**Project Report**

| | |
|---|---|
| Date | 22 Nov 2023 |
| Project Name | Vitamin Detection using Deep learning |
| Team ID | Team-592065 |

We have provided a detailed analysis and information regarding our project in the below sections.

**Index**


1. Introduction
   1.1 Project Overview
   1.2 Purpose
2. Literature Survey
   2.1 Existing Problem
   2.2 References
   2.3 Problem statement Definition
3. Ideation and proposed solution
   3.1 Empathy Map Canvas
   3.2 Ideation and Brainstorming
   3.3 Proposed Solution
4. Requirement Analysis
   4.1 Functional Requirement
   4.2 Non-Functional Requirement
5. Project Design
   5.1 Data Flow Diagram
   5.2 Solution and Technical Architecture
   5.3 User Stories
6. Project Planning and scheduling
   6.1 Sprint Planning and Estimation
   6.2 Sprint Delivery Schedule
   6.3 Reports from JIRA
7. Coding
8. Testing
9. Results
   9.1 Performance Metrics
10. Advantages and disadvantages
11. Conclusion
12. Future scope

**Introduction:**

**Project Overview:**

Our project is a creative endeavour to create a thorough and reliable predictive model that can evaluate the vitamin present in the food item that is provided to the model. The project uses deep learning model that is convolutional neural networks since the dataset contains images of different food items containing different vitamins. The project aims at reducing the time in predicting the vitamins present in the food item. The result is provided faster than the usual time taken by the traditional methods. This can be useful for organizations in performing tasks such as nutritional analysis, meal planning and other tasks.

The project involves gathering and organizing data from multiple sources, applying image preprocessing techniques, utilizing predictive model to build a strong and precise model, deploying the model, and continuously observing and enhancing the performance if the model. If the project successfully gets implemented, data scientist, researchers, and many others will be benefited from the model. This will in return help in improved decision making, increase in availability and can be used by many users.

**Purpose:**

The purpose of this project is that it helps in reducing the manual efforts and the process is easily carried by the deep learning model. It helps in nutritional analysis and users can gain insights into the vitamins present in their food, aiding in making informed dietary choices for better health. Offers a convenient and efficient method for users to determine the vitamin present in their meals. It acts as an educational tool in various ways to the users. It helps in monitoring the health of users by allowing them to analyse the vitamin present in the food item. Users can track their vitamin consumption, helping them achieve recommended daily allowances and addressing potential deficiencies.

Users following specialized diets can use the tool to ensure they are meeting their vitamin requirements. Thus, it helps in assisting individuals with specific dietary restrictions or preferences. This model supports research in nutrition and dietary science. Researchers and nutritionists can utilize the tool to analyse vitamin content in different foods, contributing to a better understanding of dietary patterns. The model showcases the capabilities of deep learning technologies in the field of nutrition. It demonstrates the application of state-of-the-art technologies to solve real-world problems in the health and wellness domain.

It empowers users with information about their food choices. Users can take a proactive role in managing their health by understanding the nutritional implications of their dietary decisions. It helps in integrating with wellness and fitness applications. Users can seamlessly incorporate nutritional information into their overall health and wellness tracking, promoting a holistic approach to personal health.

The purpose of vitamin detection using CNNs aligns with the broader goals of promoting health, providing valuable information to individuals, and leveraging advanced technologies for the betterment of society's well-being.

**Literature Survey:**

**Existing Problem:**

Traditional methods of vitamin detection often face several challenges, leading to limitations in accuracy, efficiency, and practicality. Traditional methods often involve manual sample preparation, chemical analyses, and complex laboratory procedures. The labour-intensive and time-consuming nature of these methods can slow down the detection process, making it impractical for real-time or high-throughput applications.

Many traditional methods require specialized equipment and expensive reagents. High costs associated with equipment and reagents can be a barrier, especially in resource-constrained environments, limiting widespread adoption. Some traditional methods may lack the sensitivity and specificity needed for accurate detection of low concentrations of vitamins or may cross-react with other compounds. Limited sensitivity and specificity can lead to false positives or negatives, affecting the reliability of the results.

Chemical-based methods often generate waste, leading to environmental concerns. Environmental considerations and the need for proper disposal add complexity to the overall process and contribute to sustainability challenges. Traditional methods may require complex and time-consuming sample preparation steps, including extraction and purification. Complicated sample preparation can introduce variability and increase the likelihood of errors in the analysis.

Many traditional methods are not portable and require a dedicated laboratory setting. Lack of portability limits the ability to perform on-site or field analyses, hindering applications in remote or resource-limited settings. Traditional methods may be susceptible to contamination during sample handling and processing. Contamination can compromise the accuracy of results and require rigorous quality control measures.

Addressing these challenges has motivated the exploration of alternative approaches, including the integration of modern technologies such as machine learning and deep learning, to enhance the efficiency and accuracy of vitamin detection processes.

**References:**


1. **"DeepVit: Deep learning for automatic vitamin identification from food images"**
    - **Authors:** A. R. Johnson et al.
    - **Published in:** 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)
    - **Summary:** This paper proposes DeepVit, a CNN-based model designed for automatic vitamin identification in food images. The study focuses on the application of deep learning techniques for nutrient detection.

2. **"VitaminNet: Deep learning to estimate vitamin levels in food images"**
    - **Authors:** S. Parikh et al.
    - **Published in:** 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)
    - **Summary:** VitaminNet is introduced as a CNN-based model that estimates vitamin levels in food images. The research explores the feasibility of deep learning for quantitative analysis of nutritional content.
3. **"DeepFood: Deep learning-based food image recognition for computer-aided dietary assessment"**
    - **Authors:** P. Zhuang et al.
    - **Published in:** 2015 IEEE International Conference on Multimedia and Expo (ICME)
    - **Summary:** While not specifically focused on vitamins, this paper introduces DeepFood, a CNN-based system for food image recognition. It lays the groundwork for subsequent research in nutrient analysis using deep learning.
4. **"NutriNet: A deep learning food and drink image recognition system for dietary assessment"**
    - **Authors:** Z. Xu et al.
    - **Published in:** 2017 IEEE International Conference on Multimedia and Expo (ICME)
    - **Summary:** NutriNet explores deep learning for dietary assessment, including the recognition of food items. While not exclusively about vitamins, the methodology could be extended to nutrient-specific analysis.
5. **"DeepNutri: A deep learning-based system for food portion estimation"**
    - **Authors:** Y. Zhang et al.
    - **Published in:** 2017 IEEE International Conference on Multimedia and Expo (ICME)
    - **Summary:** This paper introduces DeepNutri, a system for food portion estimation using deep learning. While not focusing on vitamins, the methodology could be relevant for nutrient quantification.

6. **"Automated Food Image Recognition and Nutrient Estimation: A Preliminary Study"**
    - **Authors:** M. Matsuda et al.
    - **Published in:** 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
    - **Summary:** The paper discusses the use of deep learning for automated food image recognition and nutrient estimation, laying the foundation for subsequent work in nutritional analysis.

**Problem Statement:**

Current methods for vitamin detection in food items rely on labour-intensive, time-consuming, and often expensive traditional techniques, leading to limitations in terms of efficiency, accuracy, and practicality. The demand for a rapid, non-invasive, and cost-effective solution for vitamin detection is growing, especially in the context of personalized nutrition and dietary assessment.

Some of the challenges are there are limited datasets available for vitamin content. These may lead to wrong results. The traditional methods of vitamin detection necessitate specialized equipment and expensive reagents, contributing to high operational costs. This poses a significant barrier, especially in resource-constrained environments where access to such resources is limited. The cost factor restricts the applicability and accessibility of vitamin detection, hindering its potential impact on public health.

Many traditional methods exhibit limited sensitivity and specificity, crucial factors for accurate vitamin detection. The inability to detect low concentrations of vitamins or the potential for cross-reactivity with other compounds can lead to inaccurate results. This limitation is particularly critical in assessing nutritional content accurately, where precision is paramount for informed decision-making.

The traditional methods also cause a lot of pollution to the environment as they involve use of chemicals. The chemicals are used and are not disposed correctly. They do not follow the correct procedure of the waste disposal leading to the contamination of the natural resources such as soil, water, and air. This will lead to lot of health issues.

The solution to these challenges lies in embracing modern approaches to vitamin detection. By leveraging machine learning and deep learning technologies, we aim to develop robust, efficient, and accessible methods that can revolutionize the field of nutritional assessment. These approaches have the potential to enhance accuracy, reduce the dependency on resource-intensive processes, and enable on-site analyses, thereby contributing to a more comprehensive understanding of nutritional content for improved public health outcomes. This project seeks to bridge the gap between traditional methodologies and the demands of contemporary nutritional science, fostering a paradigm shift towards advanced, technology-driven vitamin detection methods.

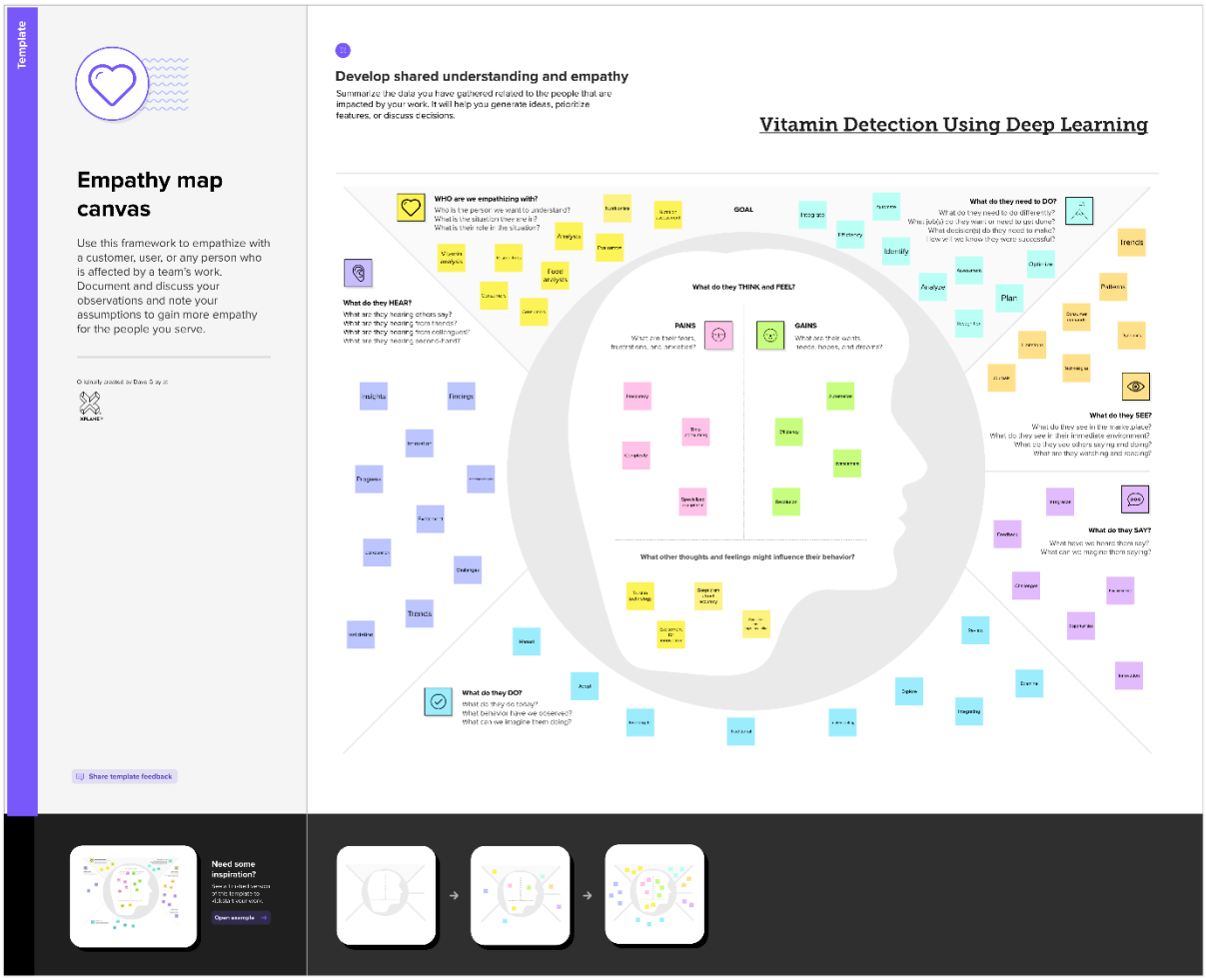
**Ideation and Proposed solution:**

**Empathy map:**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges. A problem statement often touches on the 5 w's (who, what, where, when and why) of the problem. We used mural empathy map canvas for analysing the 5 w's.

We have answered the following questions according to the users:

1. Who are we empathizing with?
2. What do they hear?
3. What do they do?
4. What do they need to do?
5. What do they see?
6. What do they say?
7. What do they think and feel?

