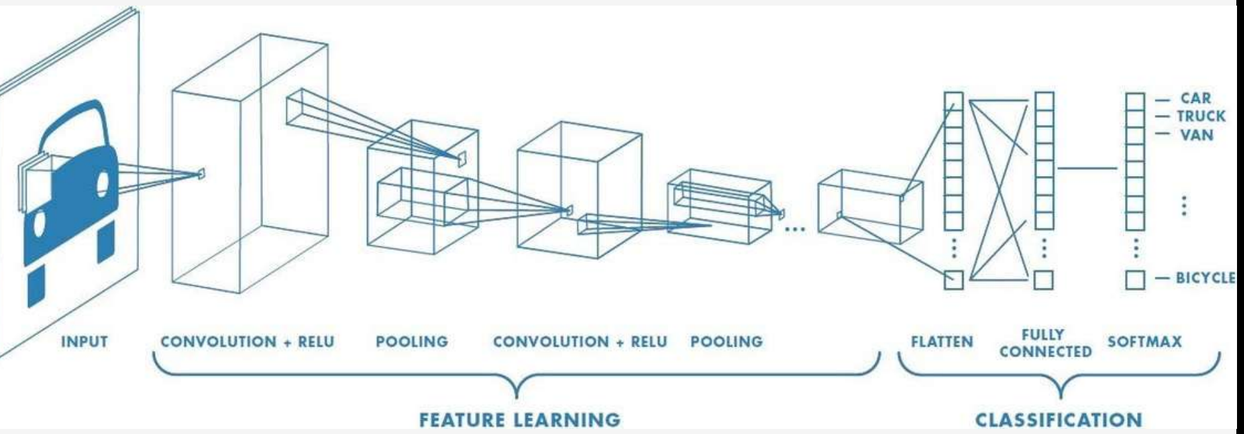
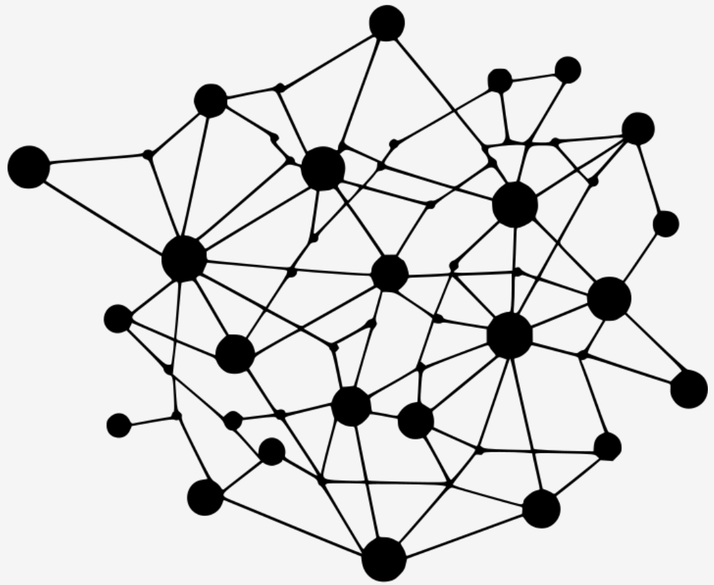
PerceptiveAnalytics

Release 1 : Team 33

COMPUTER VISION APPLICATION FOR REAL-TIME MULTI-MODAL PRODUCT DETECTION

Chetan Mahipal, Rohan Shridhar, Badarla Rohan Naidu,
Aanvik Bhatnagar, Rohan Rathee





INTRODUCTION TO PRODUCT DETECTION

- In today's fast-paced world, efficient product detection holds immense significance across diverse industries.
- Leveraging computer vision (CV) algorithms, particularly those empowered by deep neural networks, offers a promising solution to this challenge.
- This multimodal product detection algorithm, developed by our client accurately identifies products from both static images and live video feeds.

Mission STATEMENT

- The primary objective is to make these powerful CV capabilities widely accessible through a user-friendly mobile platform.
- The application will support two key functionalities: still image recognition and real-time video analysis.
- Users will be able to capture a photograph or stream video directly through the app, which will then communicate with cloud-based computing resources to process the data using existing CV algorithms.
- The processed information will be swiftly relayed back to the mobile application, ensuring users receive low latency, real-time results.

Target Market

Who are the customers you want to cater to?

Clients of Perceptive Analytics

An ordinary person who is interested in basic functionalities, and not much technical background.

Retail Stores

Admin of stores, with a little knowledge of software but are handy in using applications on mobile devices.

