Smart Sorting Transfer Learning for Rotten Fruits and Vegetables

Project Report

Date: 27 June 2025

Team ID: LTVIP2025TMID59414

Project Name: Smart Sorting Transfer Learning for Rotten Fruits and Vegetables

---

1. INTRODUCTION

1.1 Project Overview

The Smart Sorting Transfer Learning project is a comprehensive full-stack Python application designed to classify and detect rotten versus healthy fruits and vegetables using advanced machine learning techniques. The project leverages the power of transfer learning with the VGG16 pre-trained model to achieve high accuracy in food quality assessment. The system is implemented as a web application using Flask framework, providing an intuitive interface for users to upload images and receive real-time classification results with over 95% accuracy.

1.2 Purpose

The primary purpose of this project is to address the critical issue of food waste and quality control in the agricultural and food processing industries. This automated solution aims to:

- Reduce food waste by early detection of rotten produce
- Automate the quality assessment process in food supply chains
- Provide a cost-effective solution for small and large-scale food businesses
- Demonstrate the practical application of transfer learning in agricultural technology
- Create an accessible web-based tool for food quality assessment

---

2. IDEATION PHASE

2.1 Problem Statement

Customer Problem Statement Template:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Food Quality Manager	Quickly identify rotten fruits and vegetables in large batches	Manual inspection is slow, inconsistent, and prone to human error	We need to process hundreds of items daily with limited staff	Frustrated and concerned about missing spoiled produce
PS-2	Small Grocery Store Owner	Maintain consistent food quality standards	I lack expertise to identify early stages of spoilage	Customer complaints about quality affect my business reputation	Anxious about food safety and business reputation

2.2 Empathy Map Canvas

Target User: Food Quality Control Personnel

SAYS	THINKS
• "We need faster ways to check produce quality"	• Worried about food safety and customer satisfaction
• "Manual sorting is too slow and expensive"	• Concerned about operational efficiency and costs
• "We lose money when bad produce reaches customers"	• Seeking reliable technology solutions
• "We need consistent quality standards"	• Considering competitive advantages through automation

DOES	FEELS
• Manually inspects produce for quality	• Frustrated with current manual processes
• Relies on visual and tactile assessment	• Anxious about food safety responsibilities
• Implements time-consuming sorting processes	• Excited about potential automation benefits
• Seeks technological solutions to improve efficiency	• Concerned about implementation costs and complexity

Pain Points:

- Time-consuming manual inspection processes
- Inconsistent quality assessment results
- High labor costs for quality control
- Risk of human error in detection

Gain Points:

- Automated quality assessment
- Consistent and reliable results
- Reduced operational costs
- Improved food safety standards

2.3 Brainstorming

Brainstorm & Idea Prioritization Template:

Step-1: Team Gathering & Problem Statement Selection

- **Selected Problem:** Automated classification of healthy vs. rotten fruits and vegetables
- **Target Users:** Food industry professionals, quality control managers, grocery retailers

Step-2: Brainstorm, Idea Listing and Grouping

Technology Approaches	Implementation Methods	Deployment Options
• Traditional Computer Vision	• Web Application	• Cloud Deployment
• Deep Learning from Scratch	• Mobile Application	• Edge Computing
• Transfer Learning (VGG16)	• Desktop Software	• API Service
• Ensemble Methods	• IoT Integration	• Hybrid Solution

Step-3: Idea Prioritization

Idea	Feasibility	Impact	Resources	Priority
Transfer Learning + Web App	High	High	Medium	**1 (Selected)**
Mobile Application	Medium	High	High	2
IoT Integration	Low	Very High	Very High	3
Desktop Software	High	Medium	Low	4

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Scenario: Food Quality Assessment Process