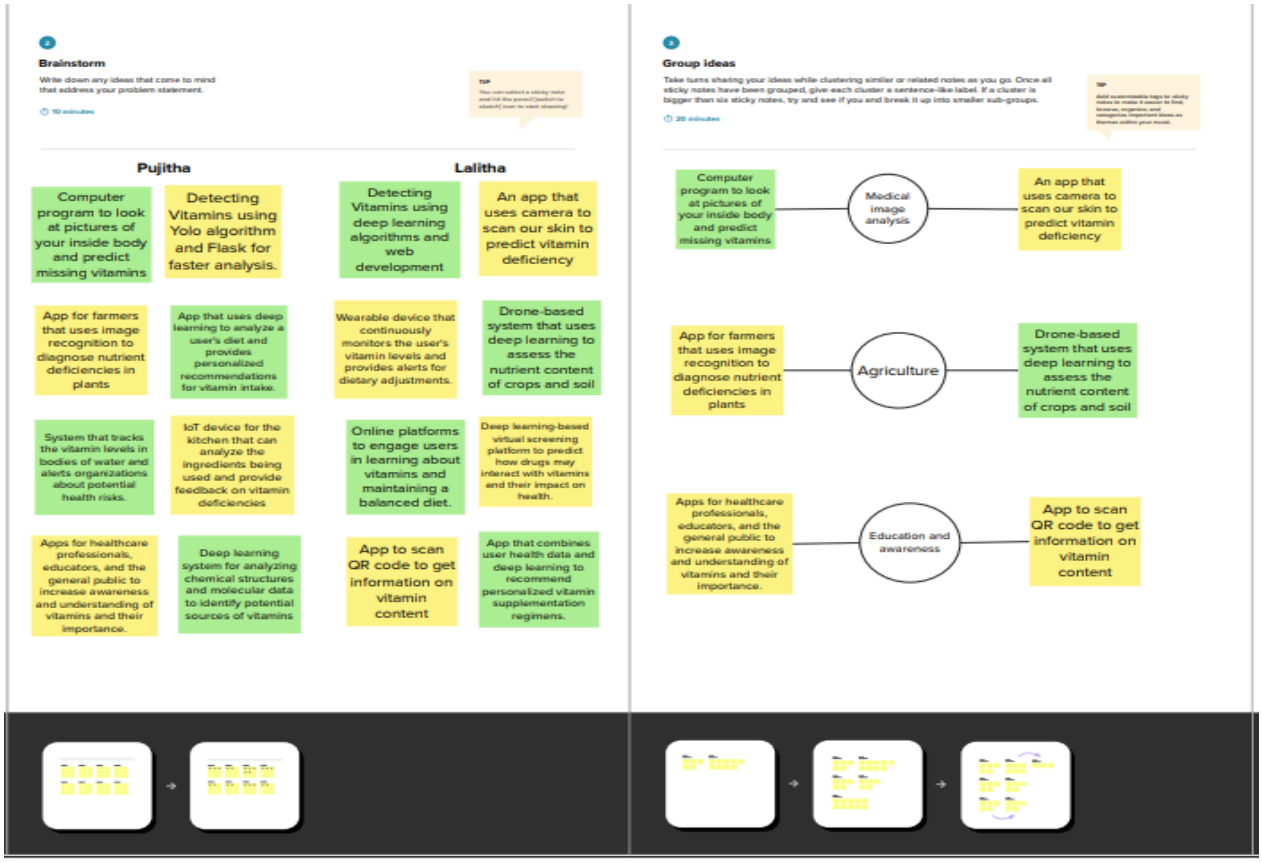


**Brainstorming Map:**

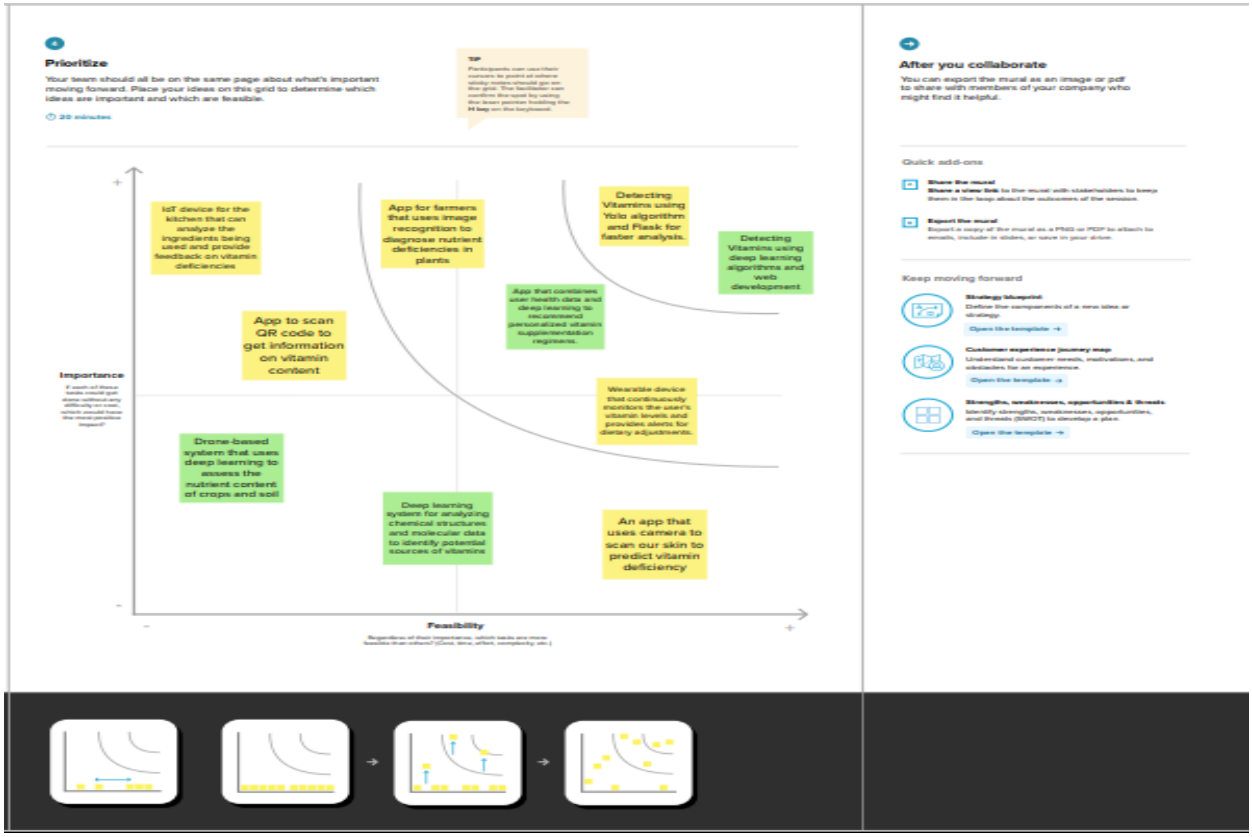
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you are not sitting in the same room. We used mural brainstorm and idea prioritization template for this phase.

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

## Step-2: Brainstorm, Idea Listing and Grouping




## Step-3: Idea Prioritization



**Project Design Phase-I**
**Proposed Solution**
**Template**

| | |
|---|---|
| Date | 31 October 2023 |
| Team ID | Team-592065 |
| Project Name | Vitamin Detection Using Deep Learning |
| Maximum Marks | 2 Marks |

**Proposed Solution Template:**

Project team shall fill the following information in proposed solution template.

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | • Traditional methods for vitamin analysis in food are time-consuming and specialized, hindering widespread and efficient assessment. |
| 2. | Idea / Solution description | • Utilize deep learning and VGG19, a powerful CNN, to automate and streamline vitamin identification in food, revolutionizing nutrition analysis. |
| 3. | Novelty / Uniqueness | • Unique application of VGG19 for rapid and automated vitamin analysis, offering a groundbreaking alternative to traditional methods. |
| 4. | Social Impact / Customer Satisfaction | • Enhance nutritional analysis and dietary planning, benefiting individuals and professionals in healthcare and nutrition. |
| 5. | Business Model (Revenue Model) | • Revenue potential through licensing to the food industry, offering the solution as a service, or forming partnerships with healthcare and nutrition professionals. |
| 6. | Scalability of the Solution | • Scalable solution leveraging deep learning techniques, capable of handling diverse and large datasets for widespread application. |

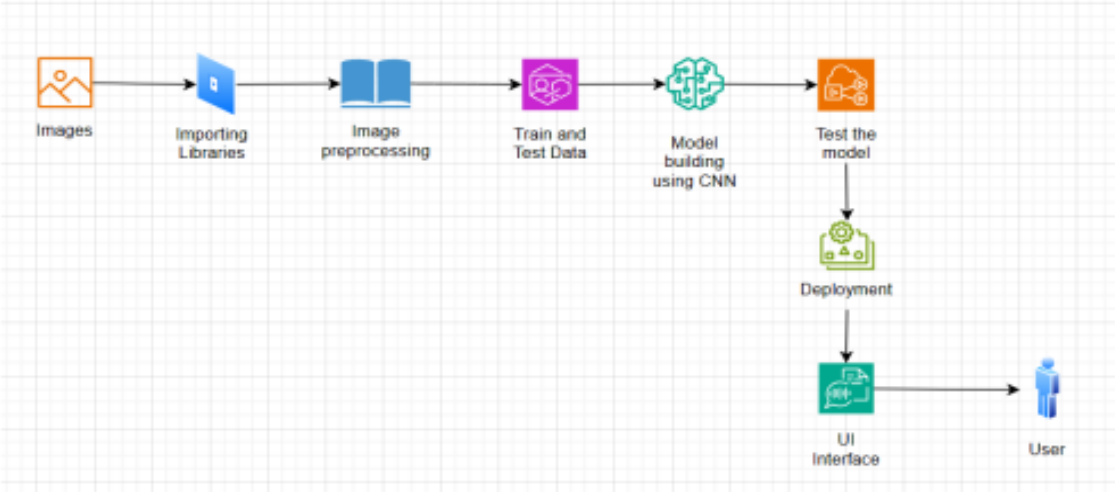
# Project Design
# Phase-I Solution
# Architecture

| | |
|---|---|
| Date | 3 November 2023 |
| Team ID | Team-592065 |
| Project Name | Vitamin Detection using deep learning |
| Maximum Marks | 4 Marks |

## About the project:

- Our project focuses on solving issues related to health and nutrition. This project helps us in identifying the vitamins in fruits and vegetables in the fastest way possible by using deep learning concepts. Convolutional Neural Networks is one of the most accurate algorithms that gives amazing results with at most accuracy of about 94 percent.
- We also built a website to easily interact with the user and allow the predictions to come in best possible way. It can easily be understood and used by the user. We have performed the required development processes and provided solutions which can further be improved by using more technologies and can be used in various fields for various purposes.
- This project is scalable and easily accessible by any user effortlessly. It also helps in saving time and money. In this way, we have built this project.

## Solution Architecture Diagram:



**Project Design Phase-II**

**Solution Requirements (Functional and Non funtional)**

| | |
|---|---|
| Date | 3 November 2023 |
| Team ID | Team-592065 |
| Project Name | Vitamin Detection using Deep learning |
| Maximum Marks | 4 Marks |

## Functional requirements:

Following are the functional requirements for the proposed solution:

| FR No: | Functional requirement | Sub requirement(story/task) |
|---|---|---|
| FR - 1 | Data Ingestion and preprocessing | • The system should have diverse datasets.<br>• The system must be preprocessed as train and test datasets. |
| FR - 2 | Model Training | • Model selection<br>• Fine Tuning |
| FR - 3 | Evaluation Metrics | • Validation<br>• Metrics reporting |
| FR - 4 | Deployment | • Integration<br>• Automation |
| FR - 5 | UI interface | • Upload Interface<br>• Results display |
| FR - 6 | Accessibility | • The UI Interface must be easily accessible by everyone. |

## Non-functional requirements:

Following are the non-functional requirements for the proposed solution:

| NFR No: | Non-functional requirements | Description |
|---|---|---|
| NFR – 1 | Usability | The user interface must be intuitive, requiring minimal training for users to perform vitamin detection tasks. |
| NFR – 2 | Security | All communication between the user interface and the backend system must be encrypted to protect user data. |
| NFR – 3 | Reliability | The system should handle errors, ensuring that a single failure does not disrupt overall functionality. |
| NFR – 4 | Performance | The system should be capable of processing a minimum of 100 images per minute for vitamin detection. |
| NFR – 5 | Availability | The system should be available during standard operating hours. |
| NFR - 6 | Scalability | The model should support scalability to accommodate an increase in the number of users and data volume. |

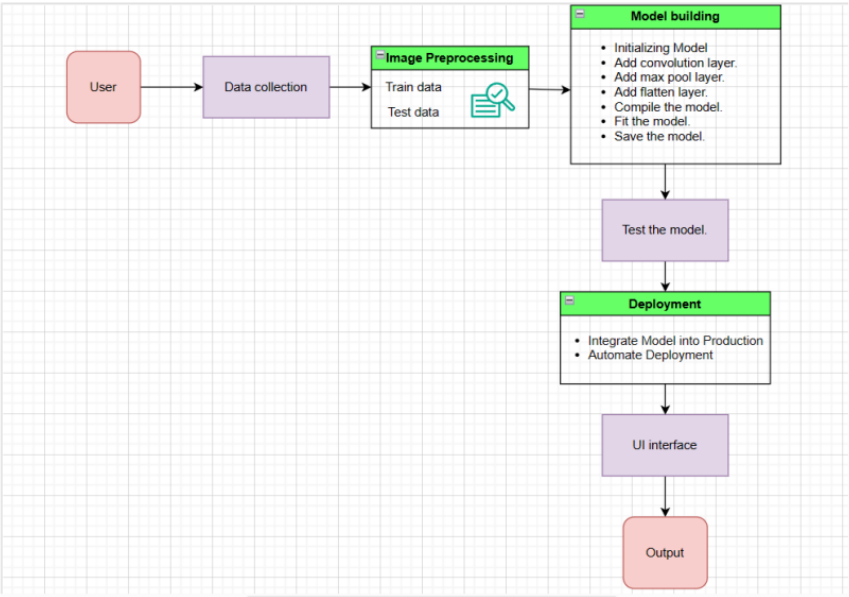
**Project Design Phase-II**
**Data Flow Diagram & User Stories**

| | |
|---|---|
| Date | 3 November 2023 |
| Team ID | Team-592065 |
| Project Name | Vitamin Detection using Deep learning |
| Maximum Marks | 4 Marks |

### Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored. The diagram provides an overview and may be adapted based on the specific details of the system architecture.

**Data flow diagram:**



**User Stories**

Use the below template to list all the user stories for the product.

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| User | Uses the model for vitamin detection. | User 1 | As a user, I want to upload the images for vitamin detection. | The system should have an upload button. | High | Sprint-1 |
| | | User 2 | As a user, I want to receive a clear results. | Results should be easily understandable. | High | Sprint-1 |
| Data Scientist | Analysis of the data used. | User 3 | As a data scientist, I want to select a pre-trained deep learning model for vitamin detection. | The system should allow the selection of a pre-trained model during the model configuration. | Medium | Sprint-2 |
| Administrator | Automation check. | User 4 | As a system administrator, I want to automate the deployment process. | The deployment process should be automated to reduce manual intervention. | High | Sprint-1 |
| User | Provides the necessary changes to be made. | User 5 | As a user, I want the system to provide real-time feedback on vitamin detection. | The system should display preliminary results as soon as the model processes the uploaded image. | Medium | Sprint-2 |
| User | Analysis of the project. | User 6 | As a user, I want to view detailed reports on model training metrics. | The system should display training metrics such as accuracy, precision, recall, and loss during the model training process. | Medium | Sprint-2 |
| User | Security check. | User 7 | As a user, I want the system to handle uploaded images securely. | Uploaded images should be stored securely and comply with data protection standards. | Medium | Sprint-2 |

**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| Date | 8 November 2023 |
| Team ID | Team- 592065 |
| Project Name | Vitamin Detection using Deep learning |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

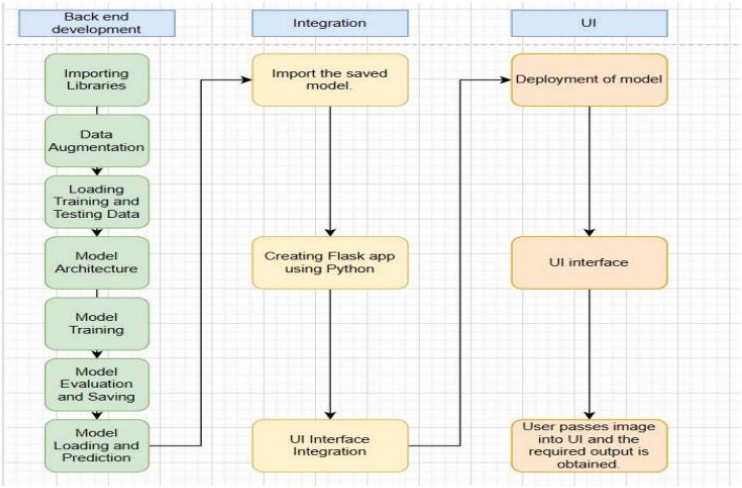


**Table-1: Components & Technologies:**

| S No: | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | How user interacts with application<br>Example: Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript. |
| 2. | Application Logic-1<br>(Data collection and Image Preprocessing) | To implement the model by utilizing the dataset provided. | Python |
| 3. | Application Logic-2 | To build the model and test the model to get the predictions required. | Tensor flow, Keras |
| 4. | Application Logic-3<br>(UI Interface and deployment) | To implement the web application and deploy resulting in the required output. | Flask |
| 5. | Database | Collecting the required images for the project. We have collected images of fruits, vegetables, and various food items. | Local system, Kaggle, Google. |
| 6. | File Storage | File storage requirements | Local File system, Google Drive. |
| 7. | Frame work | Used to integrate the web application. | Python Flask |
| 8. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud<br>Local Server Configuration:<br>Cloud Server Configuration: | Local system. |

**Table-2: Application Characteristics:**

| S No: | Component | Description | Technology |
|---|---|---|---|
| 1. | Image Classification | The primary task of the application is to classify images containing food items or supplements into different categories based on the presence of specific vitamins. | Python |
| 2. | Deep Learning Model | Utilizes deep learning models, such as convolutional neural networks (CNNs), for feature extraction and learning hierarchical representations from input images. | Python |
| 3. | Training data | It has diverse dataset of images representing various food items or products containing different vitamins. The model learns from this dataset during the training phase. | Python |
| 4. | Vitamin specific cases | The model is trained to recognize specific classes related to different vitamins. For example, classes may include "Vitamin A," "Vitamin B," "Vitamin C," and so on. | Python |
| 5. | Preprocessing Techniques | Incorporates image preprocessing techniques, such as resizing, normalization, and augmentation, to enhance the model's ability to generalize and improve performance. | Python |
| 6. | UI interface | May include a user-friendly web interface for users to interact with the application. Users can upload images, and the application provides predictions. | HTML, CSS, Javascript |
| 7. | Evaluation metrics | Evaluates model performance using accuracy metrics and potentially other metrics relevant to the application. | Python |

