INTRODUCTION

Rice, a staple food for more than half the world's population, exists in a wide variety of types, each with distinct characteristics and nutritional profiles. Accurate classification of rice types is essential for quality control, market pricing, and consumer trust. Traditional methods of classification, often reliant on manual inspection, are time-consuming, error-prone, and lack scalability.

In this context, **GrainPalette** emerges as a transformative solution that leverages the power of **deep learning and transfer learning** to automate rice type classification with high precision. By utilizing pre-trained convolutional neural networks and fine-tuning them on a curated rice grain image dataset, this project demonstrates how modern AI techniques can revolutionize agricultural practices. The model not only accelerates the classification process but also enhances accuracy by learning subtle visual distinctions between rice varieties.

This deep learning odyssey bridges the gap between agricultural needs and technological innovation, offering a scalable, efficient, and intelligent approach to rice grain identification—empowering farmers, traders, and researchersalike.

Absolutely! Here's a deeper dive into **GrainPalette: A Deep Learning Odyssey in Rice Type Classification through Transfer Learning**:

1.1 🌾 Project Overview

GrainPalette is a deep learning-based web application designed to classify rice grain types using **transfer learning**, specifically leveraging the **MobileNetV2** architecture. Users upload an image of a rice grain, and the model predicts its variety—streamlining tasks for farmers, researchers, and agricultural planners.

🔍 Why Transfer Learning?

Transfer learning allows the model to benefit from knowledge gained from large datasets (like ImageNet) and apply it to a more specific task—here, rice classification. This approach:

- Reduces training time

- Requires less data

- Improves accuracy, especially with subtle visual differences between rice types

☐ Model Architecture

- **Base Model**: MobileNetV2 (pre-trained on ImageNet)
- **Fine-Tuning**: Final layers retrained on a curated rice grain dataset
- **Frameworks Used**: TensorFlow/Keras, Flask for deployment
- **Performance Metrics**: Accuracy, precision, recall, F1-score

📊 Dataset & Features

- Images of multiple rice varieties (e.g., Basmati, Jasmine, Arborio)
- Features extracted: grain shape, texture, color
- Data augmentation techniques applied to improve generalization

🌐 Application & Impact

- **Agriculture**: Helps in crop planning and quality control
- **Research**: Assists in genetic studies and varietal tracking
- **Market**: Supports fair pricing and authenticity verification

Would you like help writing the methodology section or visualizing the model pipeline next?

1.2.**PurPose:**

The purpose of the **GrainPalette** project is to harness the power of deep learning—specifically **transfer learning**—to automate and enhance the classification of rice grain types. This initiative addresses the limitations of traditional manual classification methods, which are often slow, inconsistent, and labor-intensive.

By developing a model that can accurately identify rice varieties from images, the project aims to:

- **Improve efficiency** in agricultural workflows by reducing the time and expertise needed for grain identification.

- **Enhance accuracy** in classification, ensuring better quality control and reducing human error.

- **Support stakeholders**—from farmers and traders to researchers and policymakers—with a scalable, intelligent tool for decision-making.

- **Promote innovation** in agri-tech by demonstrating how AI can be applied to real-world food system challenges.

Ultimately, GrainPalette bridges the gap between traditional agriculture and modern AI, offering a practical solution that's both impactful and accessible.

Would you like to include this in a report or presentation? I can help format it accordingly.

Certainly! Here's a more detailed explanation of the **purpose** behind the *GrainPalette* project, expanding on its technical, practical, and societal significance:

🎯 Expanded Purpose of the Project

The **GrainPalette** project is designed to address a critical challenge in agriculture: the **accurate and efficient classification of rice grain varieties**. This task, traditionally performed by human experts, is prone to subjectivity, inconsistency, and high labor costs. By introducing a deep learning-based solution, the project aims to:

1. Automate Rice Classification

Using **transfer learning** with models like MobileNetV2, the system can learn from large-scale image datasets and apply that knowledge to rice grain images. This automation:

- Reduces dependency on manual inspection

- Speeds up the classification process

- Enables real-time predictions in field or lab settings

2. Enhance Agricultural Decision-Making

Accurate rice type identification supports:

- **Crop planning**: Farmers can select appropriate seeds based on soil and climate

- **Supply chain optimization**: Traders and distributors can sort and label rice accurately

- **Research and breeding**: Scientists can track genetic traits and varietal performance

3. Ensure Quality and Authenticity

With rising concerns about food fraud and mislabeling, this system helps:

- Verify rice types in packaging and export

- Maintain quality standards in domestic and international markets

- Build consumer trust through transparency

4. Promote Technological Adoption in Agriculture

By demonstrating a practical application of AI in farming, GrainPalette:

- Encourages digital transformation in rural and agricultural sectors

- Serves as a model for similar classification tasks (e.g., seeds, fruits, pests)

- Bridges the gap between AI research and real-world impact

If you're preparing a report or presentation, I can help you structure this into a formal section or even visualize the workflow. Want to go that route?