Inventory Management

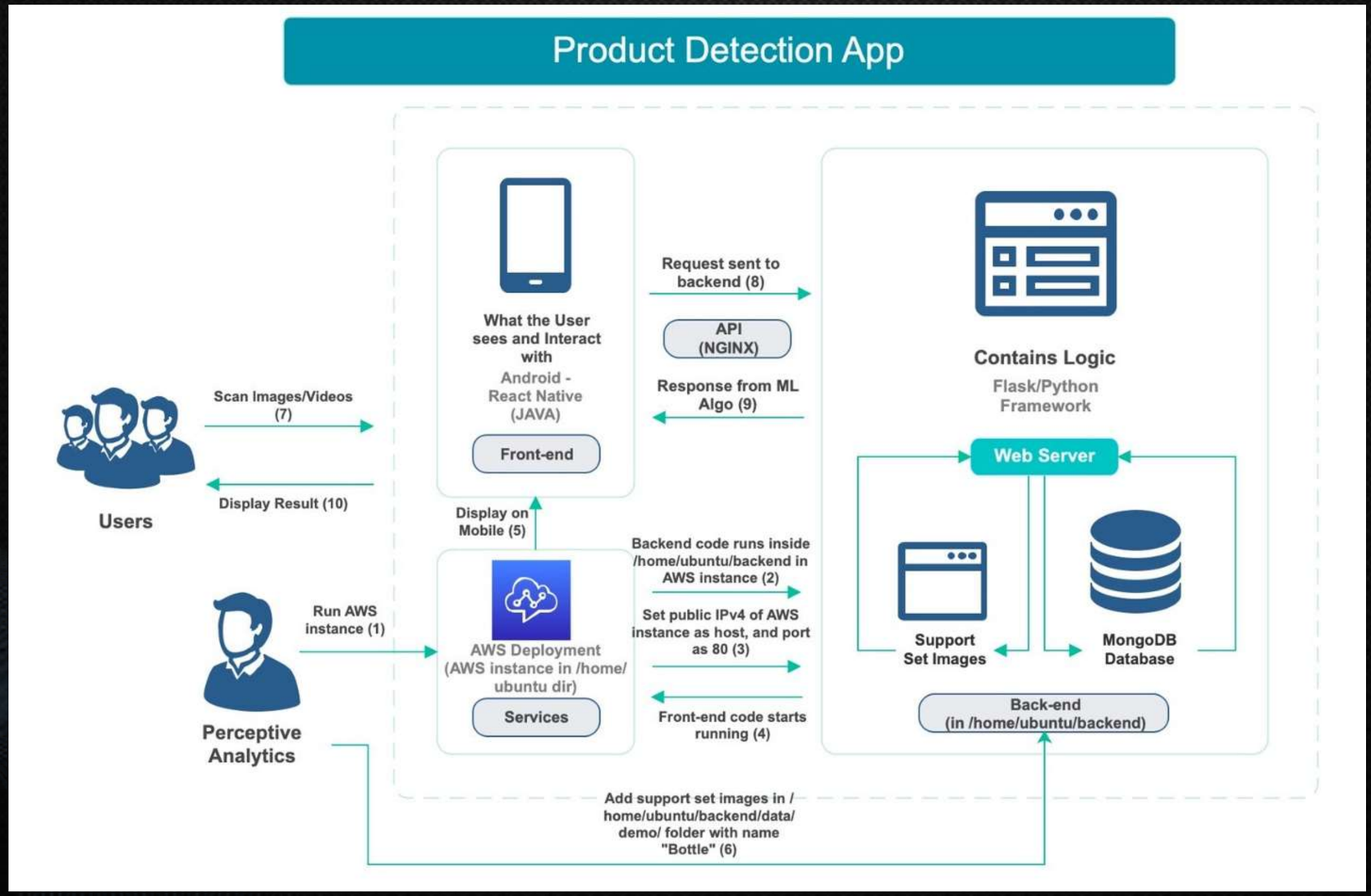
Helps in keeping the stock updates and need for
restocking when required.

Delivery System Companies

Officials who keep a check at the quantity delivered by delivery guy, who is handy with mobile apps.

Current App Architecture

How the currrent implementation of app deployed on AWS looks like



Current App Issues

1

Lacks user friendly interface

The current implementation of the app is not visually appealing, and is not self-explanatory. We need a responsive and seamless experience.

2

Technical complexities

The codebase includes some errors, like deprecated frameworks, image incompatibility and other generic bugs.

3

Cloud Service Connectivity

For the scalability of the app to a larger level, we need to look into all the deployment options and provide them a mechanism.