## Project Planning Phase

### Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

| | |
|---|---|
| Date | 18 November 2023 |
| Team ID | Team-592065 |
| Project Name | Vitamin Detection using deep learning |
| Maximum Marks | 8 Marks |

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

Use the below template to create product backlog and sprint schedule

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint 1 | Project setup | USN-1 | Set the development environment by downloading the required software and tools to start the vitamin detection using deep learning project. | 2 | High | Pujitha |
| Sprint 1 | Data Collection | USN-2 | Collect diverse dataset of different types of images containing vitamin A, vitamin B, vitamin C, vitamin D and vitamin E. | 1 | High | Pujitha |
| Sprint 2 | Data preprocessing | USN-3 | Preprocess the collected dataset by resizing images, normalizing pixel values, and splitting the dataset into train and test sets. | 2 | High | Pujitha |
| Sprint 2 | Model development | USN-4 | Evaluate deep learning models such as CNN to select the best model for vitamin detection. | 2 | High | Pujitha and Lalitha |
| Sprint 3 | Training | USN-5 | Implement data augmentation techniques to improve accuracy of the model. | 1 | Medium | Lalitha |
| Sprint 4 | Model deployment and integration. | USN-6 | Developing an UI Interface to detect the vitamin in the food item. | 2 | High | Lalitha |
| Sprint 4 | Testing the model | USN-7 | Testing the model by uploading image of a food item and getting the required prediction. | 1 | High | Lalitha |

**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

| 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 | 20 | 4 Days | 30 Oct 2023 | 02 Nov 2023 | 18 | 03 Nov 2023 |
| Sprint-2 | 20 | 5 Days | 03 Nov 2023 | 07 Nov 2023 | 16 | 10 Nov 2023 |
| Sprint-3 | 20 | 10 Days | 08 Nov 2023 | 17 Nov 2023 | 17 | 19 Nov 2023 |
| Sprint-4 | 20 | 4 Days | 18 Nov 2023 | 22 Nov 2023 | 15 | 25 Nov 2023 |

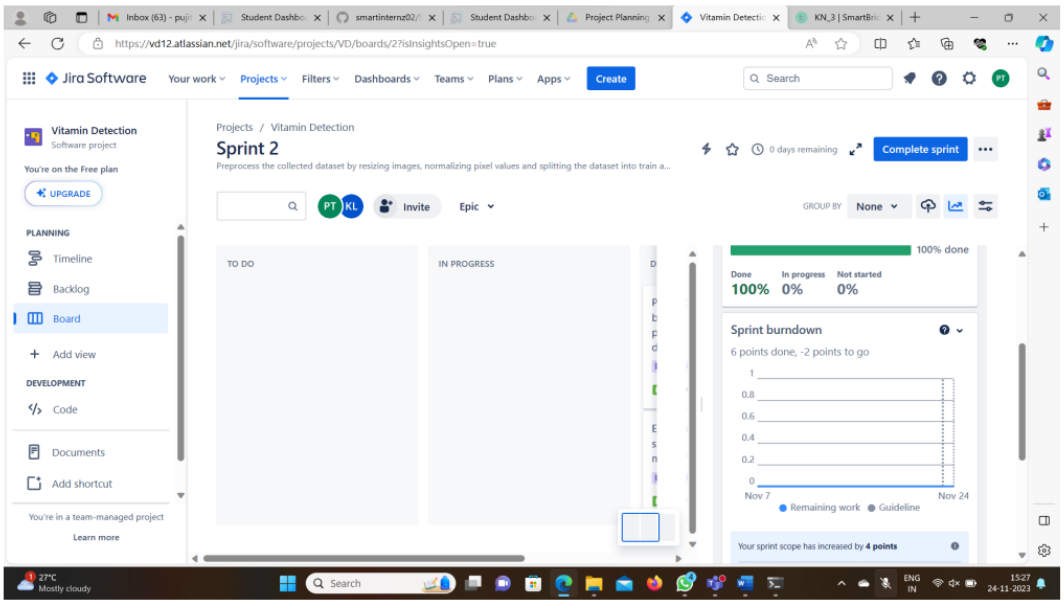
**Velocity:**

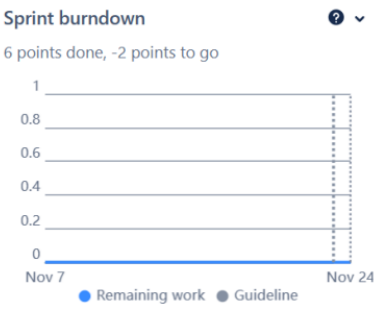
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let us calculate the team's average velocity (AV) per iteration unit (story points per day)

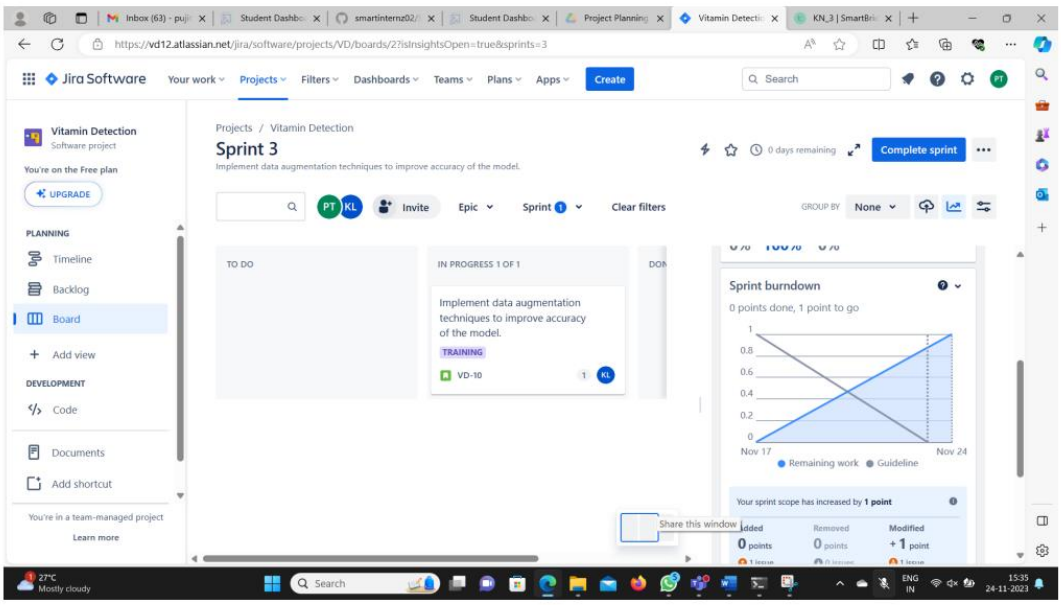
Therefore, AV = sprint duration/velocity = (18+16+17+15)/4 = 16.5

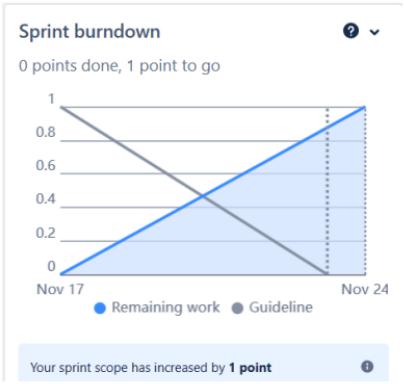
**Burndown Chart:**

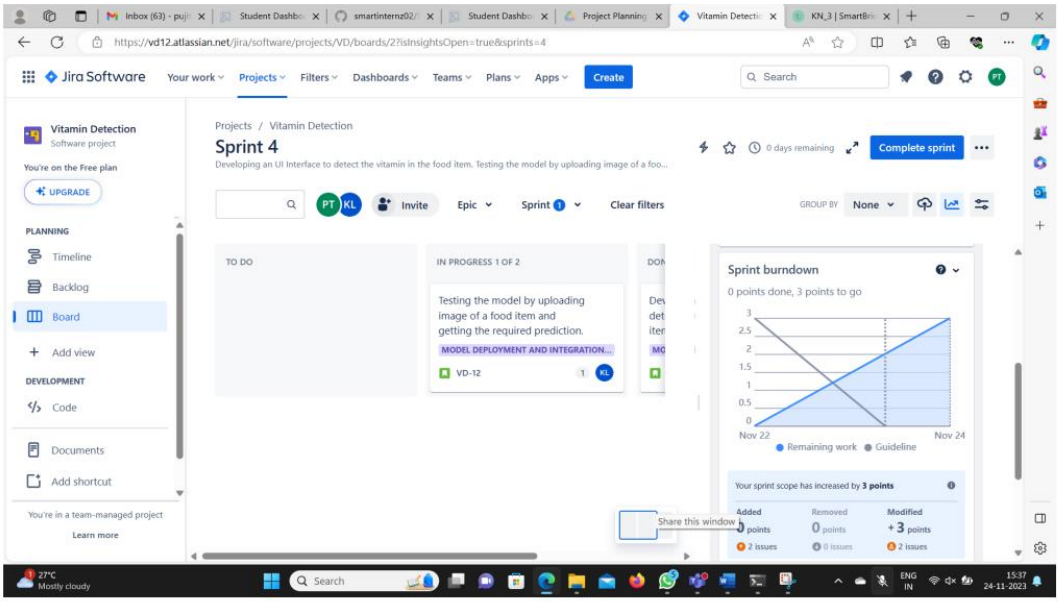
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

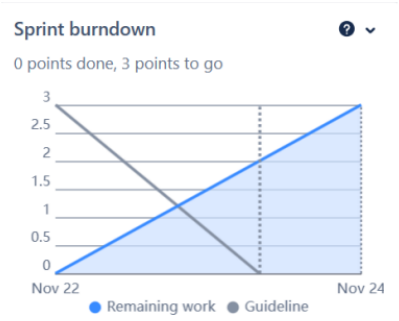

Jira Software
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Search
Vitamin Detection
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
Documents
Add shortcut
You're in a team-managed project
Learn more
Projects / Vitamin Detection
Sprint 2
Preprocess the collected dataset by resizing images, normalizing pixel values and splitting the dataset into train a...
0 days remaining
Complete sprint
Invite
Epic
GROUP BY
None
TO DO
IN PROGRESS
100% done
Done
In progress
Not started
100%
0%
0%
Sprint burndown
6 points done, -2 points to go
1
0.8
0.6
0.4
0.2
0
Nov 7
Nov 24
Remaining work
Guideline
Your sprint scope has increased by 4 points
27°C
Mostly cloudy
Search
ENG
IN
15:27
24-11-2023

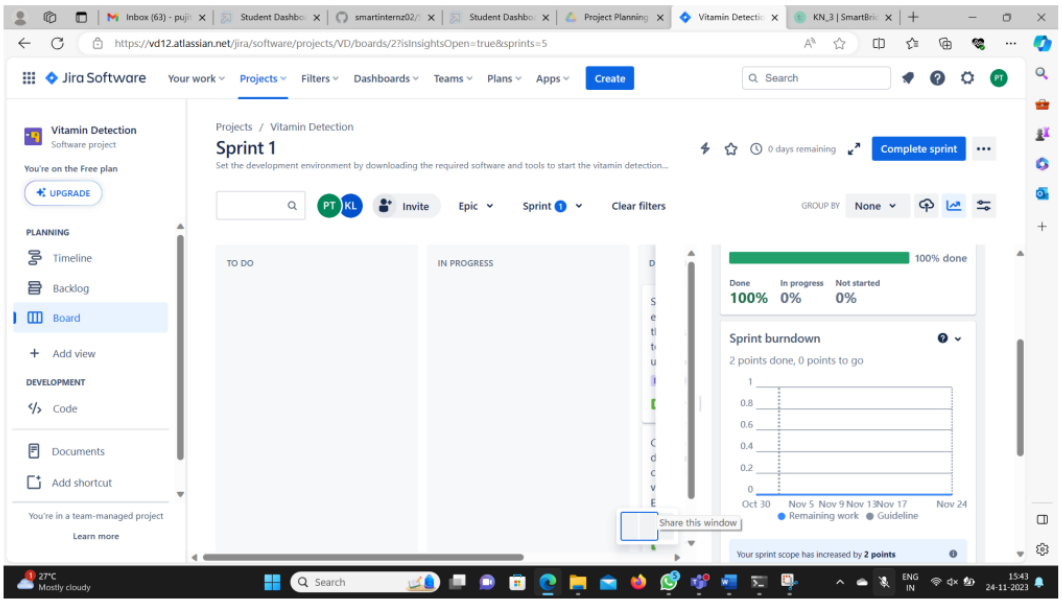

Sprint burndown
6 points done, -2 points to go
1
0.8
0.6
0.4
0.2
0
Nov 7
Nov 24
Remaining work
Guideline

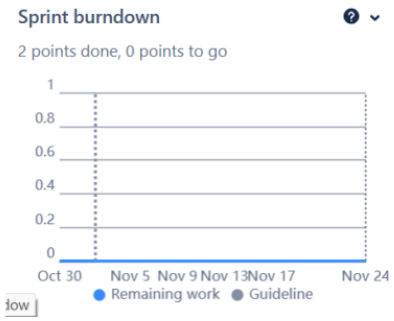

Jira Software
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Search
Vitamin Detection
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
Documents
Add shortcut
You're in a team-managed project
Learn more
Projects / Vitamin Detection
Sprint 3
Implement data augmentation techniques to improve accuracy of the model.
0 days remaining
Complete sprint
Invite
Epic
Sprint
Clear filters
GROUP BY
None
TO DO
IN PROGRESS 1 OF 1
Implement data augmentation techniques to improve accuracy of the model.
TRAINING
VD-10
Sprint burndown
0 points done, 1 point to go
1
0.8
0.6
0.4
0.2
0
Nov 17
Nov 24
Remaining work
Guideline
Your sprint scope has increased by 1 point
Added
0 points
Removed
0 points
Modified
+1 point
Share this window
27°C
Mostly cloudy
Search
ENG
IN
15:35
24-11-2023

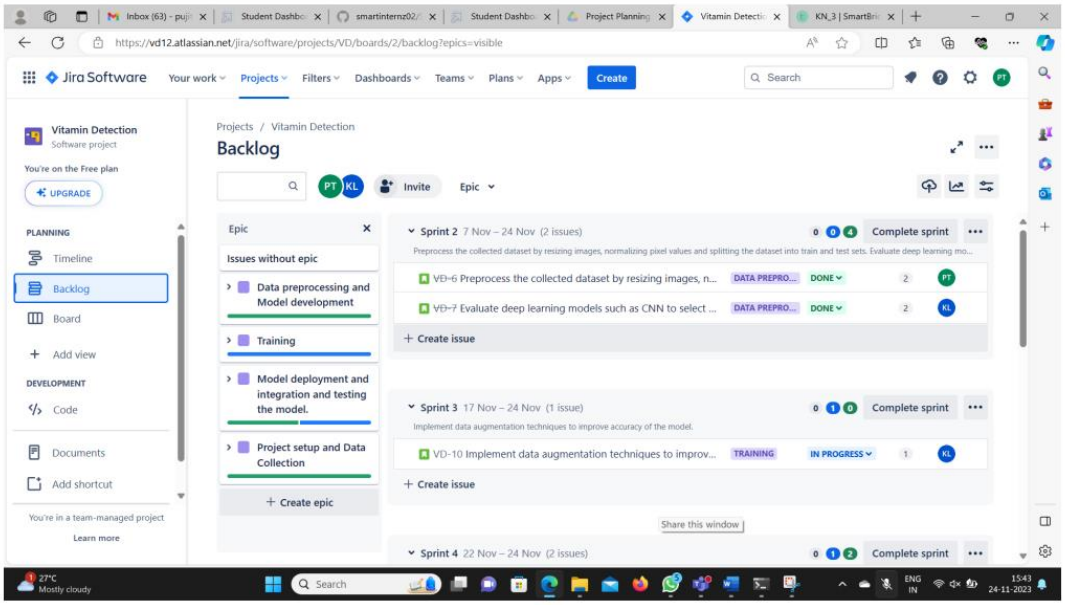

Sprint burndown
0 points done, 1 point to go
1
0.8
0.6
0.4
0.2
0
Nov 17
Nov 24
Remaining work
Guideline
Your sprint scope has increased by 1 point