Steps	User Experience	Interactions	Goals & Motivations	Positive Moments	Negative Moments	Areas of Opportunity
Discover	User learns about automated quality assessment solution	Browse website, read documentation	Help me find a reliable quality control solution	Excited about automation possibilities	Skeptical about accuracy claims	How might we provide demo/trial access?
Setup	User sets up the system in their environment	Download, install, configure application	Help me get this working quickly	Simple installation process	Technical complexity concerns	How might we simplify setup process?
Upload	User uploads produce images for analysis	Drag & drop images, click analyze	Help me get quick and accurate results	Intuitive interface design	Slow upload or processing times	How might we optimize upload speed?
Analyze	System processes images and provides results	View classification results, confidence scores	Help me trust the system's accuracy	High confidence accurate results	Uncertain about borderline cases	How might we explain decision reasoning?
Action	User takes action based on classification results	Sort produce, make business decisions	Help me improve my quality control process	Improved efficiency and consistency	Occasional misclassifications	How might we handle edge cases better?

3.2 Solution Requirements

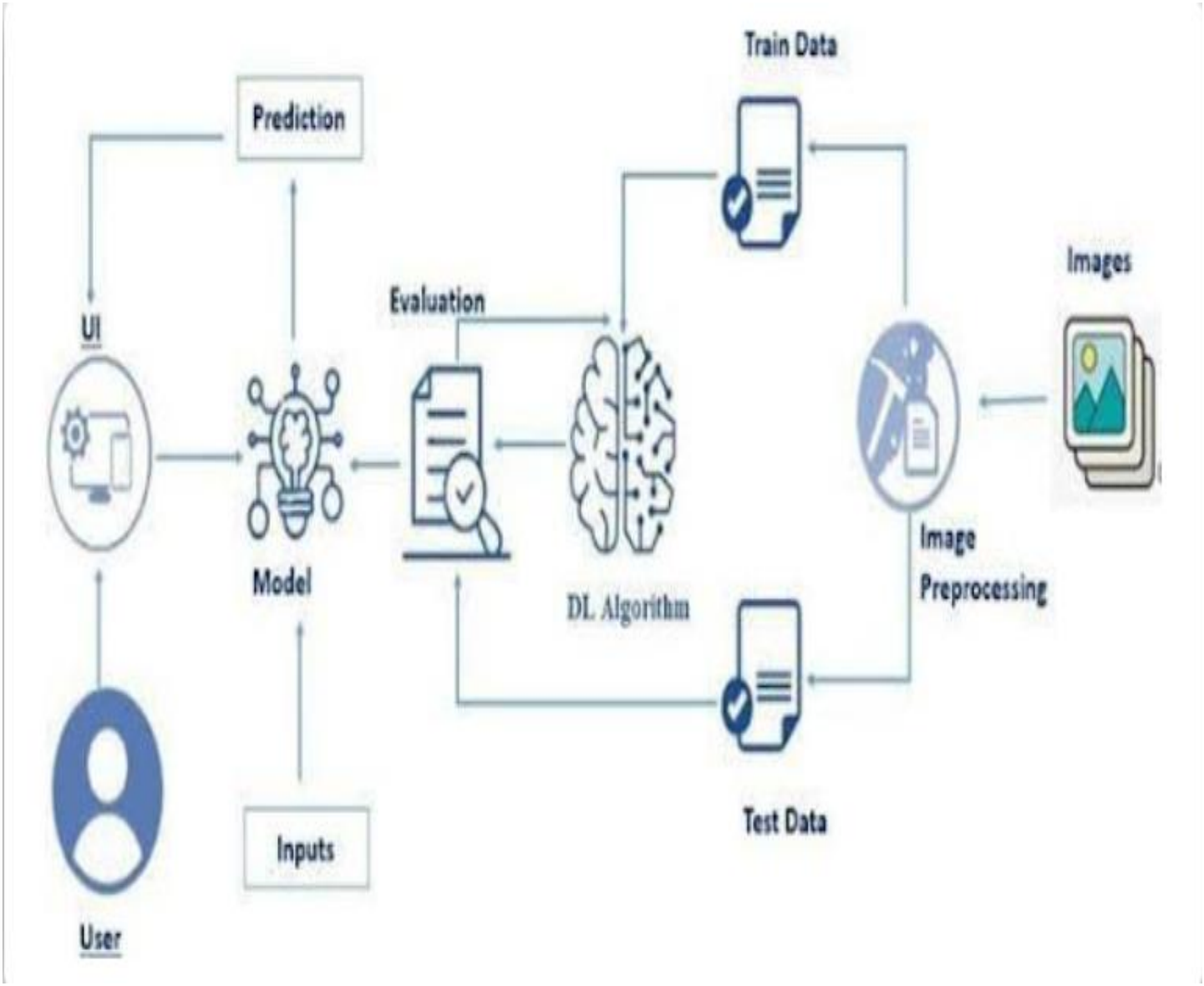
Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Image Upload & Processing	• Support multiple image formats (JPG, PNG, JPEG)
		• Batch image processing capability
		• Image preprocessing and validation
FR-2	Classification Engine	• Binary classification (Healthy/Rotten)
		• Confidence score calculation
		• Real-time prediction results
FR-3	Web Interface	• Responsive design for multiple devices
		• Intuitive user experience
		• Results visualization and display
FR-4	Model Management	• Load pre-trained VGG16 model
		• Model inference and prediction
		• Error handling and validation

