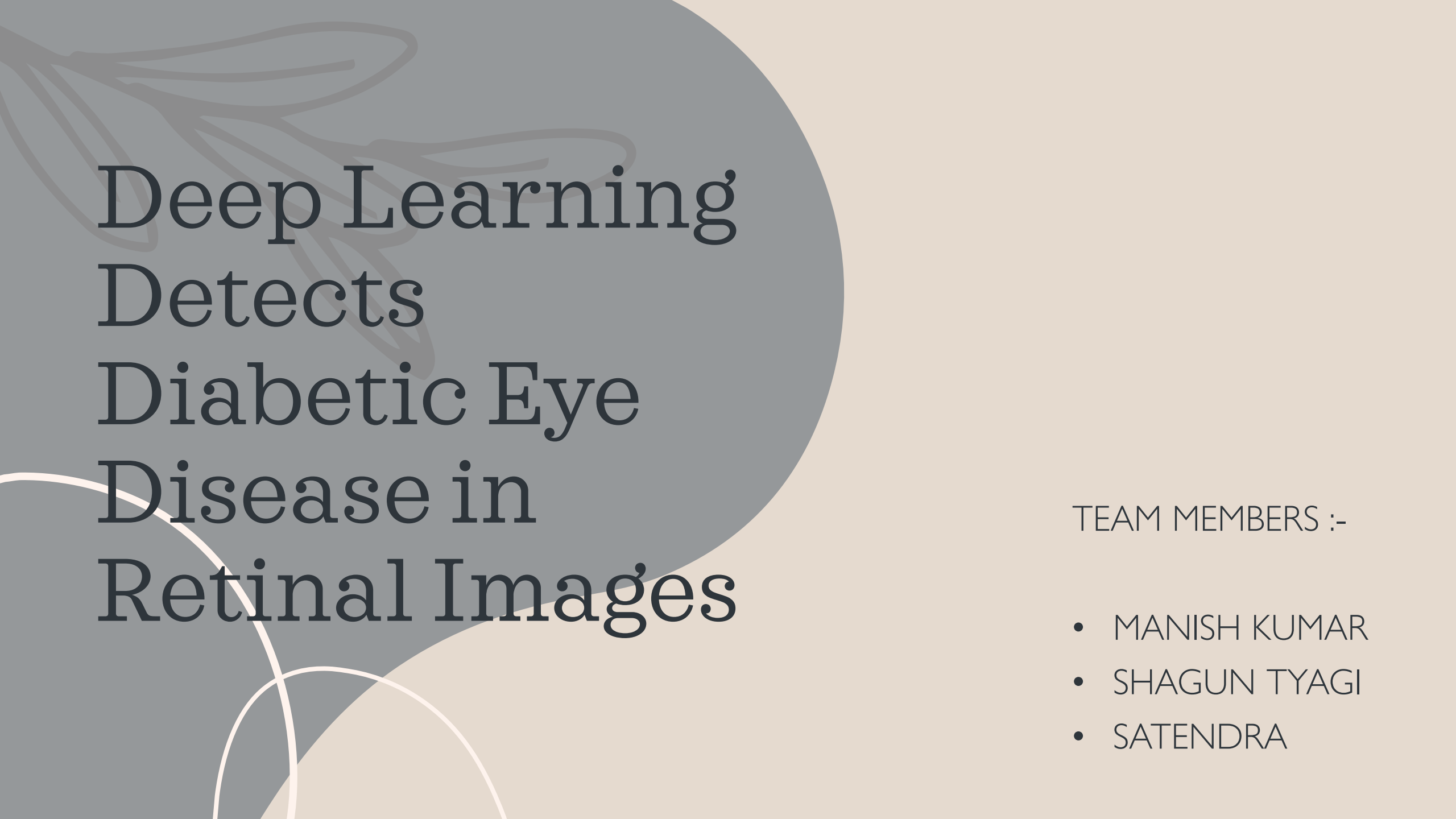
Deep Learning Detects Diabetic Eye Disease in Retinal Images

TEAM MEMBERS :-

- MANISH KUMAR
- SHAGUN TYAGI
- SATENDRA

INDEX

INDEX

INTRODUCTION

Brief overview of the project:

This project aims to use deep learning techniques to detect diabetic eye disease in retinal images. The project consists of a web application with a Python backend that hosts a pre-trained deep learning model for predicting the presence of diabetic eye disease, and a React.js frontend for user interaction.

Goals and Objectives: The goal of the project is to provide a user-friendly, accurate and accessible way to detect diabetic eye disease.
The objectives include developing a deep learning model that achieves high accuracy in detecting diabetic eye disease, implementing a user-friendly web interface for users to upload retinal images for analysis, and ensuring that the application is easy to use and accessible to users with varying levels of technical expertise.

Problem statement: Diabetic eye disease is a major cause of blindness and vision loss in people with diabetes. Early detection and treatment can significantly reduce the risk of blindness, but diagnosis can be difficult without specialized training and equipment. This project aims to provide a simple, accessible and accurate way to detect diabetic eye disease, allowing for earlier diagnosis and better outcomes for patients.

ARCHITECTURE

ARCHITECTURAL EXPLANATION

Explanation of each component and how they work together:

The frontend, built using React.js and @mui/material, provides a user-friendly interface for users to upload retinal images for analysis. The Python backend, built using Flask, hosts a pre-trained deep learning model for predicting the presence of diabetic eye disease in retinal images. The backend receives the image data from the frontend, processes it using the deep learning model, and returns the prediction result to the frontend. User authentication is handled by a Node.js backend that uses JSON Web Tokens (JWTs) for secure authentication.

MongoDB is used to store user data and authentication tokens. Mention the use of Flask for the Python backend, Node.js for user authentication, and MongoDB for the database: Flask is a lightweight web framework for Python that is well-suited for building RESTful APIs. Node.js is a popular JavaScript runtime that is used for building scalable and high-performance web applications. MongoDB is a NoSQL document database that provides scalability, flexibility, and performance. Together, these technologies provide a powerful and flexible platform for building web applications with deep learning capabilities.

primary
goals

summary

At Contoso, we believe in giving 110%. By using our next-generation data architecture, we help organizations virtually manage agile workflows. We thrive because of our market knowledge and great team behind our product. As our CEO says, "Efficiencies will come from proactively transforming how we do business."

Thank You