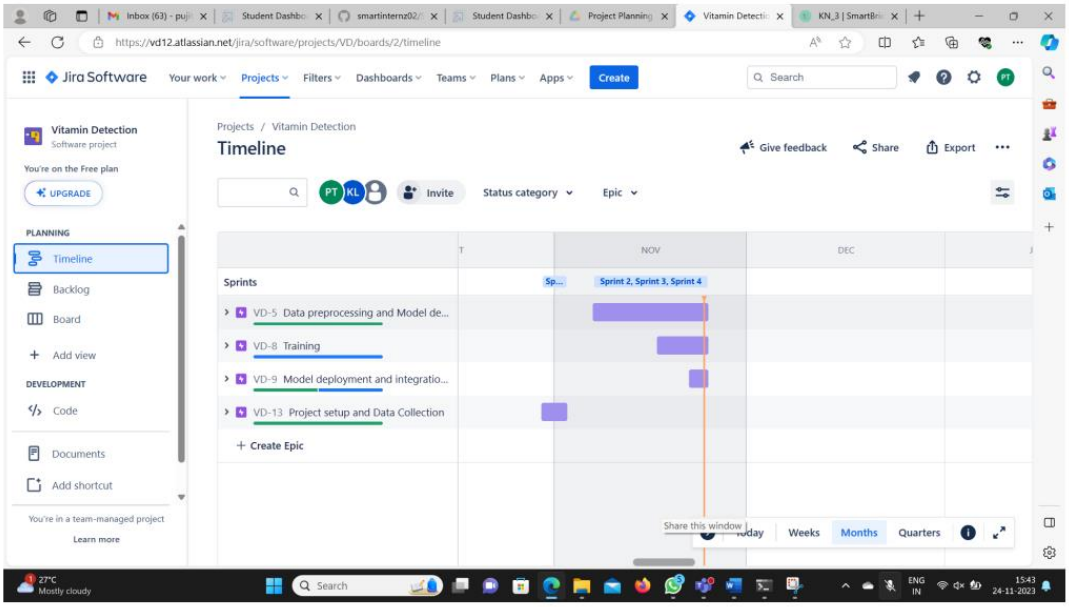

Jira Software
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Search
Vitamin Detection
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
Documents
Add shortcut
You're in a team-managed project
Learn more
Projects / Vitamin Detection
Sprint 4
Developing an UI Interface to detect the vitamin in the food item. Testing the model by uploading image of a foo...
0 days remaining
Complete sprint
Invite
Epic
Sprint
Clear filters
GROUP BY
None
TO DO
IN PROGRESS 1 OF 2
Testing the model by uploading image of a food item and getting the required prediction.
MODEL DEPLOYMENT AND INTEGRATION...
VD-12
Sprint burndown
0 points done, 3 points to go
3
2.5
2
1.5
1
0.5
0
Nov 22
Nov 24
Remaining work
Guideline
Your sprint scope has increased by 3 points
Added
0 points
2 issues
Removed
0 points
0 issues
Modified
+ 3 points
2 issues
Share this window
Search
ENG
IN
15:37
24-11-2023

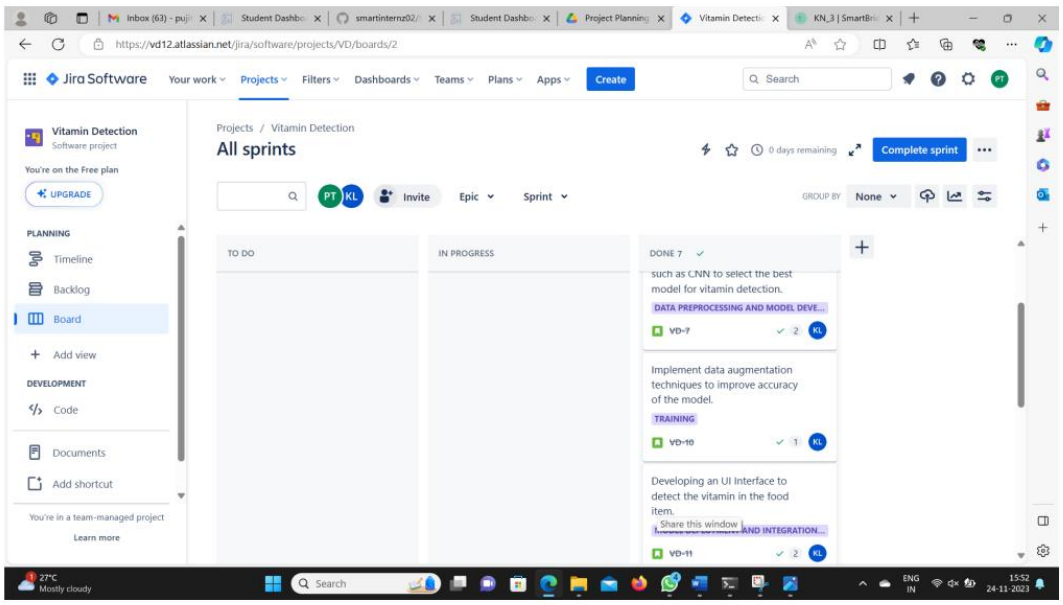

Sprint burndown
0 points done, 3 points to go
3
2.5
2
1.5
1
0.5
0
Nov 22
Nov 24
Remaining work
Guideline


Jira Software
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Search
Vitamin Detection
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
Documents
Add shortcut
You're in a team-managed project
Learn more
Projects / Vitamin Detection
Sprint 1
Set the development environment by downloading the required software and tools to start the vitamin detection...
0 days remaining
Complete sprint
Invite
Epic
Sprint
Clear filters
GROUP BY
None
TO DO
IN PROGRESS
100% done
Done
100%
In progress
0%
Not started
0%
Sprint burndown
2 points done, 0 points to go
1
0.8
0.6
0.4
0.2
0
Oct 30
Nov 5
Nov 9
Nov 13
Nov 17
Nov 24
Remaining work
Guideline
Share this window
Your sprint scope has increased by 2 points
27°C
Mostly cloudy
Search
ENG
IN
15:43
24-11-2023


Sprint burndown
2 points done, 0 points to go
1
0.8
0.6
0.4
0.2
0
Oct 30
Nov 5
Nov 9
Nov 13
Nov 17
Nov 24
Remaining work
Guideline
dow


Jira Software
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Search
Vitamin Detection
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
Documents
Add shortcut
You're in a team-managed project
Learn more
Projects / Vitamin Detection
Backlog
Invite
Epic
Issues without epic
Data preprocessing and Model development
Training
Model deployment and integration and testing the model.
Project setup and Data Collection
Create epic
Sprint 2 7 Nov – 24 Nov (2 issues)
Complete sprint
VD-6 Preprocess the collected dataset by resizing images, n...
VD-7 Evaluate deep learning models such as CNN to select ...
Create issue
Sprint 3 17 Nov – 24 Nov (1 issue)
VD-10 Implement data augmentation techniques to improv...
Share this window
Sprint 4 22 Nov – 24 Nov (2 issues)


Jira Software
Your work
Projects
Filters
Dashboards
Teams
Plans
Apps
Create
Search
Vitamin Detection
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
Documents
Add shortcut
You're in a team-managed project
Learn more
Projects / Vitamin Detection
Timeline
Give feedback
Share
Export
Invite
Status category
Epic
NOV
DEC
Sprints
Sprint 2, Sprint 3, Sprint 4
VD-5 Data preprocessing and Model de...
VD-8 Training
VD-9 Model deployment and integratio...
VD-13 Project setup and Data Collection
Create Epic
Share this window
Today
Weeks
Months
Quarters




## Coding:

## Python code:

In [1]:
```
#import model building libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

# Image Preprocessing

In [2]:
```
from keras.preprocessing.image import ImageDataGenerator
```

In [3]:
```
#2.configure image data generator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [4]:
```
#3.Apply image data generator functionality to train and test images
x_train=train_datagen.flow_from_directory(r'C:\Users\kanth\Downloads\vitamins_detection',target_size=(64,64),batch_size=32,class_
x_test = test_datagen.flow_from_directory(r'C:\Users\kanth\Downloads\test_data',target_size = (64,64),batch_size = 32,class_mode
```

```
Found 8968 images belonging to 5 classes.
Found 224 images belonging to 5 classes.
```

In [5]:
```
print(x_train.class_indices)
```

```
{'vitaminA': 0, 'vitaminB': 1, 'vitaminC': 2, 'vitaminD': 3, 'vitaminE': 4}
```

## Model Building

In [6]:
```
#2.initializing the model
model=Sequential()
```

In [7]:
```
#3.add convolution layer(no.of filters,size of filter,input shape)
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation="relu"))
```

In [8]:
```
#add max pool layer(pool_size)
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [9]:
```
#add flatten layer  ---input of ann
model.add(Flatten())
```

In [10]:
```
#add hidden layer
model.add(Dense(units=128,activation="relu"))
```

In [11]:
```
#add output layer
model.add(Dense(units=5,activation="softmax"))
```

In [12]:
```
#Compile the model (loss fucntion,accuracy,optimizer)
model.compile(loss="categorical_crossentropy",optimizer="adam",metrics="accuracy")
```

In [13]:
```
#fit model (x_train,steps_per epoch,epochs,validation_data,validation_steps)
model.fit(x_train,steps_per_epoch=280,epochs=100,validation_data=x_test)
```

```
Epoch 1/100
280/280 [==============================] - 60s 210ms/step - loss: 1.3305 - accuracy: 0.4633 - val_loss: 1.0259 - val_accuracy:
0.5580
Epoch 2/100
280/280 [==============================] - 46s 164ms/step - loss: 1.1891 - accuracy: 0.5208 - val_loss: 1.0125 - val_accuracy:
0.5848
Epoch 3/100
280/280 [==============================] - 47s 166ms/step - loss: 1.1300 - accuracy: 0.5443 - val_loss: 0.9816 - val_accuracy:
0.5804
Epoch 4/100
280/280 [==============================] - 47s 168ms/step - loss: 1.0961 - accuracy: 0.5608 - val_loss: 0.9057 - val_accuracy:
0.6473
Epoch 5/100
280/280 [==============================] - 46s 166ms/step - loss: 1.0633 - accuracy: 0.5758 - val_loss: 1.0016 - val_accuracy:
0.5938
Epoch 6/100
280/280 [==============================] - 46s 163ms/step - loss: 1.0280 - accuracy: 0.5924 - val_loss: 0.8850 - val_accuracy:
0.6339
Epoch 7/100
280/280 [==============================] - 46s 165ms/step - loss: 1.0086 - accuracy: 0.5956 - val_loss: 1.0027 - val_accuracy:
```

In [14]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

 dense (Dense)               (None, 128)               3936384

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 3,937,925
Trainable params: 3,937,925
Non-trainable params: 0
_________________________________________________________________
```

In [15]:
```
#save our model
model.save("vitamins.h5")
```

## Test the model

In [16]:
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

In [17]:
```
import tensorflow as tf
```

In [18]:
```
model=tf.keras.models.load_model(r"C:\Users\kanth\Downloads\vitamins.h5",compile=False)
```

In [19]:
```
#D:\SmartBridge\VIT_morning_slot\dataset\Testing\elephants\nature_3306013__340.jpg
img=image.load_img(r"C:\Users\kanth\Downloads\carrot.jpeg",target_size=(64,64))
```

In [20]:
```
img
```

Out[20]:

In [21]:
```
x=image.img_to_array(img)
x
```

Out[21]:
```
array([[[167., 158., 153.],
        [163., 154., 149.],
        [174., 165., 160.],
        ...,
        [173., 173., 171.],
        [169., 170., 165.],
        [193., 192., 197.]],

       [[138., 129., 124.],
        [164., 155., 150.],
        [179., 170., 165.],
        ...,
        [174., 174., 172.],
        [176., 177., 172.],
        [196., 196., 198.]],

       [[153., 144., 139.],
        [152., 143., 138.],
        [178., 169., 164.],
        ...,
        [176., 176., 174.],
        [201., 202., 197.],
        [185., 185., 183.]],

       ...,


       ...,

       [[ 66.,  57.,  52.],
        [ 93.,  84.,  79.],
        [ 89.,  80.,  75.],
        ...,
        [ 43.,  36.,  30.],
        [ 58.,  44.,  33.],
        [ 98.,  88.,  79.]],

       [[ 70.,  61.,  56.],
        [ 88.,  79.,  74.],
        [ 81.,  72.,  67.],
        ...,
        [ 23.,  20.,  13.],
        [ 67.,  57.,  45.],
        [ 93.,  86.,  80.]],

       [[ 88.,  82.,  82.],
        [ 78.,  72.,  72.],
        [ 78.,  72.,  72.],
        ...,
        [ 36.,  29.,  23.],
        [ 98.,  84.,  73.],
        [ 99.,  84.,  81.]]], dtype=float32)
```