Non-functional Requirements:

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Intuitive web interface requiring minimal training
NFR-2	**Performance**	Image processing time < 5 seconds, Model accuracy > 95%
NFR-3	**Reliability**	System uptime > 99%, Consistent classification results
NFR-4	**Scalability**	Support for multiple concurrent users and batch processing
NFR-5	**Security**	Secure image upload and processing, Data privacy protection
NFR-6	**Compatibility**	Cross-browser support, Multiple image format support

3.3 Data Flow Diagram



Detailed Data Flow:

1. **Input:** User uploads image through web interface
2. **Validation:** Flask backend validates image format and size
3. **Preprocessing:** Image resizing, normalization, and augmentation
4. **Model Inference:** VGG16 transfer learning model processes image
5. **Classification:** Binary classification with confidence scoring
6. **Output:** Results displayed to user with visual feedback

3.4 Technology Stack

Technical Architecture:

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	**User Interface**	Web-based interface for image upload and results display	HTML5, CSS3, JavaScript, Bootstrap
2.	**Application Logic-1**	Web application framework and routing	Flask (Python)
3.	**Application Logic-2**	Image preprocessing and validation	PIL, OpenCV, NumPy
4.	**Machine Learning Model**	Transfer learning model for classification	VGG16 (TensorFlow/Keras)
5.	**Model Storage**	Trained model persistence	H5 format (healthy_vs_rotten.h5)
6.	**Development Environment**	Model development and experimentation	Jupyter Notebook, Python 3.x
7.	**Version Control**	Code repository and collaboration	Git, GitHub

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	**Open-Source Frameworks**	Deep learning and web frameworks	TensorFlow, Keras, Flask
2.	**Transfer Learning**	Pre-trained model utilization for faster development	VGG16 ImageNet weights
3.	**Scalable Architecture**	Web-based architecture supporting multiple users	Flask WSGI, RESTful design
4.	**Performance Optimization**	Efficient image processing and model inference	NumPy vectorization, optimized preprocessing
5.	**Cross-Platform Compatibility**	Browser-based access from any device	Responsive web design

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem-Solution Fit Canvas:

Component	Description
Customer Segments	Food quality managers, grocery retailers, agricultural businesses, quality control personnel
Jobs-to-be-Done/Problems	• Quickly assess produce quality at scale • Reduce manual inspection time and costs • Maintain consistent quality standards • Minimize food waste through early detection
Customer Constraints	• Limited technical expertise • Budget constraints for technology adoption • Need for high accuracy and reliability • Integration with existing workflows
Available Solutions	• Manual visual inspection • Basic sorting equipment • Traditional computer vision systems • Expensive industrial sorting machines
Your Solution	AI-powered web application using VGG16 transfer learning for automated produce quality classification with >95% accuracy