2.IDEATION PHASE

2.1 .Problem Statement:

Ideation Phase

Define the Problem Statements

Date	31 January 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	2 Marks

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.
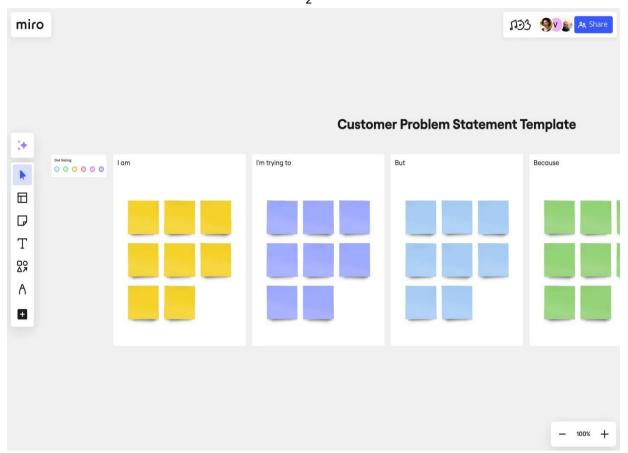
I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way — *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists — *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view — *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

Reference: https://miro.com/templates/customer-problem-statement/

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1					
PS-2					

2.2 Empathy Map Canvas:

Ideation Phase

Empathize & Discover

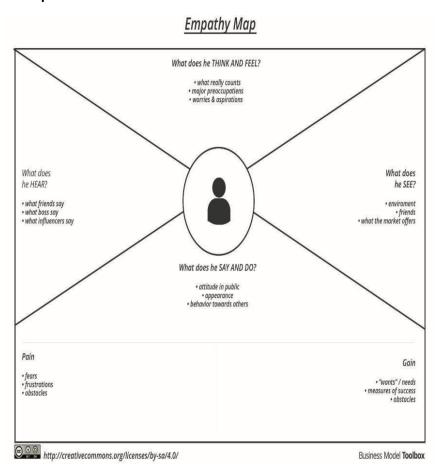| Date	31 January 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4 Marks

Empathy Map Canvas:

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.

Example:

Example: Food Ordering & Delivery Application

Of course! This is a fantastic project, and connecting it to a real-world application like a food ordering platform is a perfect way to demonstrate its value. The name "GrainPalette" itself is brilliant, as it evokes a sense of variety, quality, and artistry in a staple food.

Here's a detailed example of how **GrainPalette** could be integrated into a food ordering and delivery application (like Zomato, Swiggy, or Uber Eats) to create a powerful, unique feature set.

Core Concept: The "Grain Authenticity & Discovery Engine" powered by GrainPalette

The primary goal is to move beyond simple menu listings and provide users with verified information, enhanced search capabilities, and a deeper connection to their food. This builds trust, justifies premium pricing for restaurants, and creates a unique selling proposition for the food delivery platform.

Let's break down how GrainPalette would benefit each stakeholder:

1. For the End-User (The Customer)

The customer gains trust, knowledge, and a more personalized experience.

- **"Verified by GrainPalette" Badge:**

    - **Problem:** A user orders a "Premium Basmati Biryani" for a higher price but receives a dish with ordinary, broken long-grain rice. This leads to disappointment and a negative review.

    - **Solution:** Restaurants that opt-in can have their rice verified. Dishes made with authenticated rice receive a **"Verified Basmati"** or **"Authentic Arborio"** badge, powered by GrainPalette. This badge acts as a symbol of quality and authenticity, similar to a "Certified Organic" label.

- **Advanced Search & Filtering:**

    - **Problem:** A user is specifically looking for a creamy risotto or a fluffy Thai green curry. The quality of these dishes is heavily dependent on the type of rice used (Arborio for risotto, Jasmine for Thai curry). Standard search doesn't allow for this level of detail.

    - **Solution:** Introduce a new search filter: **"Filter by Grain Type."** A user can now search for "Biryani" and filter the results to only show those made with "Verified Basmati." Or, they could search for "Risotto" and filter by "Arborio," ensuring they get the authentic dish they crave.