In [22]:
```
x=np.expand_dims(x,axis=0)
```

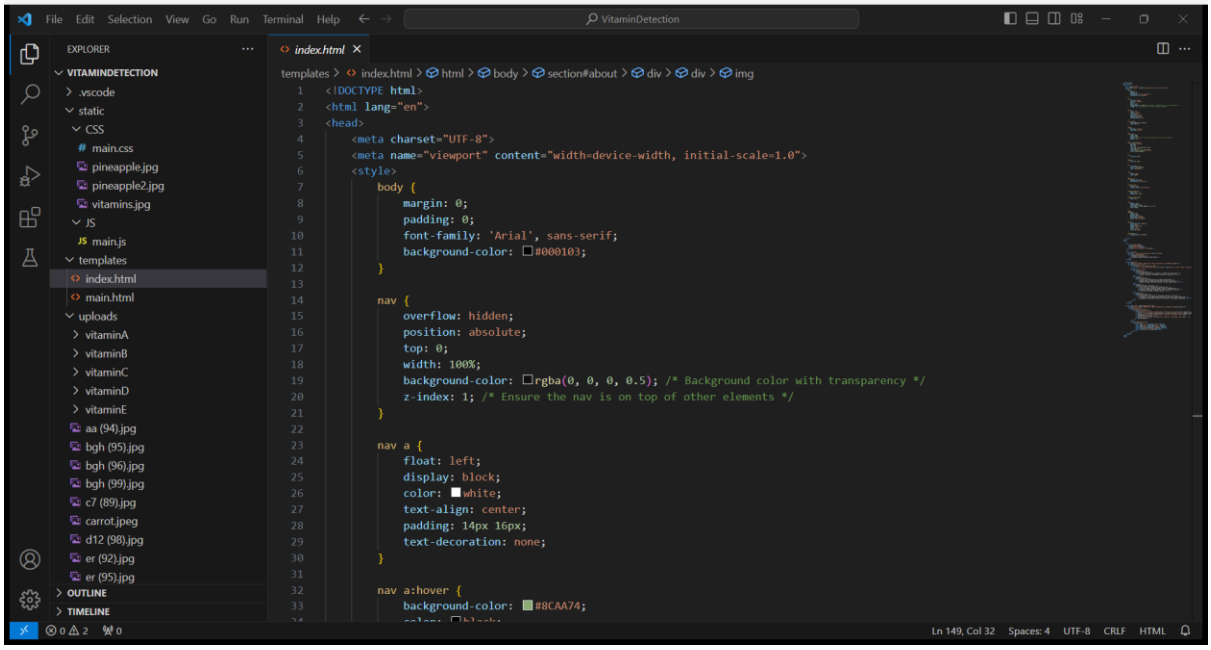
In [23]:
```
x.ndim
```

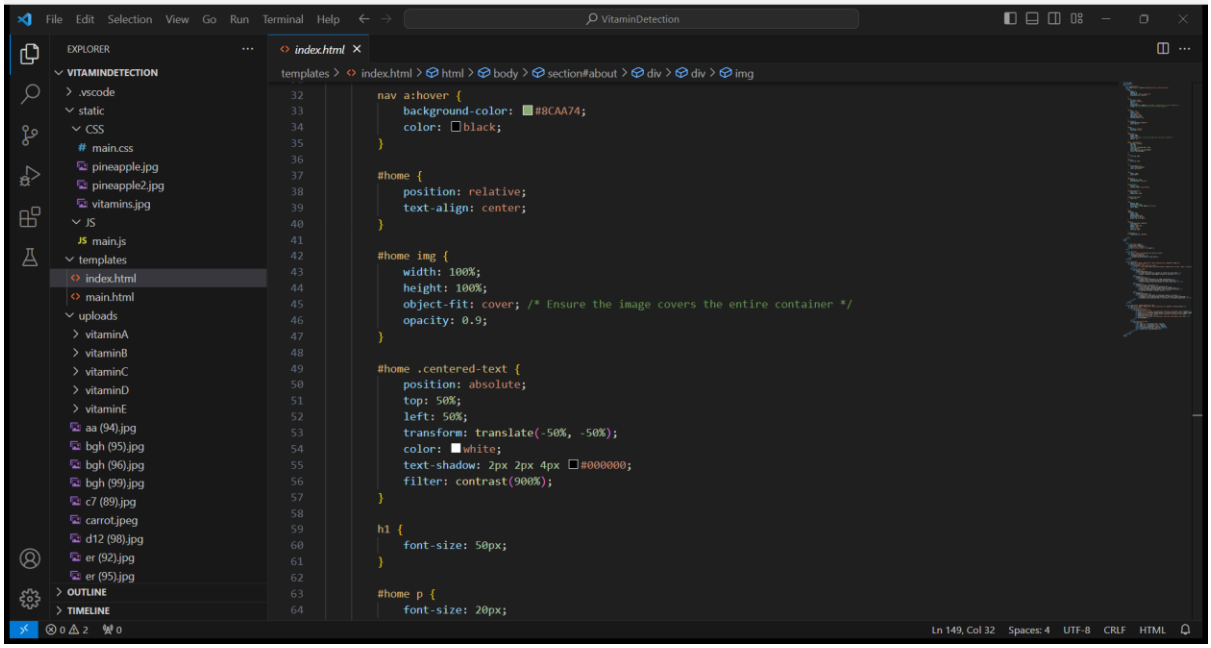
Out[23]:
```
4
```

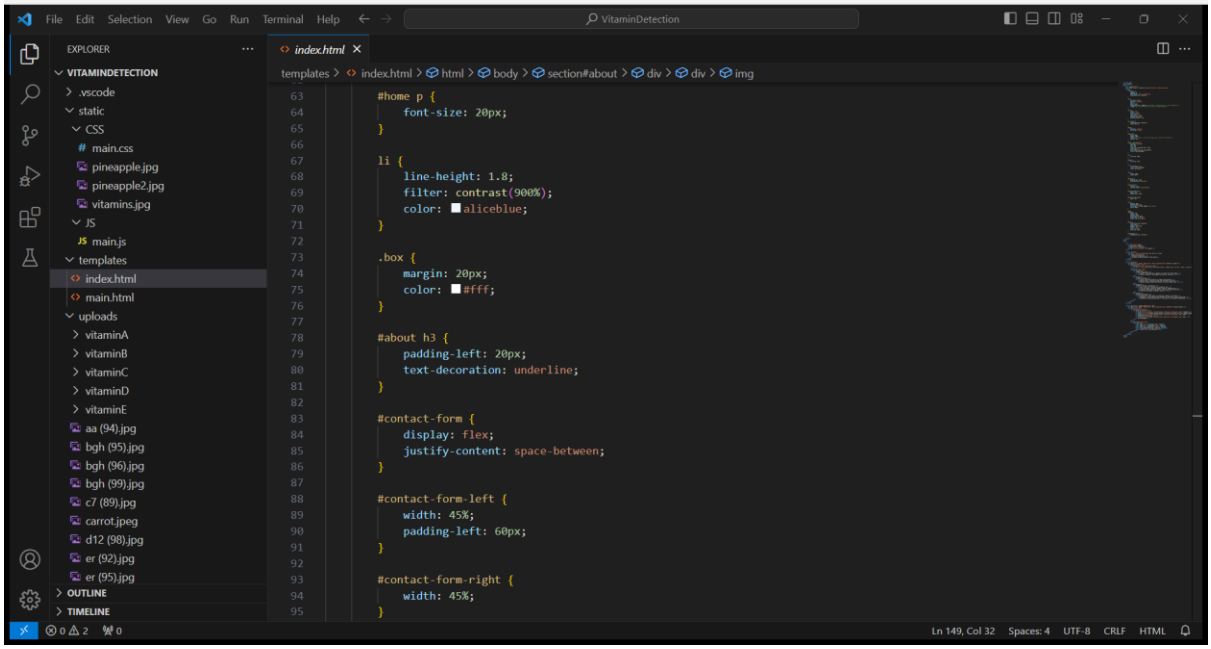
```
In [24]: x.shape
Out[24]: (1, 64, 64, 3)

In [25]: pred=model.predict(x)
         1/1 [==============================] - 1s 621ms/step

In [26]: pred
Out[26]: array([[1., 0., 0., 0., 0.]], dtype=float32)

In [27]: pred_class=np.argmax(pred,axis=1)

In [28]: pred_class[0]
Out[28]: 0

In [30]: index=['vitamin A', 'vitamin B', 'vitamin C', 'vitamin D','vitamin E']
         result=str(index[pred_class[0]])

In [31]: result
Out[31]: 'vitamin A'
```

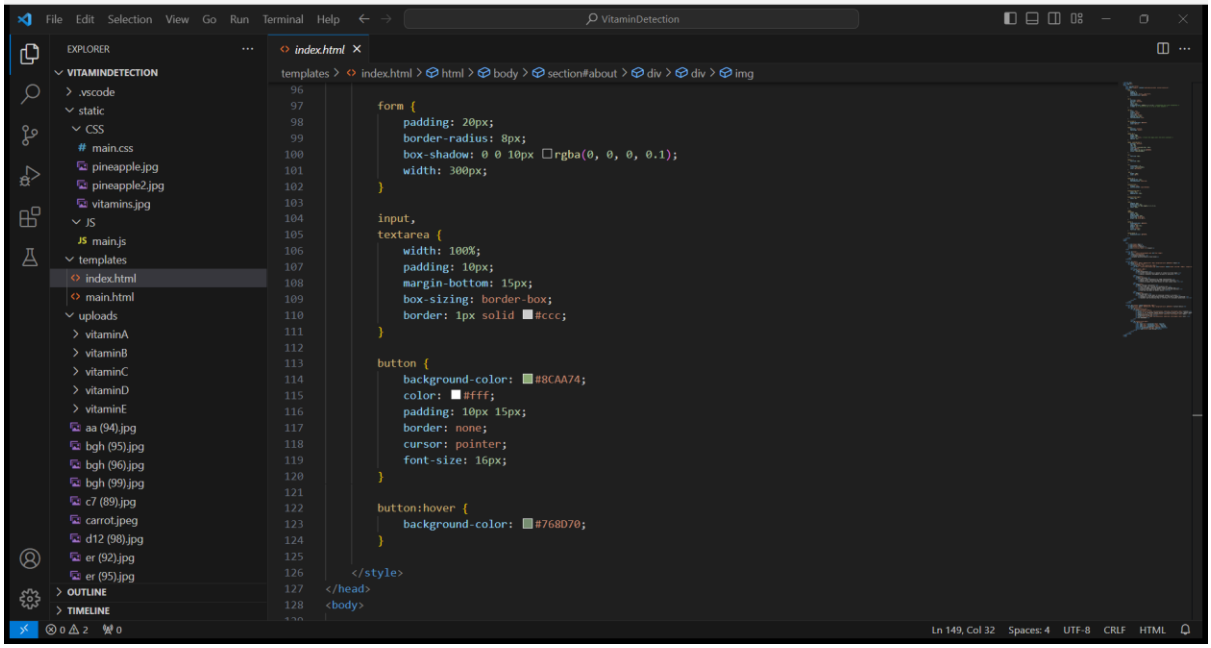
## HTML pages code:

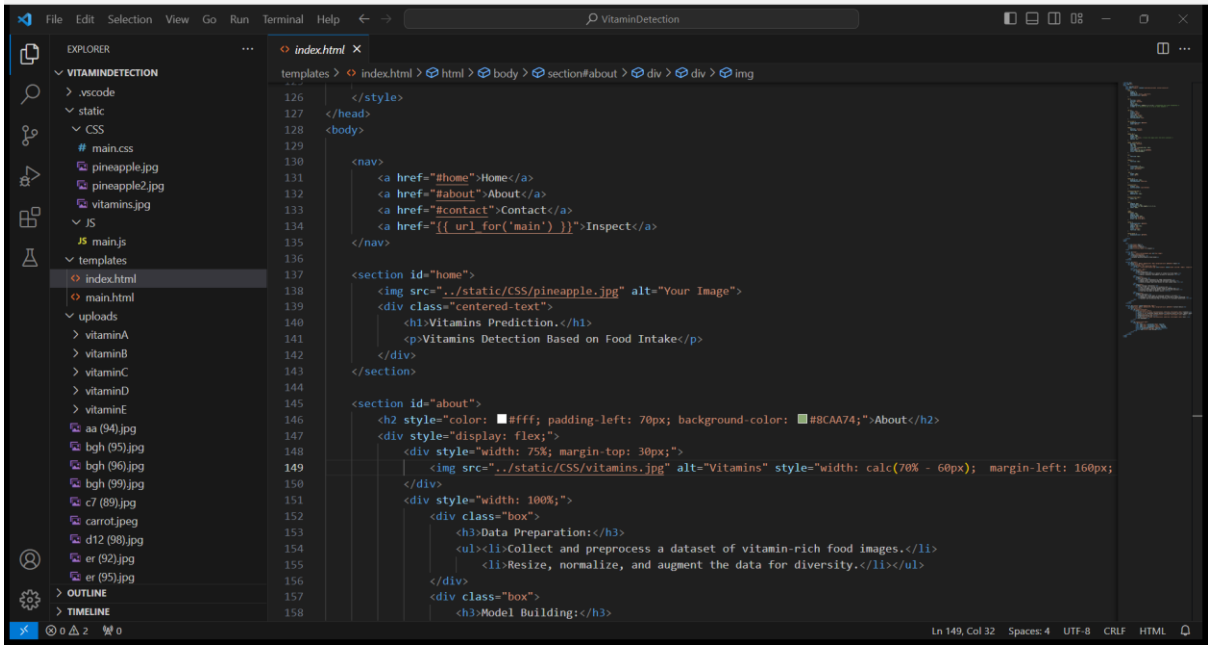
## Index.html file




File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
index.html
templates > index.html > html > body > section#about > div > div > img
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
nav a:hover {
background-color: #8CAA74;
color: black;
}
#home {
position: relative;
text-align: center;
}
#home img {
width: 100%;
height: 100%;
object-fit: cover; /* Ensure the image covers the entire container */
opacity: 0.9;
}
#home .centered-text {
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
color: white;
text-shadow: 2px 2px 4px #000000;
filter: contrast(900%);
}
h1 {
font-size: 50px;
}
#home p {
font-size: 20px;
Ln 149, Col 32 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
index.html
templates > index.html > html > body > section#about > div > div > img
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
#home p {
font-size: 20px;
}
li {
line-height: 1.8;
filter: contrast(900%);
color: aliceblue;
}
.box {
margin: 20px;
color: #fff;
}
#about h3 {
padding-left: 20px;
text-decoration: underline;
}
#contact-form {
display: flex;
justify-content: space-between;
}
#contact-form-left {
width: 45%;
padding-left: 60px;
}
#contact-form-right {
width: 45%;
}
Ln 149, Col 32 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
index.html
templates > index.html > html > body > section#about > div > div > img
form {
padding: 20px;
border-radius: 8px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
width: 300px;
}
input,
textarea {
width: 100%;
padding: 10px;
margin-bottom: 15px;
box-sizing: border-box;
border: 1px solid #ccc;
}
button {
background-color: #8CAA74;
color: #fff;
padding: 10px 15px;
border: none;
cursor: pointer;
font-size: 16px;
}
button:hover {
background-color: #768D70;
}
</style>
</head>
<body>
Ln 149, Col 32 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
index.html
templates > index.html > html > body > section#about > div > div > img
</style>
</head>
<body>
<nav>
<a href="#home">Home</a>
<a href="#about">About</a>
<a href="#contact">Contact</a>
<a href="{{ url_for('main') }}">Inspect</a>
</nav>
<section id="home">
<img src="../static/CSS/pineapple.jpg" alt="Your Image">

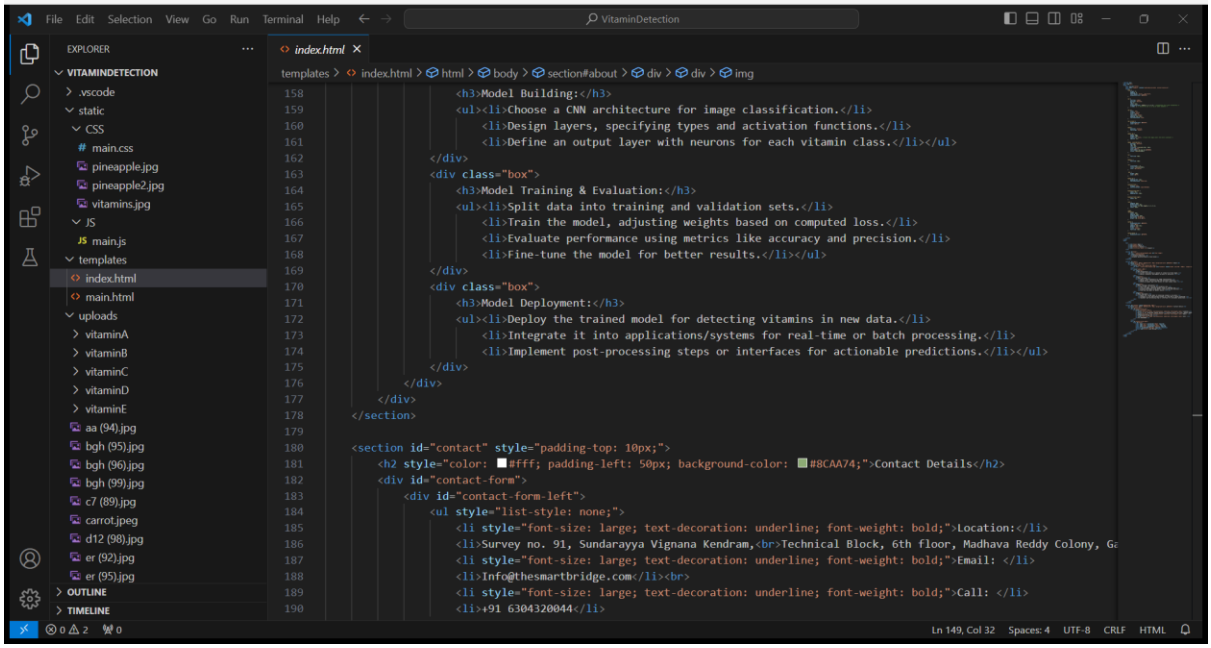
<h1>Vitamins Prediction.</h1>
Vitamins Detection Based on Food Intake