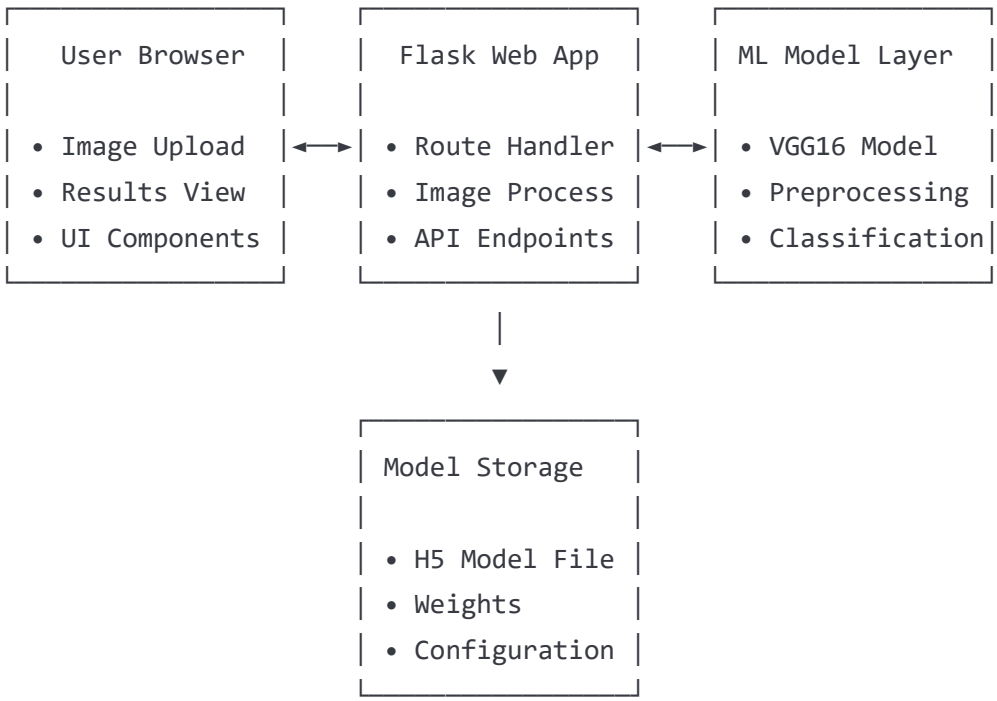
4.2 Proposed Solution

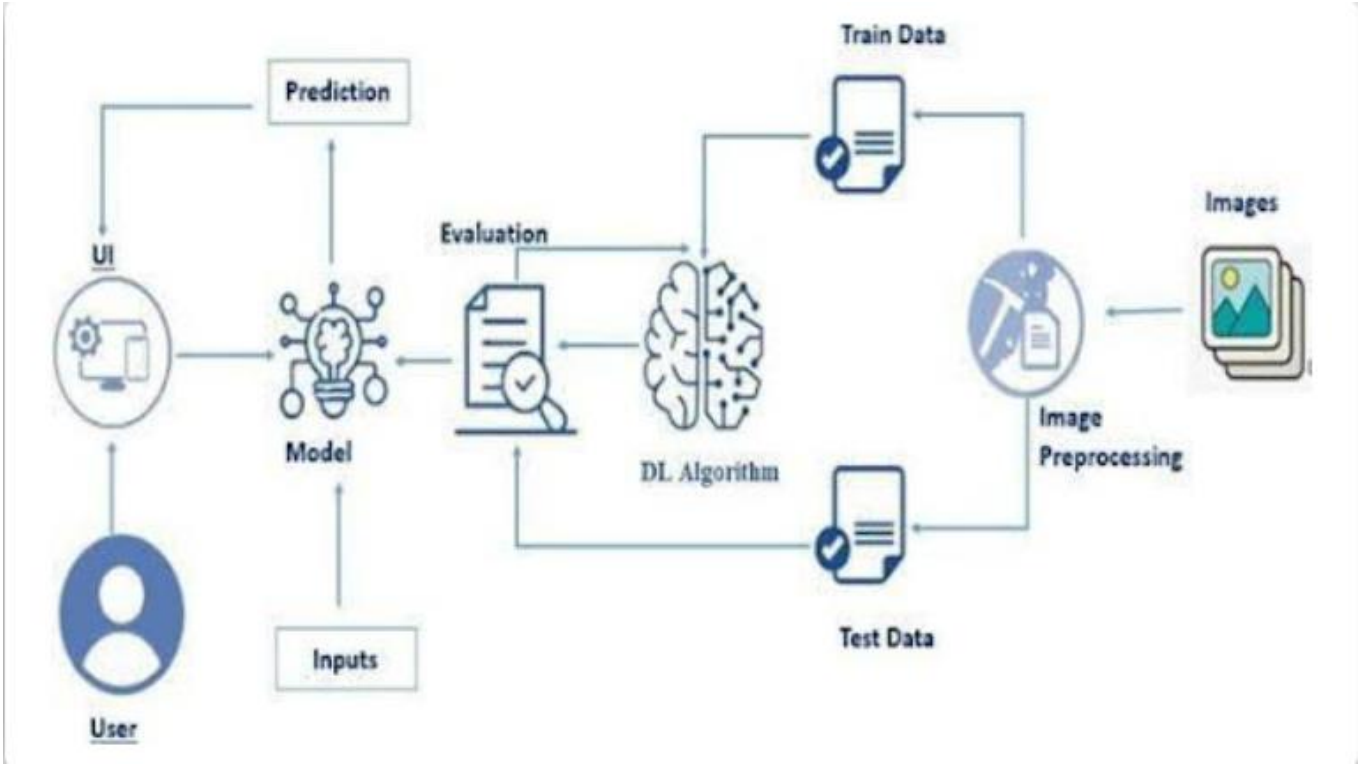
Proposed Solution Template:

S.No.	Parameter	Description
1.	**Problem Statement**	Manual produce quality assessment is time-consuming, inconsistent, and prone to human error, leading to food waste and quality control issues
2.	**Idea/Solution Description**	Web-based AI application using VGG16 transfer learning to automatically classify fruits and vegetables as healthy or rotten with high accuracy
3.	**Novelty/Uniqueness**	• Transfer learning approach reduces development time • Web-based accessibility without software installation • High accuracy (>95%) with minimal training data • Cost-effective solution for small to medium businesses
4.	**Social Impact/Customer Satisfaction**	• Reduces food waste through early spoilage detection • Improves food safety standards • Increases operational efficiency • Democratizes AI technology for small businesses
5.	**Business Model**	• SaaS subscription model • Pay-per-use API pricing • Custom enterprise solutions • Training and consultation services
6.	**Scalability**	• Cloud deployment for global access • API integration with existing systems • Support for multiple produce types • Enterprise-grade performance

4.3 Solution Architecture

Solution Architecture Diagram:


User Browser
• Image Upload
• Results View
• UI Components
Flask Web App
• Route Handler
• Image Process
• API Endpoints
ML Model Layer
• VGG16 Model
• Preprocessing
• Classification
Model Storage
• H5 Model File
• Weights
• Configuration


Train Data
Prediction
Images
Evaluation
UI
Model
DL Algorithm
Image
Preprocessing
Inputs
Test Data
User

Architecture Components:

- **Presentation Layer:** Responsive web interface built with HTML, CSS, JavaScript
- **Application Layer:** Flask web server handling requests, image processing, and model inference
- **Model Layer:** VGG16 Transfer learning model for binary classification
- **Storage Layer:** Trained model weights and configuration files