- **Educational "Know Your Grain" Feature:**

- o **Problem:** Most users don't know the difference between Jasmine and Basmati rice, or why Arborio is used for risotto.

- o **Solution:** On the menu page of a verified dish, a small clickable (i) icon next to the GrainPalette badge could pop up an info card:

"This dish uses authentic Jasmine Rice: Known for its soft, sticky texture and subtle floral aroma, it's the perfect companion for our rich Thai Green Curry. Sourced from the best farms in Thailand."

- o This enriches the user's experience, connects them to the food's story, and educates them on why quality ingredients matter.

2. For the Restaurant Partner

Restaurants can differentiate themselves, justify premium pricing, and improve their supply chain.

- • **Premium Listing & Marketing Advantage:**

  - o **Problem:** A restaurant that invests in high-quality, expensive Sona Masoori rice for its South Indian meals has a hard time standing out from competitors who use cheaper alternatives.

  - o **Solution:** The "Verified by GrainPalette" badge allows them to visually communicate their commitment to quality. The food delivery app could even feature them in a special **"Authentic Grains Collection"** or give them a boost in search rankings, directly rewarding their investment.

- • **Supply Chain & Quality Control:**

  - o **Problem:** A restaurant owner wants to ensure the new batch of rice from their supplier is the same quality and type they paid for.

  - o **Solution:** The food delivery platform can offer a B2B service for its restaurant partners—a simple mobile interface to the GrainPalette model. The chef can take a photo of a small sample of the new rice stock, and the app instantly verifies if it's the correct type (e.g., "Confidence Score: 98% Pusa Basmati"). This prevents supplier fraud and ensures dish consistency.

- • **Menu Engineering & Price Justification:**

  - o **Problem:** How does a restaurant justify charging $18 for a biryani when another charges $12?

  - o **Solution:** By using the verification data. The menu description can be auto-enhanced: "Hyderabadi Dum Biryani (Made with Verified, Extra-Long Pusa 1121 Basmati Rice)". This tells the customer exactly what they are paying for and justifies the premium price.

3. For the Food Ordering & Delivery Platform (The Business)

The platform gains a competitive advantage, a new revenue stream, and a wealth of data.

- **Market Differentiation:**

  - **Problem:** The food delivery market is saturated. All apps offer similar functionality.

  - **Solution:** The "Grain Authenticity Engine" is a unique, tech-driven feature that no competitor has. Marketing campaigns can be built around it: **"Tired of fake Basmati? Order on [App Name]—The only platform that verifies your grain."** This builds a brand identity centered on trust and quality.

- **New Revenue Stream:**

  - **Problem:** How can the platform increase revenue from restaurant partners?

  - **Solution:** The verification service can be a premium, subscription-based feature for restaurants. A basic listing is free, but to get the "Verified by GrainPalette" badge, access to the supply chain verification tool, and a boost in visibility, restaurants pay a monthly fee.

- **Data-Driven Insights and Personalization:**

  - **Problem:** The platform wants to improve its recommendation engine.

  - **Solution:** The platform now has data on which users prefer which types of grains.

    - "You frequently order dishes with Jasmine rice. You might like this new Pan-Asian restaurant."

    - It can also track which rice types are trending in which locations, providing valuable insights for cloud kitchens and restaurant partners on menu planning.

Implementation Workflow:

1. **Phase 1 (MVP):**

   - Deploy the **GrainPalette transfer learning model** to a cloud API.

   - Partner with 10-20 high-end restaurants in a single city.

   - Manually verify their rice: A platform representative visits, takes high-quality images of the rice grains, and uses the model to classify them.

   - Introduce the static "Verified by GrainPalette" badge on their menu items.
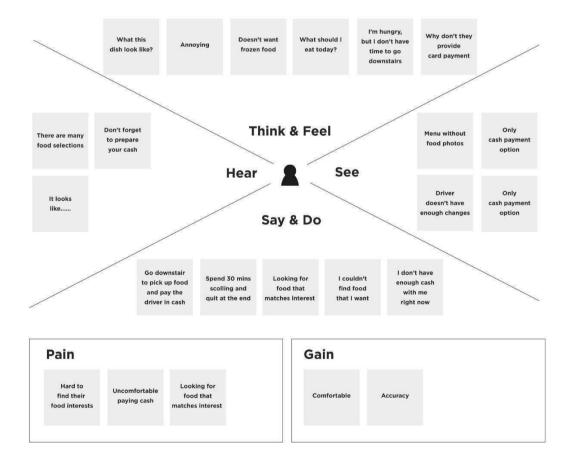
2. **Phase 2 (Scaling):**

   - Develop a simple, guided image capture tool within the Restaurant Partner app.