</section>
<section id="about">
<h2 style="color: #fff; padding-left: 70px; background-color: #8CAA74;">About</h2>

<img src="../static/CSS/vitamins.jpg" alt="Vitamins" style="width: calc(70% - 60px); margin-left: 160px;

<h3>Data Preparation:</h3>
<ul><li>Collect and preprocess a dataset of vitamin-rich food images.</li>
<li>Resize, normalize, and augment the data for diversity.</li></ul>

<h3>Model Building:</h3>
Ln 149, Col 32 Spaces: 4 UTF-8 CRLF HTML

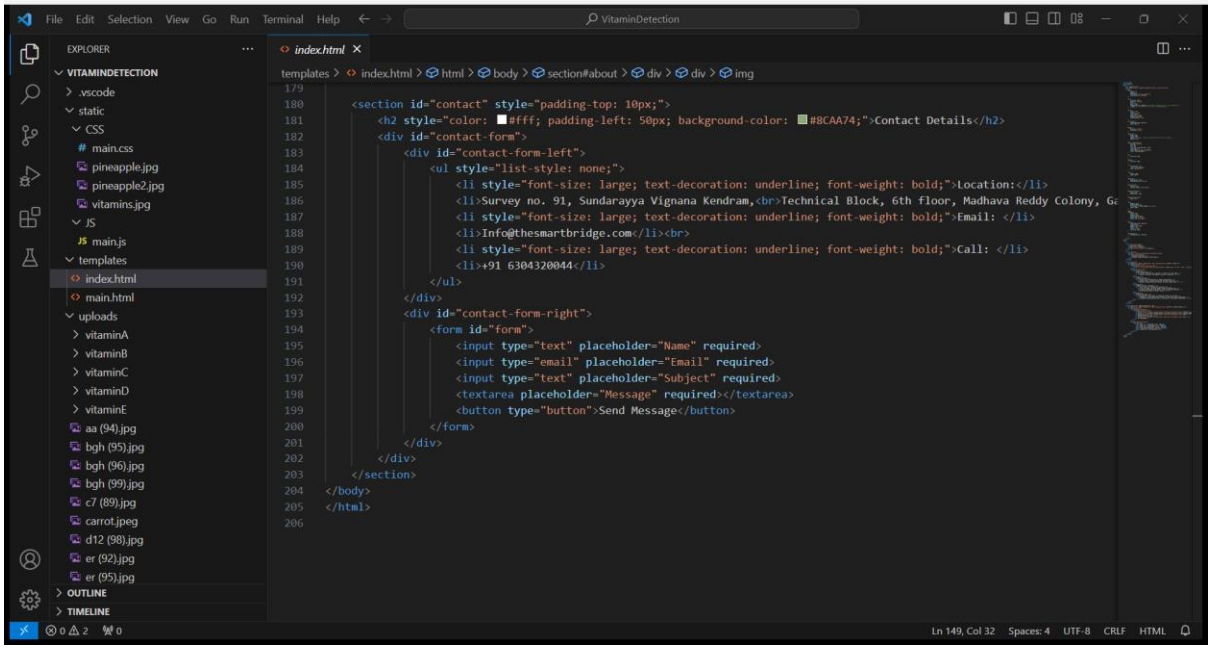

File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
index.html
templates > index.html > html > body > section#about > div > div > img
<h3>Model Building:</h3>
<ul><li>Choose a CNN architecture for image classification.</li>
<li>Design layers, specifying types and activation functions.</li>
<li>Define an output layer with neurons for each vitamin class.</li></ul>

<h3>Model Training & Evaluation:</h3>
<ul><li>Split data into training and validation sets.</li>
<li>Train the model, adjusting weights based on computed loss.</li>
<li>Evaluate performance using metrics like accuracy and precision.</li>
<li>Fine-tune the model for better results.</li></ul>

<h3>Model Deployment:</h3>
<ul><li>Deploy the trained model for detecting vitamins in new data.</li>
<li>Integrate it into applications/systems for real-time or batch processing.</li>
<li>Implement post-processing steps or interfaces for actionable predictions.</li></ul>

</section>
<section id="contact" style="padding-top: 10px;">
<h2 style="color: #fff; padding-left: 50px; background-color: #8CAA74;">Contact Details</h2>

<ul style="list-style: none;">
<li style="font-size: large; text-decoration: underline; font-weight: bold;">Location:</li>
<li>Survey no. 91, Sundarayya Vignana Kendram,<br>Technical Block, 6th floor, Madhava Reddy Colony, Ga
<li style="font-size: large; text-decoration: underline; font-weight: bold;">Email: </li>
<li>Info@thesmartbridge.com</li><br>
<li style="font-size: large; text-decoration: underline; font-weight: bold;">Call: </li>
<li>+91 6304320044</li>
Ln 149, Col 32 Spaces: 4 UTF-8 CRLF HTML

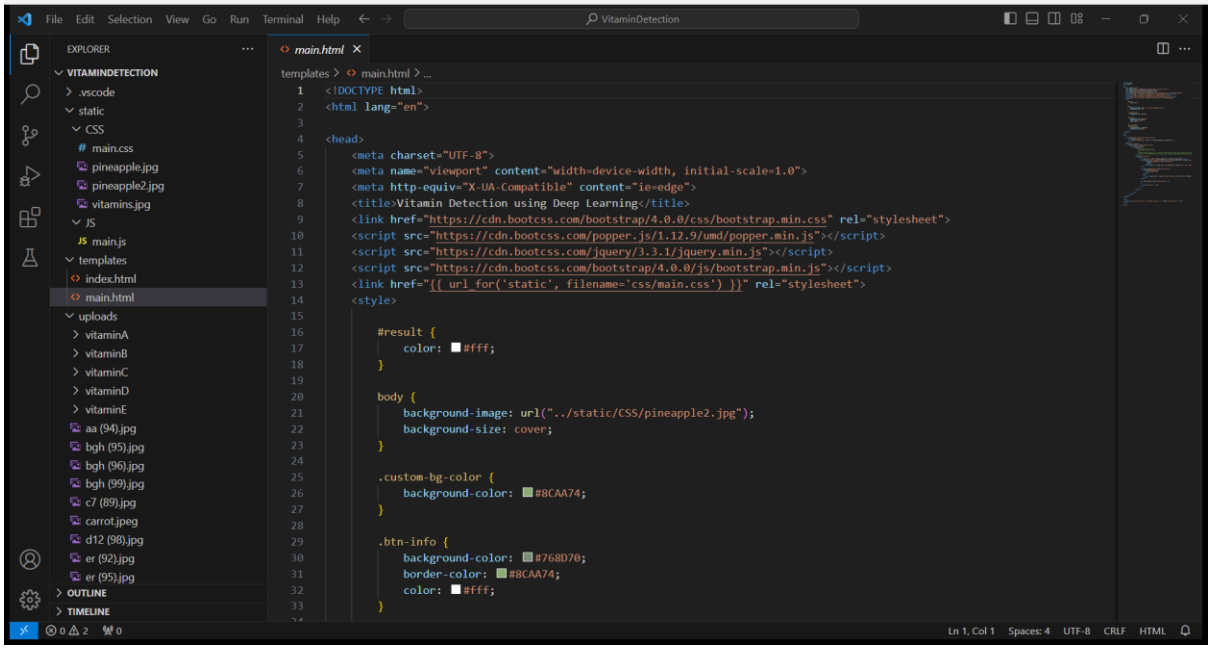

File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
index.html
templates > index.html > html > body > section#about > div > div > img
<section id="contact" style="padding-top: 10px;">
<h2 style="color: #fff; padding-left: 50px; background-color: #8CAA74;">Contact Details</h2>

<ul style="list-style: none;">
<li style="font-size: large; text-decoration: underline; font-weight: bold;">Location:</li>
<li>Survey no. 91, Sundarayya Vignana Kendram,<br>Technical Block, 6th floor, Madhava Reddy Colony, Ga
<li style="font-size: large; text-decoration: underline; font-weight: bold;">Email: </li>
<li>Info@thesmartbridge.com</li><br>
<li style="font-size: large; text-decoration: underline; font-weight: bold;">Call: </li>
<li>+91 6304320044</li>
</ul>

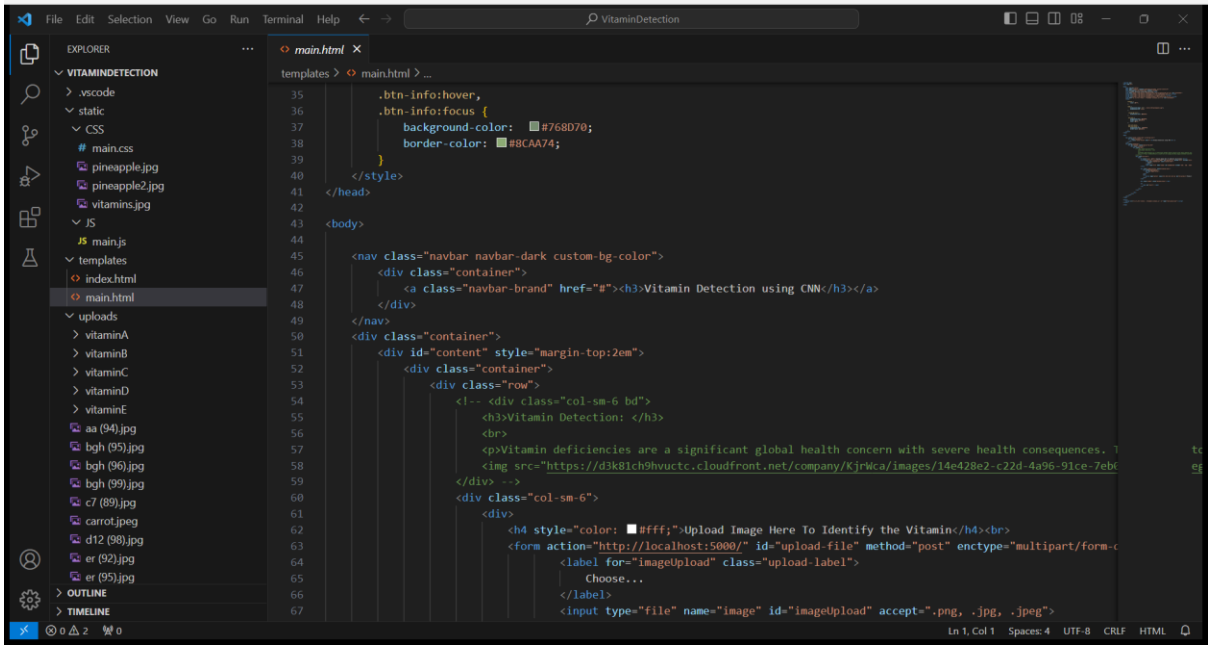
<form id="form">
<input type="text" placeholder="Name" required>
<input type="email" placeholder="Email" required>
<input type="text" placeholder="Subject" required>
<textarea placeholder="Message" required></textarea>
<button type="button">Send Message</button>
</form>

</section>
</body>
</html>
Ln 149, Col 32 Spaces: 4 UTF-8 CRLF HTML

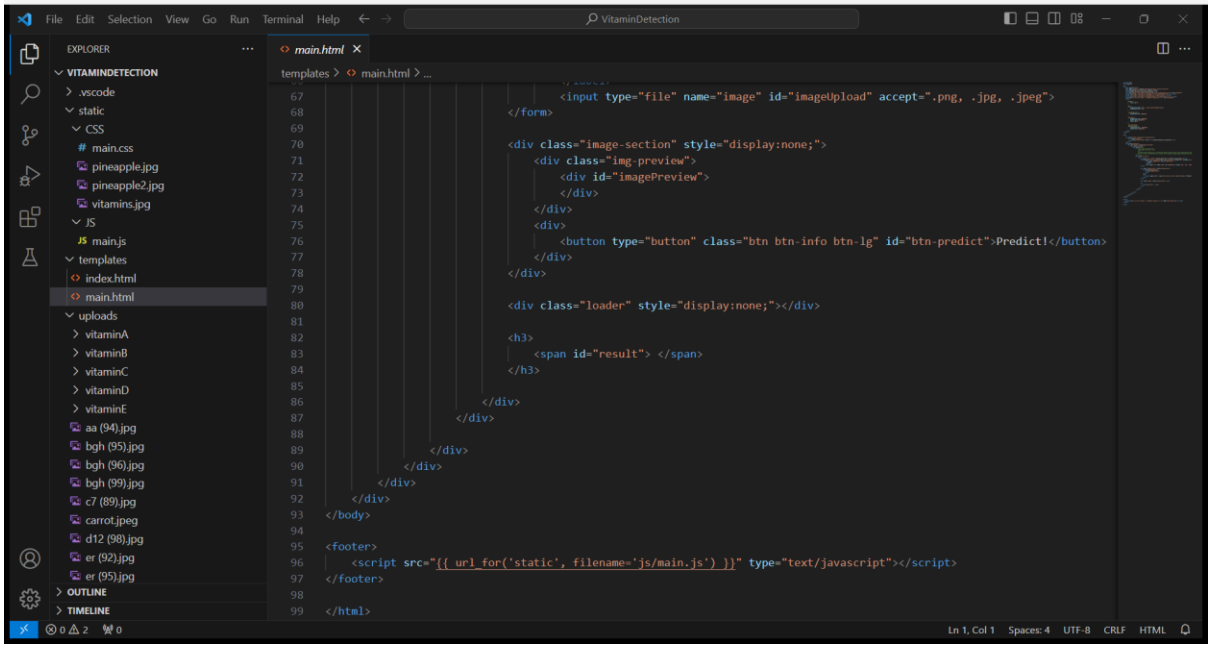
## Main.html page:



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Vitamin Detection using Deep Learning</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
    <style>

        #result {
            color: #fff;
        }

        body {
            background-image: url("../static/CSS/pineapple2.jpg");
            background-size: cover;
        }

        .custom-bg-color {
            background-color: #8CAA74;
        }

        .btn-info {
            background-color: #768D70;
            border-color: #8CAA74;
            color: #fff;
        }
```



```
        .btn-info:hover,
        .btn-info:focus {
            background-color: #768D70;
            border-color: #8CAA74;
        }
    </style>
</head>