Deliverables

1 Responsive UI

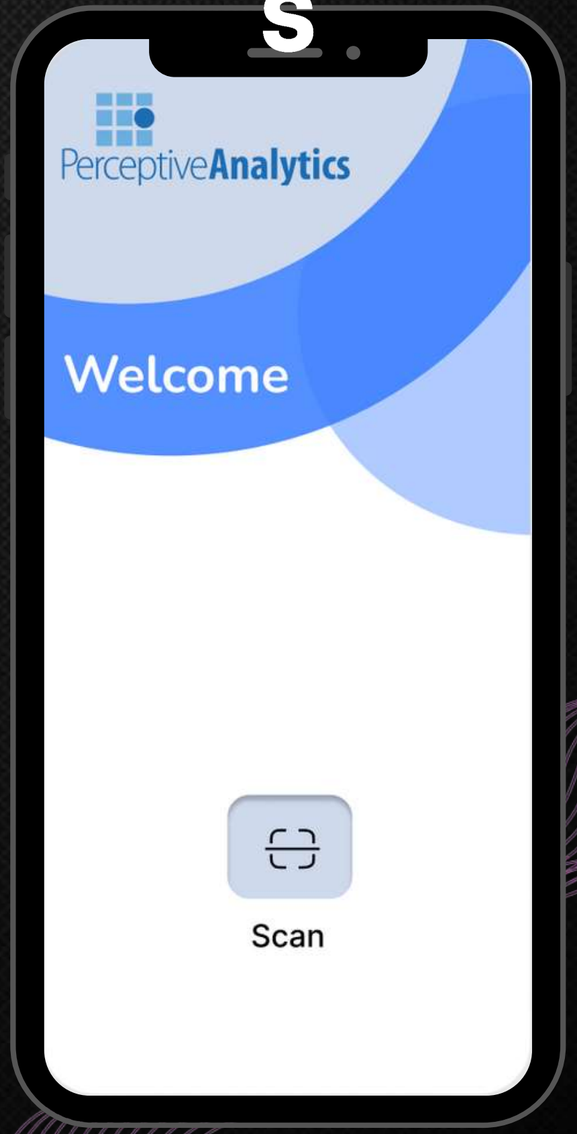
We aimed to improve the user interface by better design layout, and easily understandable procedural steps.

2 Robust App Architecture

Based on the current framework(s) of the app that the codebase is using, deprecations are removed, and updated compatible versions.

3 Cloud Deployment Options

Leveraging cloud computing, our solution aims to provide swift and accurate results directly to users' mobile devices.



PHASE PLAN

Phase 1

Frontend:

- Home Page UI
- Camera Page UI

Backend:

- Image Processing

Phase 2

Frontend:

- Items Page UI
- Detected Frames UI

Backend:

- Image Viewer
- Items Page Backend

Release 1

Phase 3

Backend:

- Video Viewer
- AWS Deployment
- Video Analysis

Documentation

Improvements before Final Release

Technology REQUIREMENTS

Based on the current implementation framework and codebase of the app, and upon reading the available documentation, we have divided the requirements into three broad categories. For the detailed system requirements please refer to SRS.

React Native + NodeJS (Frontend)

- Support for third party tools
- Fast and Responsive across cross platforms
- Extensive Documentation and Tutorials
- Good for lightweight application

Python3 + Flask (Backend)

- More control over app design
- Easily scalable for smaller applications
- Python Libraries for AI-ML Support
- Flexibility on Structuring

ExpoGo (Phone Rendering)

- Runs seamlessly on both iOS and Android platforms without the need for separate codebases.
- Supports hot reloading, allowing developers to see changes in real-time as they modify code.
- Provides access to native device features and APIs through its extensive library of pre-built modules.

Product FEATURES

01

Home Page

This is the first page that will open when the app is run. There is a scan button which redirects the user to camera page where they can scan any object.

02

Camera Page

Here the user has three options: clicking a picture, taking a video, or uploading image/video from gallery.

03

Items Page

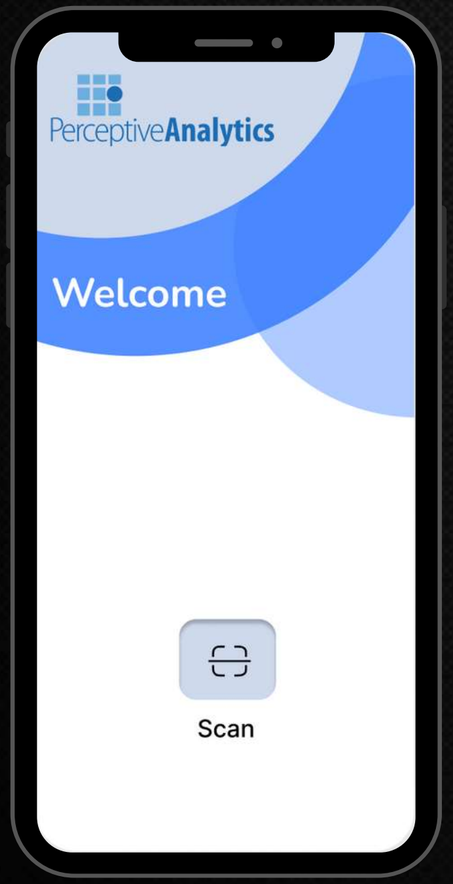
After the image/video is sent to scan from Camera Page, then the detected items are displayed in a table, where quantity can be modified, and image can be viewed.