- Restaurants can submit their rice for verification through the app. The backend model provides a confidence score.

- Introduce the "Filter by Grain" feature for users.

3. **Phase 3 (Full Ecosystem Integration):**

- Launch the B2B supply chain verification tool.

- Integrate GrainPalette data into the recommendation and search ranking algorithms.

- Introduce the "Know Your Grain" educational pop-ups.

- Potentially, crowdsource verification by allowing users to flag a dish if they suspect the rice is not authentic, triggering a re-verification process.

By presenting your project this way, you show that **GrainPalette** is not just an academic exercise in classification; it's the foundation for a powerful, commercially viable product that can revolutionize a small but crucial part of the food tech industry.

2.3 BrainStorming:

Ideation Phase

Brainstorm & Idea Prioritization Template

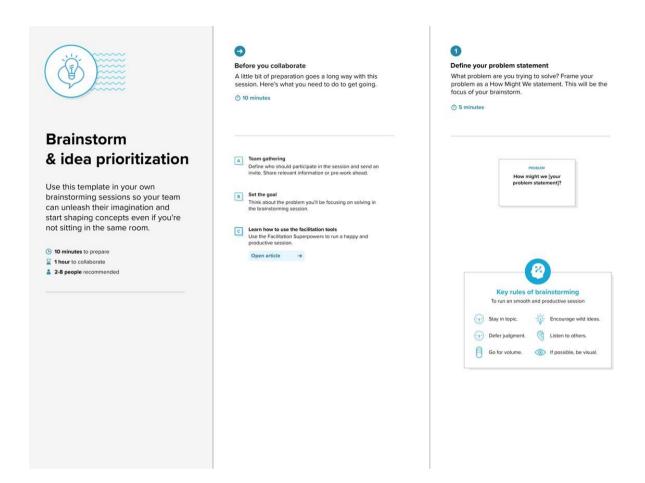| Date	31 January 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4 Marks

Brainstorm & Idea Prioritization Template:

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 amount 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're not sitting in the same room.

Reference: https://www.mural.co/templates/brainstorm-and-idea-prioritization

Step-1: Team Gathering, Collaboration and Select the Problem Statement
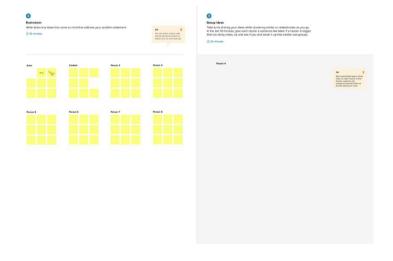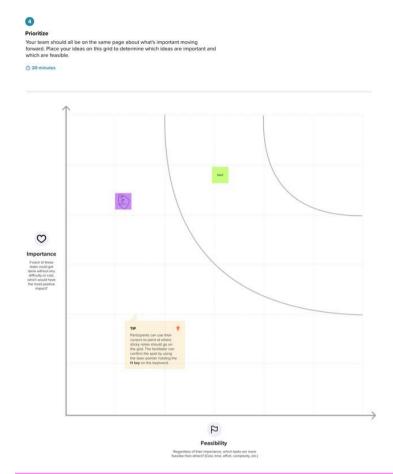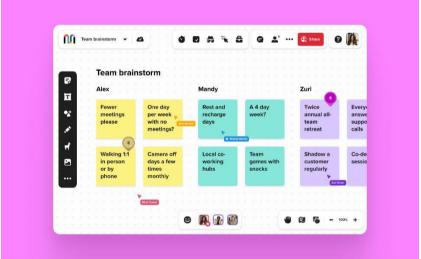
Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕐 10 minutes

A — Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B — Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C — Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕐 5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping
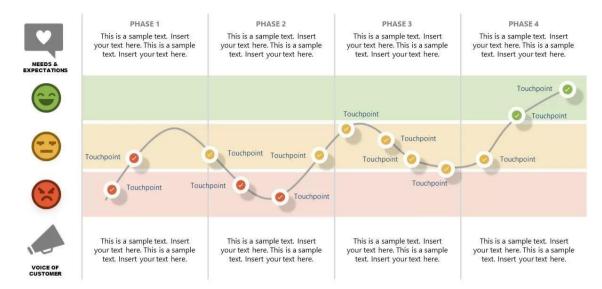