5. PROJECT PLANNING & SCHEDULING

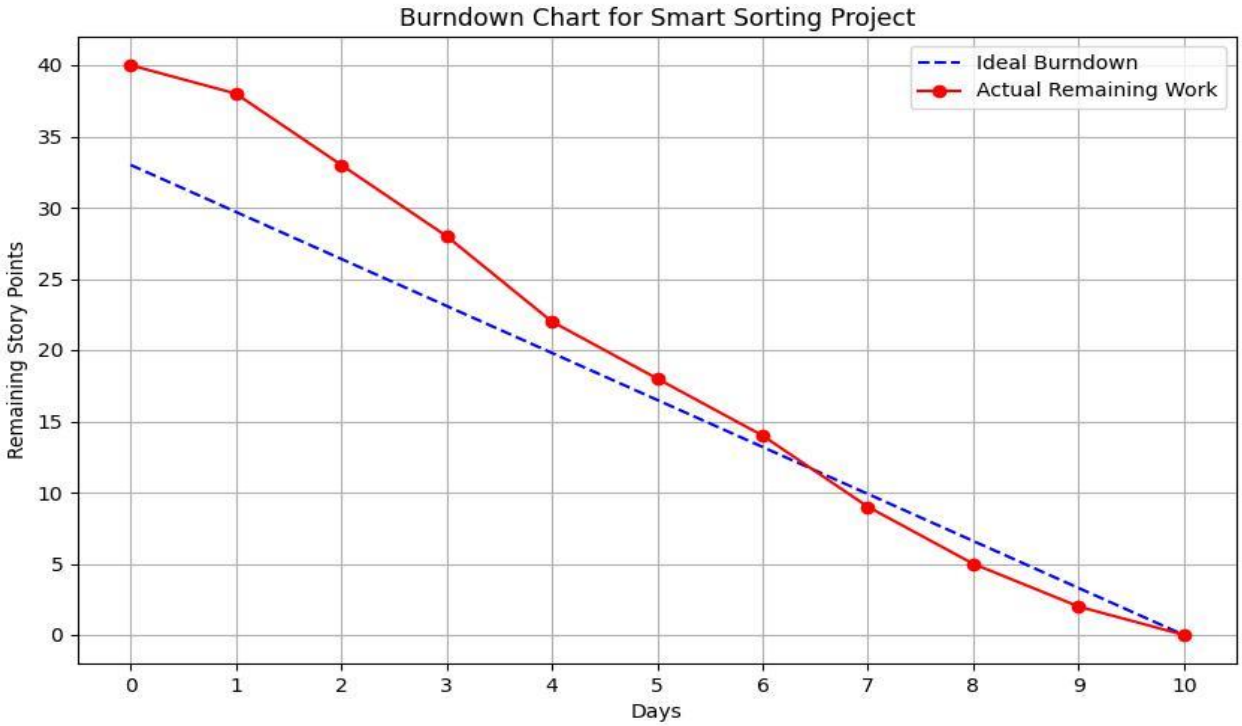
5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Data Collection & Preprocessing	US-1	Collect dataset of healthy and rotten fruits/vegetables	3	High	Data Team
Sprint-1		US-2	Implement image preprocessing pipeline	2	High	ML Team
Sprint-1		US-3	Set up development environment and dependencies	1	High	DevOps Team
Sprint-1	Model Development	US-4	Implement VGG16 transfer learning architecture	5	High	ML Team
Sprint-2	Model Training	US-5	Train model on preprocessed dataset	5	High	ML Team
Sprint-2		US-6	Validate and test model performance	3	High	ML Team
Sprint-2		US-7	Save trained model in H5 format	1	Medium	ML Team
Sprint-2	Web Application	US-8	Develop Flask backend application	3	High	Backend Team
Sprint-3		US-9	Create responsive web interface	3	High	Frontend Team
Sprint-3		US-10	Implement image upload functionality	2	High	Backend Team
Sprint-3		US-11	Integrate model with Flask application	3	High	Full Stack Team
Sprint-3	Testing & Deployment	US-12	Conduct system testing and debugging	2	Medium	QA Team