<body>

    <nav class="navbar navbar-dark custom-bg-color">
        <div class="container">
            <a class="navbar-brand" href="#"><h3>Vitamin Detection using CNN</h3></a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">
            <div class="container">
                <div class="row">
                    <!-- <div class="col-sm-6 bd">
                        <h3>Vitamin Detection: </h3>
                        <br>
                        <p>Vitamin deficiencies are a significant global health concern with severe health consequences.
                        <img src="https://d3k81ch9hvuctc.cloudfront.net/company/KjrWca/images/14e428e2-c22d-4a96-91ce-7eb
                    </div> -->
                    <div class="col-sm-6">
                        <div>
                            <h4 style="color: #fff;">Upload Image Here To Identify the Vitamin</h4><br>
                            <form action="http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-
                                <label for="imageUpload" class="upload-label">
                                    Choose...
                                </label>
                                <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
```


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
main.html
templates > main.html > ...
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
<input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
</form>

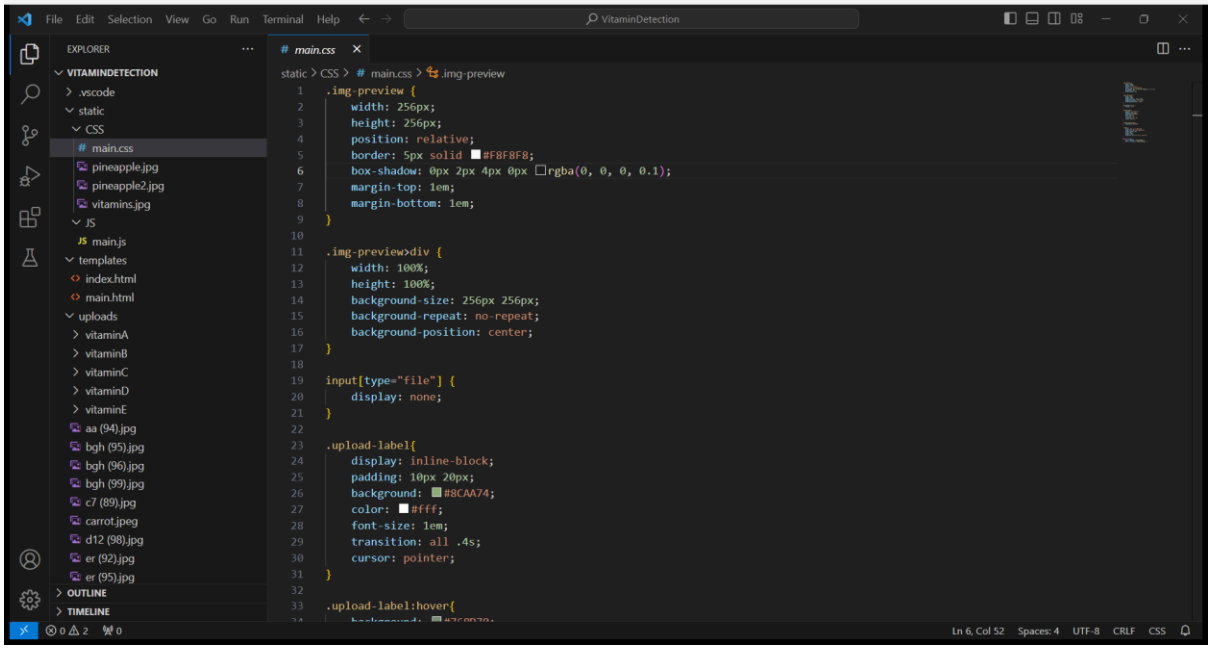
<button type="button" class="btn btn-info btn-lg" id="btn-predict">Predict!</button>

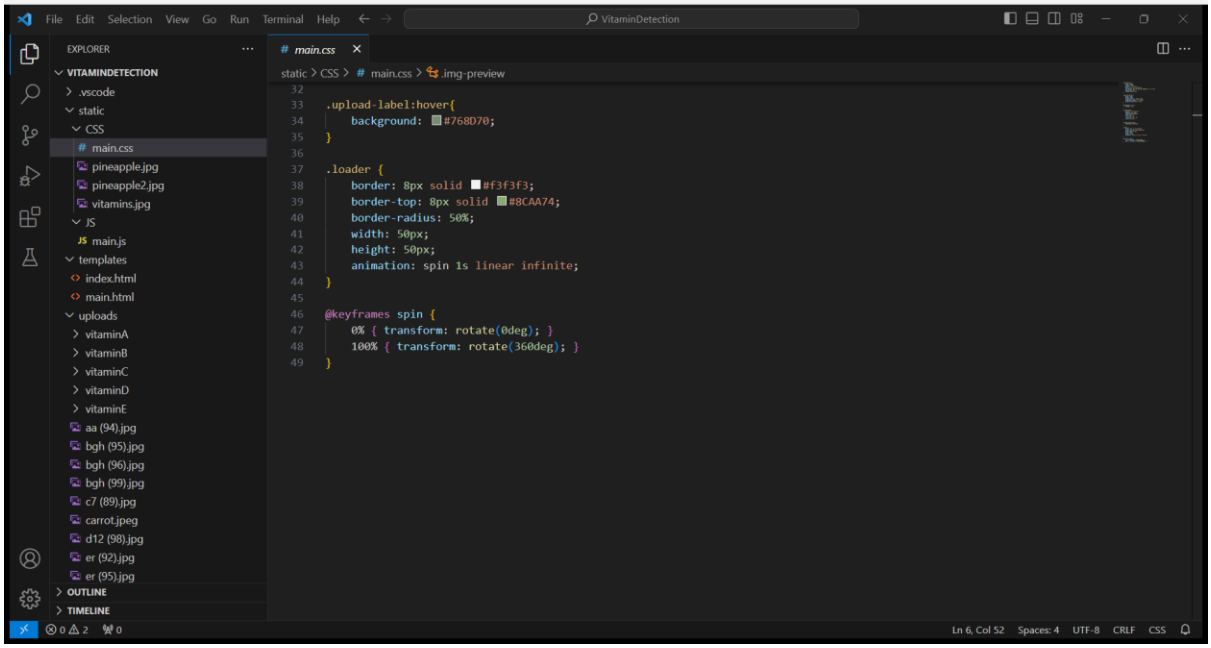
<h3>

</h3>

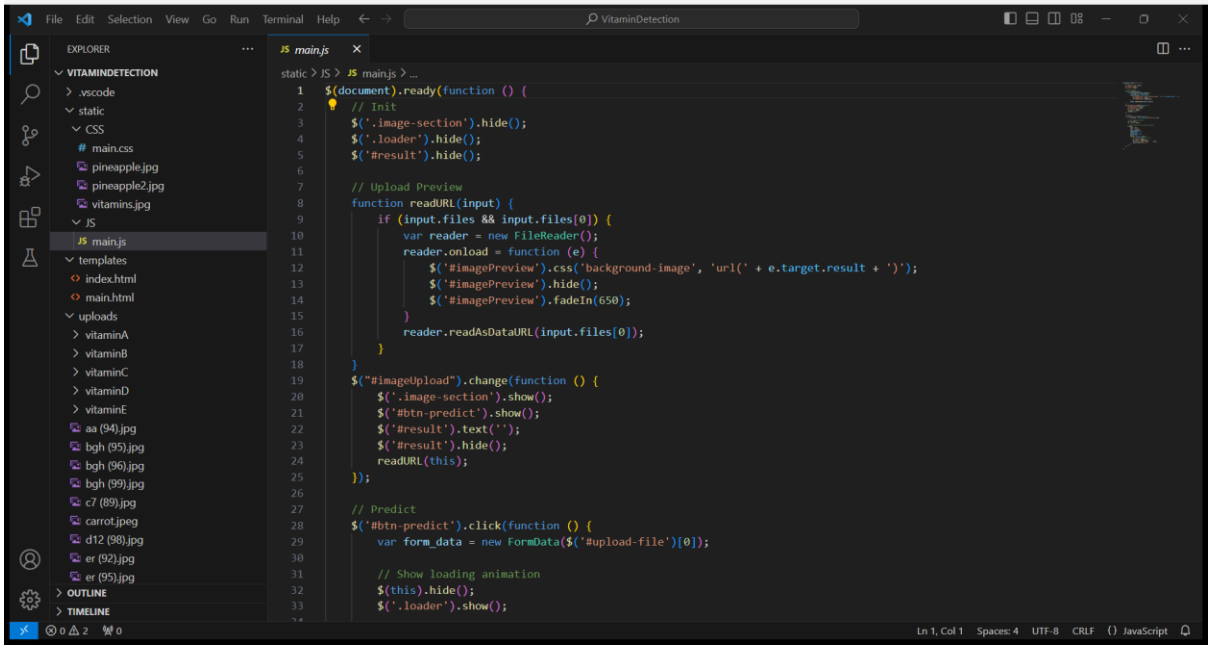
</body>
<footer>
<script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>
</html>
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML

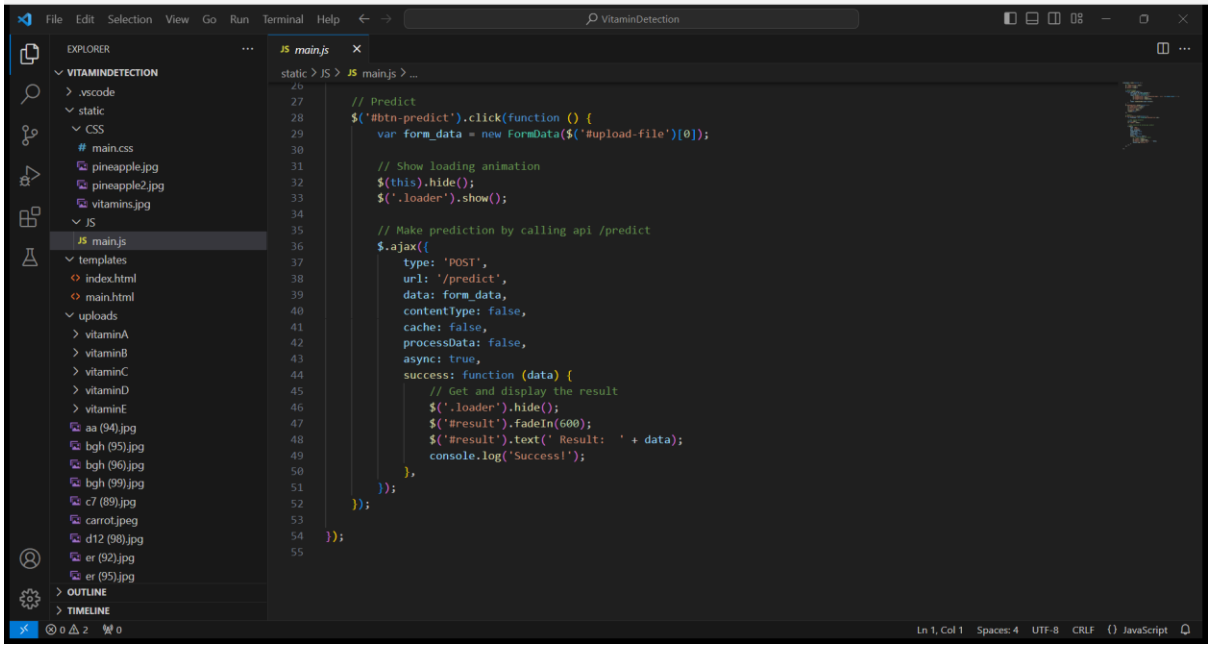
## CSS code:


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
main.css
static > CSS > main.css > .img-preview
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
.img-preview {
width: 256px;
height: 256px;
position: relative;
border: 5px solid #F8F8F8;
box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
margin-top: 1em;
margin-bottom: 1em;
}
.img-preview>div {
width: 100%;
height: 100%;
background-size: 256px 256px;
background-repeat: no-repeat;
background-position: center;
}
input[type="file"] {
display: none;
}
.upload-label{
display: inline-block;
padding: 10px 20px;
background: #8CAA74;
color: #fff;
font-size: 1em;
transition: all .4s;
cursor: pointer;
}
.upload-label:hover{
Ln 6, Col 52 Spaces: 4 UTF-8 CRLF CSS

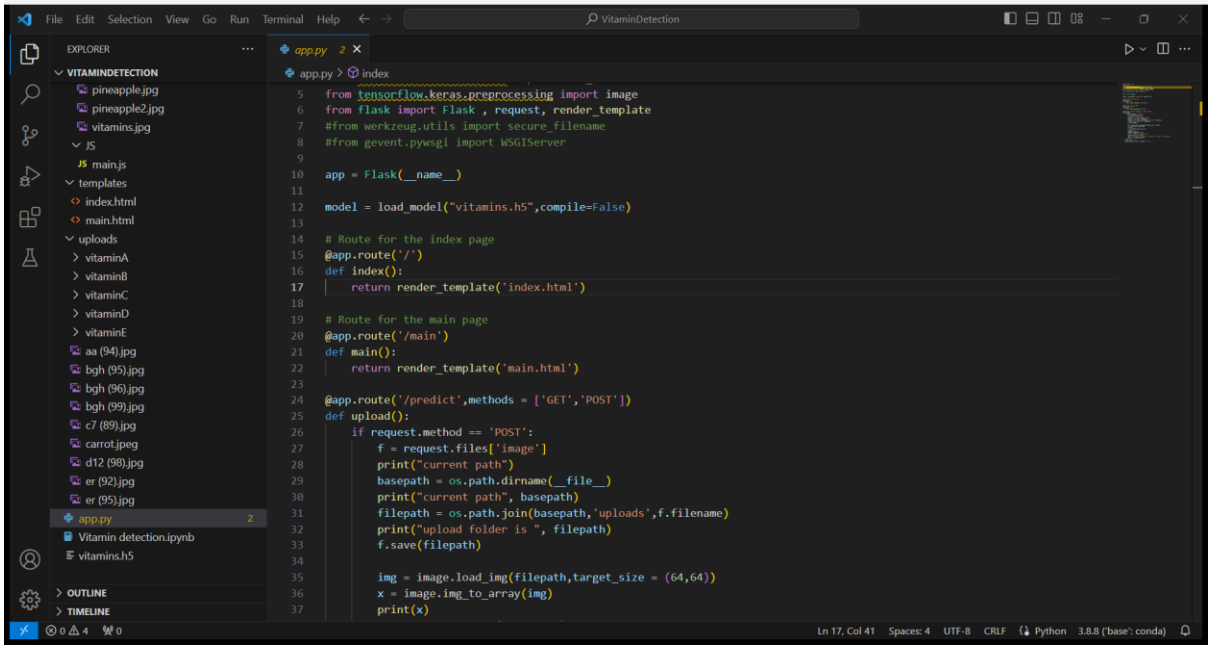

File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
main.css
static > CSS > main.css > .img-preview
.upload-label:hover{
background: #768D70;
}
.loader {
border: 8px solid #f3f3f3;
border-top: 8px solid #8CAA74;
border-radius: 50%;
width: 50px;
height: 50px;
animation: spin 1s linear infinite;
}
@keyframes spin {
0% { transform: rotate(0deg); }
100% { transform: rotate(360deg); }
}
Ln 6, Col 52 Spaces: 4 UTF-8 CRLF CSS

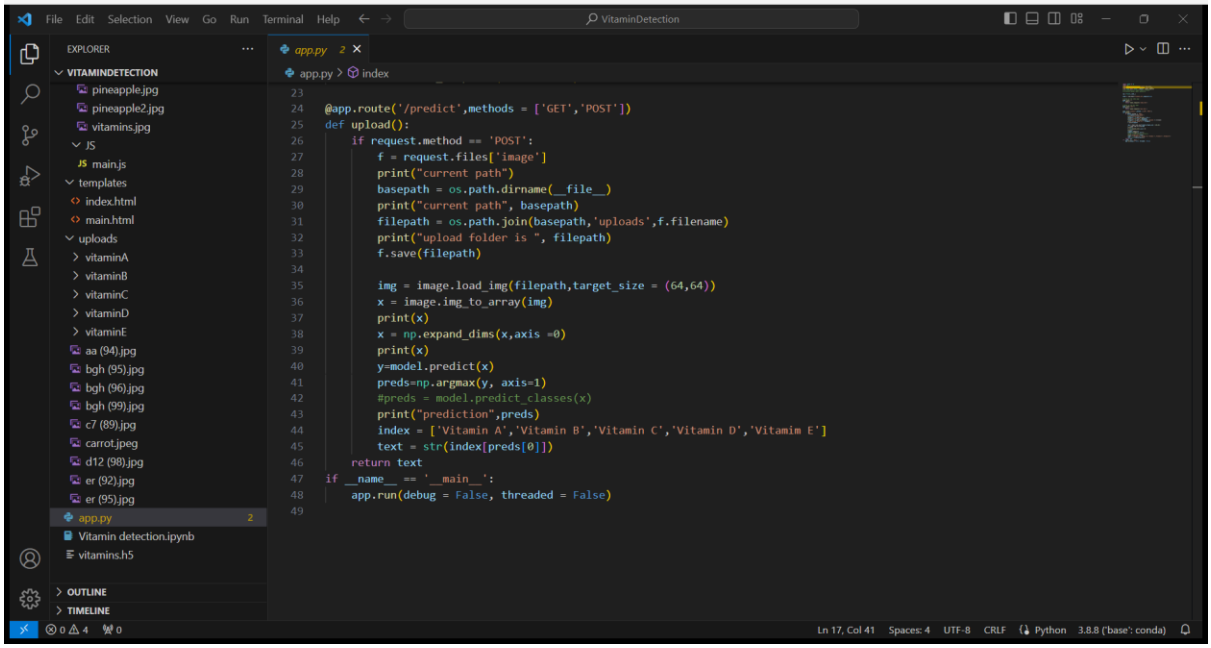
## Main.js file:


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
main.js
static > JS > main.js > ...
$(document).ready(function () {
// Init
$('.image-section').hide();
$('.loader').hide();
$('#result').hide();
// Upload Preview
function readURL(input) {
if (input.files && input.files[0]) {
var reader = new FileReader();
reader.onload = function (e) {
$('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
$('#imagePreview').hide();
$('#imagePreview').fadeIn(650);
}
reader.readAsDataURL(input.files[0]);
}
}
$("#imageUpload").change(function () {
$('.image-section').show();
$('#btn-predict').show();
$('#result').text('');
$('#result').hide();
readURL(this);
});
// Predict
$('#btn-predict').click(function () {
var form_data = new FormData($('#upload-file')[0]);
// Show loading animation
$(this).hide();
$('.loader').show();
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JavaScript

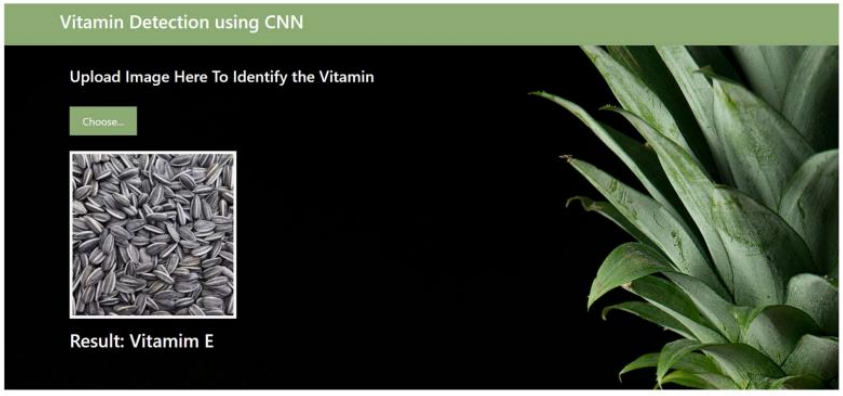

File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
.vscode
static
CSS
main.css
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
OUTLINE
TIMELINE
main.js
static > JS > main.js > ...
// Predict
$('#btn-predict').click(function () {
var form_data = new FormData($('#upload-file')[0]);
// Show loading animation
$(this).hide();
$('.loader').show();
// Make prediction by calling api /predict
$.ajax({
type: 'POST',
url: '/predict',
data: form_data,
contentType: false,
cache: false,
processData: false,
async: true,
success: function (data) {
// Get and display the result
$('.loader').hide();
$('#result').fadeIn(600);
$('#result').text(' Result: ' + data);
console.log('Success!');
},
});
});
});
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript

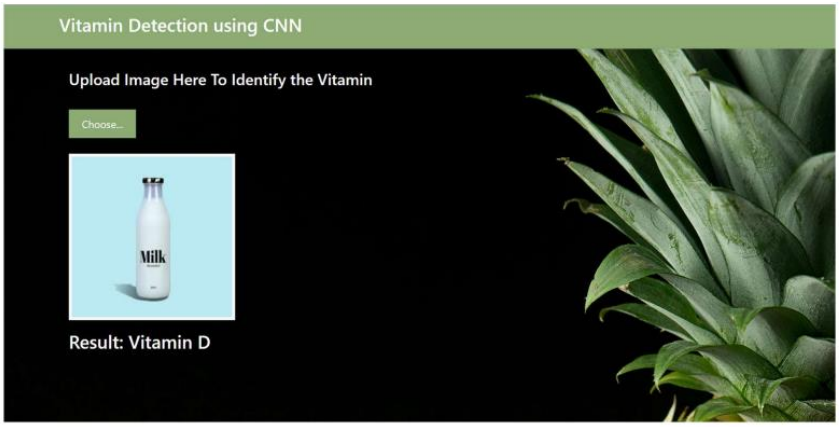
## App.py file:

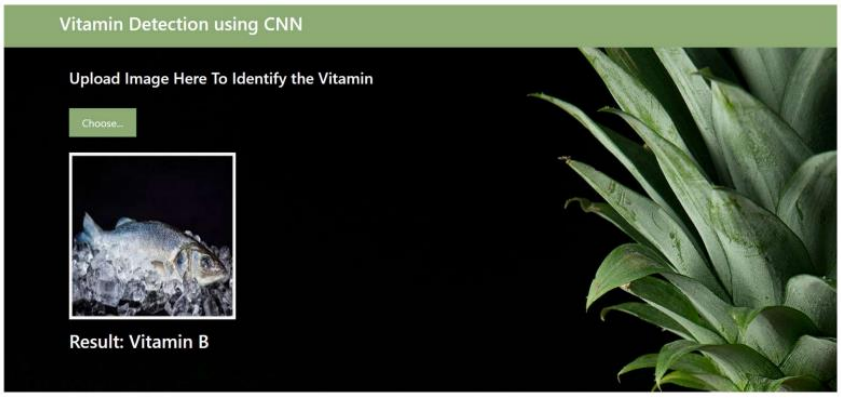

File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
app.py 2
Vitamin detection.ipynb
vitamins.h5
OUTLINE
TIMELINE
app.py
app.py > index
from tensorflow.keras.preprocessing import image
from flask import Flask , request, render_template
#from werkzeug.utils import secure_filename
#from gevent.pywsgi import WSGIServer
app = Flask(__name__)
model = load_model("vitamins.h5",compile=False)
# Route for the index page
@app.route('/')
def index():
return render_template('index.html')
# Route for the main page
@app.route('/main')
def main():
return render_template('main.html')
@app.route('/predict',methods = ['GET','POST'])
def upload():
if request.method == 'POST':
f = request.files['image']
print("current path")
basepath = os.path.dirname(__file__)
print("current path", basepath)
filepath = os.path.join(basepath,'uploads',f.filename)
print("upload folder is ", filepath)
f.save(filepath)
img = image.load_img(filepath,target_size = (64,64))
x = image.img_to_array(img)
print(x)
Ln 17, Col 41 Spaces: 4 UTF-8 CRLF Python 3.8.8 ('base': conda)


File Edit Selection View Go Run Terminal Help
VitaminDetection
EXPLORER
VITAMINDETECTION
pineapple.jpg
pineapple2.jpg
vitamins.jpg
JS
main.js
templates
index.html
main.html
uploads
vitaminA
vitaminB
vitaminC
vitaminD
vitaminE
aa (94).jpg
bgh (95).jpg
bgh (96).jpg
bgh (99).jpg
c7 (89).jpg
carrot.jpeg
d12 (98).jpg
er (92).jpg
er (95).jpg
app.py
Vitamin detection.ipynb
vitamins.h5
OUTLINE
TIMELINE
app.py 2
app.py > index
@app.route('/predict',methods = ['GET','POST'])
def upload():
if request.method == 'POST':
f = request.files['image']
print("current path")
basepath = os.path.dirname(__file__)
print("current path", basepath)
filepath = os.path.join(basepath,'uploads',f.filename)
print("upload folder is ", filepath)
f.save(filepath)
img = image.load_img(filepath,target_size = (64,64))
x = image.img_to_array(img)
print(x)
x = np.expand_dims(x,axis =0)
print(x)
y=model.predict(x)
preds=np.argmax(y, axis=1)
#preds = model.predict_classes(x)
print("prediction",preds)
index = ['Vitamin A','Vitamin B','Vitamin C','Vitamin D','Vitamim E']
text = str(index[preds[0]])
return text
if __name__ == '__main__':
app.run(debug = False, threaded = False)
Ln 17, Col 41 Spaces: 4 UTF-8 CRLF Python 3.8.8 ('base': conda)

## Testing:


Vitamin Detection using CNN
Upload Image Here To Identify the Vitamin
Choose...
Result: Vitamim E


Vitamin Detection using CNN
Upload Image Here To Identify the Vitamin
Choose...
Milk
Result: Vitamin D




**Performance & Final Submission Phase**
**Model Performance Test**

| | |
|---|---|
| Date | 15 November 2023 |
| Team ID | Team-592065 |
| Project Name | Vitamin Detection using Deep Learning |
| Maximum Marks | 10 Marks |