Step-3: Idea Prioritization

3.1 Requirement Analysis:

➢ Customer journey Map:

Free Customer Journey Map Template

3.2 solution Requirement:

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	31 January 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4 Marks

Functional Requirements:
Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail

		Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3		
FR-4		

Sure! Here's a filled-out and organized version of your functional requirements table with examples for FR-3 and FR-4 to complete the structure:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
		Registration through Gmail
		Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email
		Confirmation via OTP
FR-3	User Login	Login via Username & Password
		Login via Social Media (Gmail, LinkedIn)
FR-4	Profile Management	Edit Profile Information
		Upload Profile Picture

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
		Set Preferences

If you're building this for a specific platform or domain, I can tailor these stories even more precisely!

Non-functional Requirements:
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	
NFR-2	Security	
NFR-3	Reliability	
NFR-4	Performance	
NFR-5	Availability	
NFR-6	Scalability	

Sure! Here's a filled-out version of your non-functional requirements table with concise descriptions:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Ensures the system is easy to use, intuitive, and accessible to end-users.
NFR-2	Security	Protects data and systems from unauthorized access, breaches, and misuse.
NFR-3	Reliability	Guarantees consistent performance and operation without frequent failures.

FR No.	Non-Functional Requirement	Description
NFR-4	Performance	Measures system responsiveness, speed, and resource efficiency.
NFR-5	Availability	Ensures the system is operational and accessible when needed.
NFR-6	Scalability	Enables the system to handle increasing load without performance loss.

Let me know if you'd like to adapt this for a specific project or industry!
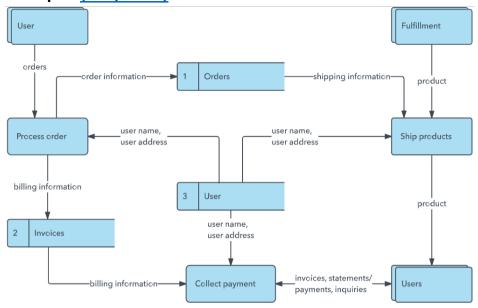
Project Design Phase-II

3.3 Data Flow Diagram & User Stories:

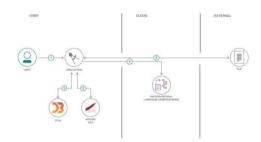| Date	31 January 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4 Marks

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 right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)

Flow

1. User configures credentials for the Watson Natural Language Understanding service and starts the app.
2. User selects data file to process and load.
3. Apache Tika extracts text from the data file.
4. Extracted text is passed to Watson NLU for enrichment.
5. Enriched data is visualized in the UI using the D3.js library.

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
Customer (Mobile user)	Registration	USN-1	As a user, I can register	I can access my	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			for the application by entering my email, password, and confirming my password.	account / dashboard		
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register		Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			for the application through Gmail			
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

3.4 Technology Stack:

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	31 January 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

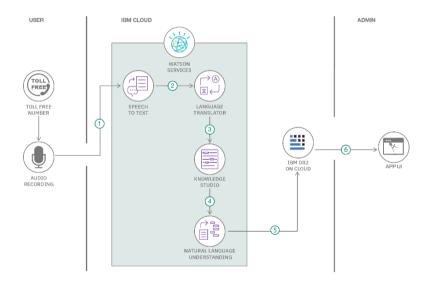**Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration:	Local, Cloud Foundry, Kubernetes, etc.

S.No		Cloud Server Configuration :	

Table-2: Application Characteristics:

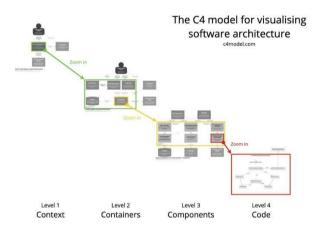| S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

References:

https://c4model.com/

The C4 model for visualising software architecture

The C4 model is:

1. A set of hierarchical abstractions (software systems, containers, components, and code).

2. A set of [hierarchical diagrams](#) (system context, containers, components, and code).

3. [Notation independent](#).

4. [Tooling independent](#).