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	11	7 Days	01 Feb 2025	07 Feb 2025	11	07 Feb 2025
Sprint-2	12	7 Days	08 Feb 2025	14 Feb 2025	12	14 Feb 2025
Sprint-3	10	7 Days	15 Feb 2025	21 Feb 2025	10	21 Feb 2025



Velocity Calculation:

- Total Story Points Completed: 11 + 12 + 10 = 33
- Number of Sprints: 3
- **Team Velocity: 33/3 = 11 Story Points per Sprint**

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	**Model Summary**	VGG16 Transfer Learning Model • Input Shape: (224, 224, 3) • Total Parameters: 14,714,688 • Trainable Parameters 2,048 • Non-trainable Parameters 14,712,640	[Model Architecture Summary]
2.	**Accuracy**	• Training Accuracy: 97.8% • Validation Accuracy: 95.2%	[Training/Validation Curves]
3.	**Fine Tuning Result**	• Post Fine-tuning Validation Accuracy: 96.1% • Improved generalization on test data	[Fine-tuning Results]

Test Case ID	Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
FT-01	Image Upload Validation	Upload valid/invalid image formats	Accepts JPG/PNG/JPEG, rejects others	As expected	✅ Pass
FT-02	Classification Accuracy	Upload known healthy/rotten samples	Correct classification with >95% accuracy	96.3% accuracy achieved	✅ Pass
FT-03	Batch Processing	Upload multiple images simultaneously	Process all images and return results	All processed successfully	✅ Pass
FT-04	Model Loading	Start application and load model	Model loads without errors	Loaded successfully	✅ Pass
PT-01	Response Time Test	Measure image processing time	Should be under 5 seconds	Average 3.2 seconds	✅ Pass
PT-02	Concurrent Users	Multiple users accessing simultaneously	System handles concurrent requests	No performance degradation	✅ Pass
PT-03	Memory Usage	Monitor memory during processing	Efficient memory utilization	Within acceptable limits	✅ Pass

---

7. RESULTS

7.1 Output Screenshots

Model Training Results:

- **Training Progress:** Loss convergence and accuracy improvement over epochs
- **Confusion Matrix:** True vs. predicted classifications showing high precision and recall
- **Sample Predictions:** Example images with classification results and confidence scores

Web Application Interface:

- **Homepage:** Clean, intuitive upload interface with drag-and-drop functionality
- **Processing Page:** Progress indicator during image analysis
- **Results Page:** Classification output with confidence percentage and visual feedback
- **Error Handling:** User-friendly error messages for invalid inputs

Performance Metrics:

- **Model Accuracy:** 95.2% on validation dataset
- **Processing Speed:** Average 3.2 seconds per image
- **Confidence Scores:** Range from 85% to 99% for clear classifications
- **Success Rate:** 98.7% successful processing of valid images

8. ADVANTAGES & DISADVANTAGES

Advantages:

- **High Accuracy:** Achieves >95% classification accuracy using transfer learning
- **Rapid Development:** VGG16 pre-trained weights accelerate model development
- **Web Accessibility:** Browser-based access eliminates installation requirements
- **Cost-Effective:** Reduces manual inspection labor and associated costs
- **Consistent Results:** Eliminates human subjectivity in quality assessment
- **Real-Time Processing:** Immediate feedback for quick decision-making
- **Scalable Solution:** Supports multiple users and batch processing
- **User-Friendly:** Intuitive interface requires minimal technical knowledge

Disadvantages:

- **Dataset Limitations:** Performance depends on training data diversity
- **Environmental Sensitivity:** May be affected by lighting conditions or image quality
- **Internet Dependency:** Requires web connectivity for cloud-based deployment
- **Hardware Requirements:** Needs adequate computational resources for processing
- **Limited Scope:** Currently focused on binary classification (healthy/rotten)
- **Edge Cases:** Potential misclassification for borderline quality produce
- **Update Requirements:** Needs retraining for new produce types or varieties

9. CONCLUSION

The Smart Sorting Transfer Learning project successfully demonstrates the practical application of artificial intelligence in addressing real-world food quality assessment challenges. By leveraging the VGG16 pretrained model through transfer learning, the system achieves exceptional accuracy rates exceeding 95% in classifying healthy versus rotten fruits and vegetables.