04

Detected Image Frames

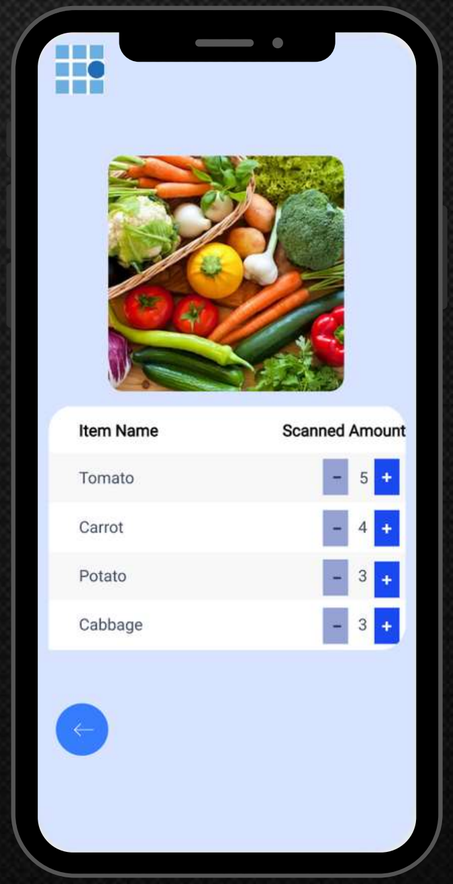
On the items page, if the user wants to see the frames that have been detected from CV Algorithm, then by clicking on image, they can see all the frames.

Home Page



Camera Page

Items Page



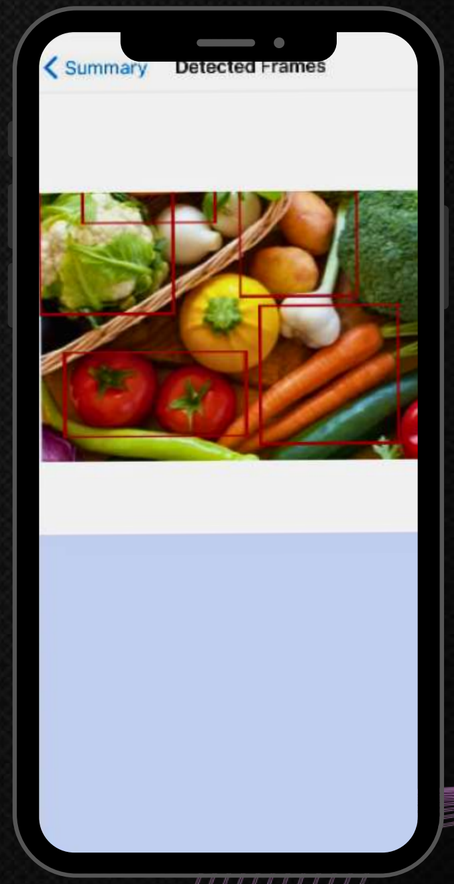


Image Frames

Challenges Faced Before Release 1

Deprecated Frameworks

The Expo Go updated it's version with compatibility issues with react native and it's libraries.

Fix: Created a Bash Script that would make the package.json with compatible versions

base64 Image Processing

Detected image in base64 format was not getting displayed on Items Page

Fix: base64 image needed to be decoded in the codebase

Node JS version

The latest version of Node is 21.7.1 but for the functioning of app, version 16 is required

Fix: Used 'npm' for version control from 21 to 16 as default

Digital Ocean Deployment

Client asked us to check whether Digital Ocean VM can be used for deployment, however MongoDB was not getting installed there. Asked for AWS in R2.

Future Roadmap

Our plan after Release 1 (also mentioned in Project Plan v2)

Release 1 Documentation

19TH MARCH 2024

Video Processing

2ND APRIL 2024

AWS Deployment

4TH APRIL 2024

Real Time Testing (Video+AWS)

10TH APRIL 2024

The Team

Aanvik Bh

BACKEND TEAM

- Image Processing
- SRS Document
- Status Tracker
- Cloud Deployment
- Cost Analysis
- R1 Presentation

Rohan S

FRONTEND TEAM

- Home Page
- SRS Document
- Figma Actions and UI

Rohan N

FRONTEND TEAM

- Home Page
- Camera Page
- Design Document
- Item’s List
- SRS Document

Chetan M

BACKEND TEAM

- Deprecation Removal
- Project Plan and Synopsis
- Image Viewer & Processing
- Items Page Backend
- Test Plan Tracker
- App Architecture

Rohan Sh

FRONTEND TEAM

- Item’s Page and Viewer
- Design Document
- Figma Actions and UI
- Upload Functionality
- Camera Page

Thank You