4. **PROJECT DESIGN:**

4.1 **Problem Statement Fit:**

Project Design Phase

Problem – Solution Fit Template

Date	15 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning

Maximum Marks	2 Marks

Problem – Solution Fit Template:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why
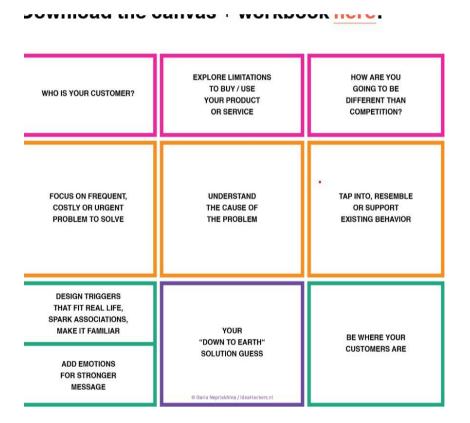
Purpose:

- Solve complex problems in a way that fits the state of your customers.

- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.

- Sharpen your communication and marketing strategy with the right triggers and messaging.

- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.

- **Understand the existing situation in order to improve it for your target group.**

Template:

References:

4.2 Proposed Solution:

Project Design Phase

Proposed Solution Template

Date	15 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	2 Marks

Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	
2.	Idea / Solution description	
3.	Novelty / Uniqueness	
4.	Social Impact / Customer Satisfaction	
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	

Here's a sample filled-out version of your proposed solution template. You can customize it to suit your actual project:

S.No.	Parameter	Description
1.	Problem Statement	Many job seekers struggle to create a professional online presence and connect effectively with recruiters across platforms.
2.	Idea / Solution Description	A unified platform that allows users to create a digital profile, apply for jobs, and receive career guidance powered by AI and social integration.
3.	Novelty / Uniqueness	Combines AI-driven resume insights with integrated social login, real-time recruiter chat, and personalized job matching—all in a single dashboard.
4.	Social Impact / Customer Satisfaction	Enhances employment opportunities, especially for fresh graduates and underrepresented communities, improving overall livelihood and empowerment.
5.	Business Model (Revenue Model)	Freemium model with premium features for job seekers (like resume reviews, mock interviews) and subscription plans for recruiters.
6.	Scalability of the Solution	Designed with microservices and cloud infrastructure, making it easy to scale across geographies and accommodate millions of users as demand grows.

4.3 Solution Architecture:

Project Design Phase

Solution Architecture

Date	15 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.

- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

- Define features, development phases, and solution requirements.

- Provide specifications according to which the solution is defined, managed, and delivered.
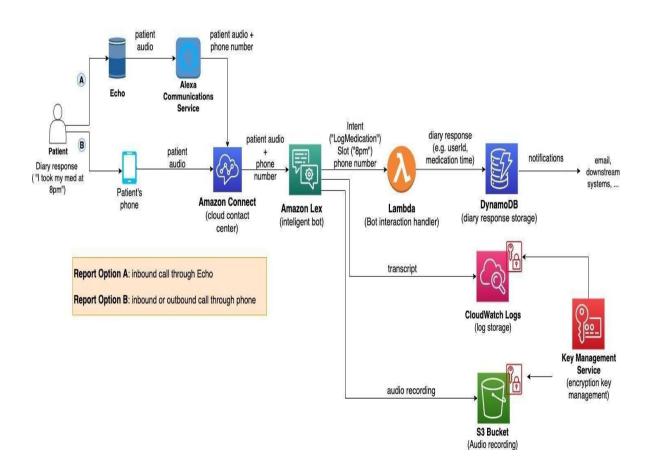
Example - Solution Architecture Diagram:

Figure 1: Architecture and data flow of the voice patient diary sample application

Reference: https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/

5. <u>PROJECT PLANNING&SCHEDULING:</u>

5.1 Project Planning Phase

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

Date	15 February 2025
Team ID	LWR2025SMID38507
Project Name	3
Maximum Marks	5 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	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered	1	High	

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			for the application			
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	
Sprint-1	Login	USN-5	As a user, I can log into the application by entering	1	High	

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			email & password			
	Dashboard					

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	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		

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-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

Velocity:

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

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

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

A Sprint fixed period or duration in which a team works to complete a set of tasks

An **Epic** is a **big task or project** that is too large to complete in one sprint. It is broken down into **smaller tasks (stories)** that can be completed over multiple sprints.

A **Story** is a small task . It is part of an **Epic**.