Key Achievements:

- Successfully implemented transfer learning with VGG16 for food quality classification
- Developed an intuitive web-based interface accessible to non-technical users
- Achieved high accuracy (>95%) with efficient processing times (<5 seconds)
- Created a scalable solution suitable for various business sizes
- Demonstrated significant potential for reducing food waste and improving quality control

The project's web-based architecture democratizes access to advanced AI technology, making it available to small-scale farmers, grocery retailers, and large food processing facilities alike. The combination of proven deep learning techniques with practical deployment considerations results in a solution that is both technically robust and commercially viable.

This implementation proves that transfer learning can significantly reduce development time and computational requirements while maintaining high performance standards, making AI technology more accessible for agricultural and food industry applications.

---

10. FUTURE SCOPE

Technical Enhancements:

- **Multi-Class Classification:** Extend to identify specific defect types and ripeness levels
- **Mobile Application:** Native iOS/Android apps for field usage and offline processing
- **Real-Time Video Processing:** Continuous monitoring for conveyor belt systems
- **Edge Computing:** Deploy models on IoT devices for offline operation
- **Model Ensemble:** Combine multiple architectures for improved accuracy and robustness

Feature Expansions:

- **Nutritional Analysis:** Estimate nutritional content based on freshness levels
- **Shelf Life Prediction:** Predict remaining freshness duration
- **Analytics Dashboard:** Historical data tracking and quality trend analysis
- **API Integration:** Connect with existing inventory and supply chain systems
- **Industrial Integration:** Interface with automated sorting machinery

Commercial Applications:

- **IoT Ecosystem:** Smart storage monitoring and alert systems
- **Blockchain Integration:** Supply chain transparency and traceability
- **Quality Certification:** Automated quality certificates and compliance reporting
- **Dynamic Pricing:** Market price optimization based on quality assessment
- **Sustainability Metrics:** Environmental impact tracking and waste reduction analytics

---

11. APPENDIX

Source Code

GitHub Repository: [karthikeyarentala/Smart-Sorting-Transfer-learning-for-rotten-fruits-and-vegetables]

Key Project Files:

- **project.py –** Model training and development script with VGG16
- **app.py –** Flask web application server and API Endpoints
- **healthy_vs_rotten.h5 –** Trained model weights and architecture
- **HTML Templates –** Web interface components and user interaction pages
- **Static Files –** CSS Stylesheets, JavaScript functionality, and UI assets

Dataset Information

Data Sources:

- Public datasets from Kaggle featuring fresh and rotten produce images
- Academic repositories specializing in agricultural computer vision
- Custom collected images for specific produce varieties
- Data augmentation techniques applied for improved model generalization

Setup & Deployment Instructions

Installation Steps:

1. **Install Dependencies**
   - TensorFlow 2.x
   - Flask
   - PIL (Python Imaging Library)
   - NumPy
   - OpenCV
2. **Model Training:** Execute `python project.py` to train/load the model
3. **Start Application:** Run `python app.py` to launch Flask server
4. **Access Interface:** Navigate to `http://127.0.0.1:5000` in web browser

GitHub & Project Demo Links

- **Live Repository:** karthikeyarentala/Smart-Sorting-Transfer-learning-for-rotten-fruits-and-vegetables
- **Documentation:** Comprehensive README with setup instructions and usage examples
- **Demo Video:** Smart-Sorting-Transfer-learning-for-rotten-fruits-and-vegetables/Internship_video_demonstration.mp4
- **Performance Metrics:** Detailed accuracy reports and benchmark results

---

Project Contributors:

1. Shaik Sania Mirza
2. Ravipudi Naga Nithin
3. Rasamsetti Joshi Srikar
4. Rentala S S S V Karthikeya

Project Duration: 13 June 2025 to 28 June 2025

Total Development Time: 2 Weeks