**Model Performance Testing:**

Project team shall fill the following information in model performance testing template.

| S.No. | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Metrics | **Regression Model:**<br>MAE - , MSE - , RMSE - , R2 score -<br><br>**Classification Model:**<br>Confusion Matrix - , Accuray Score- & Classification Report - | `# Confusion Matrix`<br>`cm = confusion_matrix(y_true_labels, y_pred_labels)`<br>`print("Confusion Matrix:\n", cm)`<br><br>Confusion Matrix:<br>[[ 4 11 22 4 11]<br>[11 7 12 4 8]<br>[11 15 16 8 12]<br>[ 3 5 7 7 6]<br>[ 4 12 8 6 10]]<br><br>`# Accuracy Score`<br>`accuracy = accuracy_score(y_true_labels, y_pred_labels)`<br>`print("Accuracy Score:", accuracy)`<br><br>Accuracy Score: 0.19642857142857142<br><br>`# Classification Report`<br>`report = classification_report(y_true_labels, y_pred_labels)`<br>`print("Classification Report:\n", report)`<br><br>Classification Report:<br>precision recall f1-score support<br><br>0 0.12 0.08 0.09 52<br>1 0.14 0.17 0.15 42<br>2 0.25 0.26 0.25 62<br>3 0.24 0.25 0.25 28<br>4 0.21 0.25 0.23 40<br><br>accuracy 0.20 224<br>macro avg 0.19 0.20 0.19 224<br>weighted avg 0.19 0.20 0.19 224 |
| 2. | Tune the Model | Hyperparameter Tuning - Validation Method - CNN | `model=Sequential()`<br>`model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation="relu"))`<br>`model.add(MaxPooling2D(pool_size=(2,2)))`<br>`model.add(Flatten())`<br>`model.add(Dense(units=128,activation="relu"))`<br>`model.add(Dense(units=5,activation="softmax"))` |

**Advantages:**

- CNNs enable automated analysis of food images for vitamin content.
- CNNs can achieve high levels of accuracy and precision in vitamin detection.
- CNNs can be trained to adapt to various types of food items and cuisines.
- CNNs can efficiently process large datasets of food images.
- CNNs can be optimized for real-time processing of food images.

**Disadvantages:**

- CNNs are highly dependent on the quality and quantity of training data. If the dataset is not representative or lacks diversity, the model may struggle to generalize to various foods and conditions.
- CNNs, particularly deep and complex models, often lack interpretability. Understanding how the model arrives at a particular prediction can be challenging.
- CNNs, especially when dealing with relatively small datasets, are prone to overfitting. The model may memorize the training data instead of learning general patterns.
- Training deep CNNs effectively often requires large amounts of labelled data, which may not always be readily available, especially for specific food types or cultural dishes.
- CNNs may struggle with variations in food presentation, such as different cooking styles, serving sizes, or ingredient arrangements.

**Conclusion:**

In conclusion, vitamin detection using Convolutional Neural Networks (CNNs) presents a promising avenue for revolutionizing nutritional analysis. While CNNs offer significant advantages, including automation and adaptability, certain challenges exist, such as data dependency, interpretability issues, and the need for diverse datasets. Despite these limitations, ongoing advancements in machine learning and deep learning techniques, coupled with research efforts, hold the potential to overcome these challenges. The application of CNNs in vitamin detection aligns with the broader goal of harnessing technology to enhance health outcomes, providing a foundation for more accurate and efficient methods of nutritional assessment. As the field continues to evolve, addressing these challenges will be crucial to unlock the full potential of CNNs in improving our understanding of the nutritional content of food.

**Future scope:**

The future scope of vitamin detection using Convolutional Neural Networks (CNNs) holds great promise, driven by advancements in technology, data availability, and machine learning methodologies. Here is a concise overview of the potential future directions in this field:

1. **Improvements in Model Generalization:**
    - **Scope:** Enhancing CNN architectures to improve generalization across diverse food types and cultural cuisines.
    - **Impact:** Models with increased adaptability will yield more accurate and reliable vitamin detection across a wide range of dietary patterns.
2. **Integration of Multimodal Data:**
    - **Scope:** Incorporating additional data modalities, such as nutritional databases, text-based information, or even user feedback, to augment CNN-based vitamin detection.
    - **Impact:** A holistic approach leveraging multiple data sources can enhance the overall accuracy and context-awareness of the detection system.
3. **Explainable AI for Enhanced Trust:**
    - **Scope:** Advancing techniques for model interpretability and explainability in CNNs used for vitamin detection.
    - **Impact:** Increased transparency will enhance user trust, facilitate regulatory compliance, and encourage wider adoption in healthcare and nutrition-related applications.
4. **Transfer Learning for Limited Data Environments:**
    - **Scope:** Developing robust transfer learning strategies to address challenges associated with limited labeled data for vitamin-rich foods.
    - **Impact:** Effective transfer learning will enable the deployment of vitamin detection models in diverse contexts, including regions with specific dietary preferences.
5. **Real-time and Edge Computing Applications:**
    - **Scope:** Optimizing CNN models for real-time processing and deployment on edge devices.
    - **Impact:** This will facilitate on-the-spot vitamin analysis, catering to applications in smart kitchen appliances, mobile health apps, and other real-time dietary assessment tools.
6. **Personalized Nutrition Recommendations:**
    - **Scope:** Integrating vitamin detection results into personalized nutrition recommendations based on individual health profiles and dietary preferences.
    - **Impact:** Tailored dietary advice will empower individuals to make informed choices, contributing to overall health and wellness.

7. **Cross-disciplinary Collaboration:**
    - **Scope:** Collaborations between experts in nutrition, computer science, and data science to refine models and datasets.
    - **Impact:** Cross-disciplinary efforts will ensure a more comprehensive understanding of nutritional requirements and effective implementation of AI-based vitamin detection.
8. **User-Friendly Interfaces and Mobile Applications:**
    - **Scope:** Designing intuitive and user-friendly interfaces, possibly in the form of mobile applications, for seamless integration into users' daily routines.
    - **Impact:** Accessible and user-friendly applications will encourage widespread adoption and engagement with vitamin detection tools.
9. **Continuous Learning and Adaptation:**
    - **Scope:** Implementing mechanisms for continuous learning and model adaptation based on real-world feedback and evolving dietary patterns.
    - **Impact:** This ensures that vitamin detection models remain relevant and accurate over time, adapting to changing nutritional trends.
10. **Addressing Ethical and Privacy Considerations:**
    - **Scope:** Developing robust ethical frameworks and privacy-preserving methodologies for handling health-related data in vitamin detection applications.
    - **Impact:** Ensuring ethical practices will enhance user confidence, compliance with regulations, and the responsible deployment of vitamin detection systems.

The future of vitamin detection using CNNs is poised to revolutionize nutritional assessment, offering innovative solutions that are more accurate, adaptable, and user-centric.