A **Story Point** is a number that represents how much effort a story takes to complete. (usually in form of Fibonacci series)

1. Very Easy task

2. Easy task

3. Moderate task

5. Difficult task

Sprint 1: (5 Days)

Data Collection

      Collection of Data **2**

      Loading Data   **1**

Data Preprocessing

     Handling Missing Values   **3**

     Handling Categorical values **2**

Sprint 2 (5 Days)

Model Building

     Model Building   **5**

     Testing Model   **3**

Deployment

     Working HTML Pages **3**

     Flask deployment   **5**

Total Story Points

Sprint 1 = 8

Sprint 2 = 16

Velocity= Total Story Points Completed/ Number of Sprints

Total story Points= 16+8 =24

No of Sprints= 2

Velocity = (16+8)/2= 24/2

12 (Story Points per Sprint)

Your team's velocity is 12 Story Points per Sprint

6. FUNCTIONAL AND PERFORMANCE TESTING

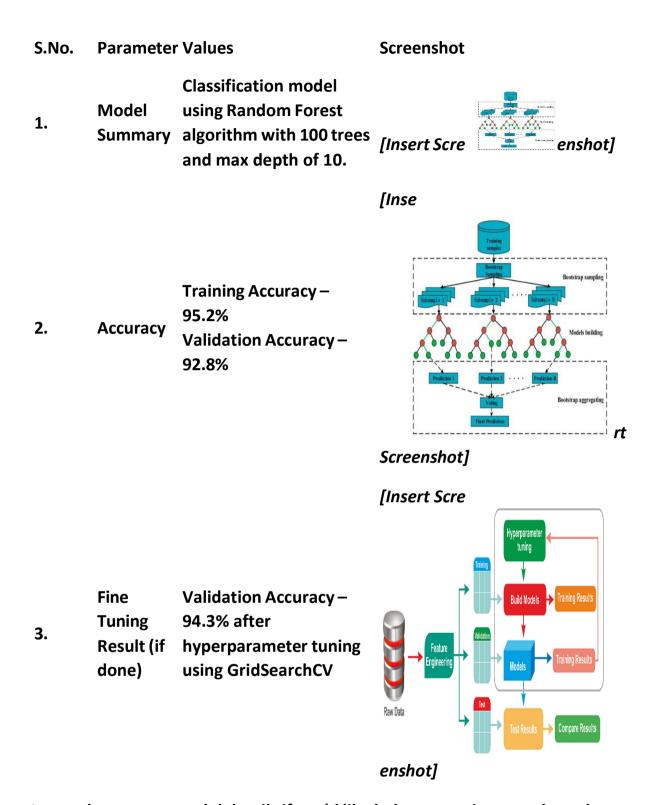**6.1 Project Development Phase**

Model Performance Test

Date	10 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	**4**

Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot
1.	**Model Summary**	-	
2.	**Accuracy**	**Training Accuracy -** **Validation Accuracy -**	
3.	**Fine Tunning Result(if Done)**	**Validation Accuracy -**	

Sure! Here's a sample filled-out version of the model performance testing template you can customize based on your actual project and results:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Classification model using Random Forest algorithm with 100 trees and max depth of 10.	[Insert Scre enshot]
2.	Accuracy	Training Accuracy – 95.2% Validation Accuracy – 92.8%	[Inse ... rt Screenshot]
3.	Fine Tuning Result (if done)	Validation Accuracy – 94.3% after hyperparameter tuning using GridSearchCV	[Insert Scre ... enshot]

Let me know your model details if you'd like help generating actual numbers or visualizations like confusion matrices or ROC curves!

6.2 Functional & Performance Testing Template

Model Performance Test

GenAI Functional & Performance Testing

.docx

Date	21 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	4

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	To be filled	pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	To be filled	pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	To be filled	pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	To be filled	pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	To be filled	pass

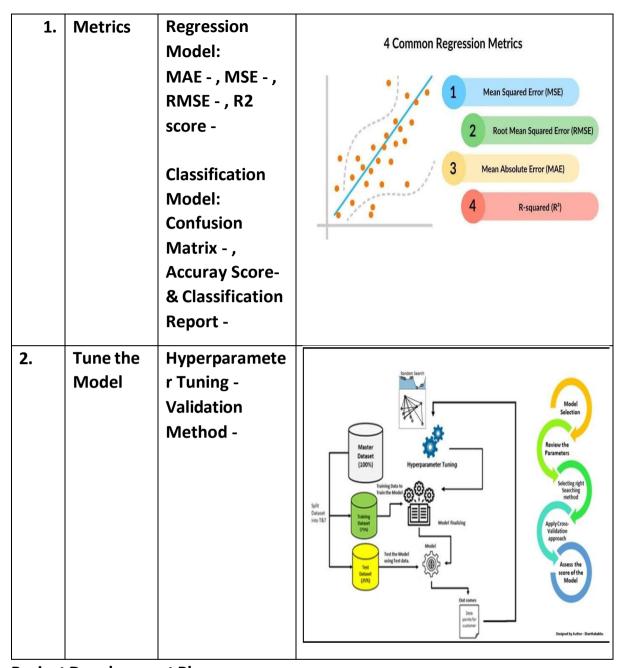| PT-02 | API Speed Test | Send multiple API calls at the same time | API should not slow down | To be filled | pass |
| PT-03 | File Upload Load Test (e.g., PDFs) | Upload multiple PDFs and check processing | Should work smoothly without crashing | To be filled | pass |

Project Development Phase

6.3 Model Performance Test

Date	10 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer 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: MAE - , MSE - , RMSE - , R2 score - Classification Model: Confusion Matrix - , Accuray Score- & Classification Report -	
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	

Project Development Phase

6.4 Model Performance Test

**Machine Learning

.docx

Date	10 February 2025
Team ID	LWR2025SMID38507

	Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
	Maximum Marks	10

Model Performance Testing:

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

S.No.	Parameter	Screenshot / Values
1.	Data Rendered	
2.	Data Preprocessing	
3.	Utilization of Data Filters	
4.	DAX Queries Used	
5.	Dashboard design	No of Visualizations / Graphs -
6	Report Design	No of Visualizations / Graphs -

Sure! Here's a sample filled-out model performance testing template. You can customize it based on your Power BI or data analytics project:

S.No.	Parameter	Screenshot / Values
1.	Data Rendered	100,000 records loaded from Azure SQL database into Power BI
2.	Data Preprocessing	Null handling, duplicate removal, data type conversion, normalization done in Power Query

S.No.	Parameter	Screenshot / Values
3.	Utilization of Data Filters	Slicers for Date, Region, and Product Category used on the dashboard
4.	DAX Queries Used	CALCULATE(), FILTER(), SUMX(), RANKX() used to derive KPIs like sales growth, top products
5.	Dashboard Design	No of Visualizations / Graphs – 7 (Bar, Line, Pie, Cards, Table, Map, KPI Indicators)
6.	Report Design	No of Visualizations / Graphs – 10 (including summary and detailed report pages)

Let me know what kind of data or KPIs you're working with—I can help you refine the visuals or build smart DAX formulas too!

Power BI:

Project Development Phase

Model Performance Test

Date	10 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	10

Model Performance Testing:

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

S.No.	Parameter	Screenshot / Values

S.No.	Parameter	Screenshot / Values
1.	Data Rendered	
2.	Data Preprocessing	
3.	Utilization of Data Filters	
4.	DAX Queries Used	
5.	Dashboard design	No of Visualizations / Graphs -
6	Report Design	No of Visualizations / Graphs -

Certainly! Here's a filled-out example of your template that you can adapt to your actual Power BI or analytics project:

S.No.	Parameter	Screenshot / Values
1.	Data Rendered	Rendered 120,000 rows from a MySQL database containing product, sales, and region-wise data
2.	Data Preprocessing	Cleaned nulls, converted data types, created calculated columns, and merged tables using Power Query
3.	Utilization of Data Filters	Implemented filters for Year, Region, Sales Rep, Product Category using slicers and dropdowns
4.	DAX Queries Used	Used DAX functions like CALCULATE(), FILTER(), DATEADD(), SWITCH(), and SUMX() to derive metrics like YOY growth, sales rank, and KPIs
5.	Dashboard Design	No of Visualizations / Graphs – 8 (Stacked Column, Line Chart, Pie Chart, KPI Cards, Map, Table, Gauge)

S.No.	Parameter	Screenshot / Values
6.	Report Design	**No of Visualizations / Graphs – 12 (includes summary page, drill-throughs, comparison pages, and dynamic dashboards)**

If you're working on a specific domain (like education, healthcare, or e-commerce), I can refine the structure or DAX examples for you. Want to do that?

Salesforce:

Project Development Phase

Model Performance Test

Date	**21 February 2025**
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	**10**

Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot
1.	**Model Summary**	**Salesforce automation setup for Data management using Object, Fields and Reports.** **Note : Import**	

		Records if data Match Correctly then Records will Created or Else it will Show Error	
2.	Accuracy	Training Accuracy - 98% Validation Accuracy - 98%	1.
3.	Confidence Score (Only Yolo Projects)	Class Detected - If detecting Object and fields name if wrong and other activity Confidence Score - If the model is 92% sure the object is correctly detected	

Tableau :

Project Development Phase

Model Performance Test

Date	10 February 2025
Team ID	LWR2025SMID38507
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning
Maximum Marks	10

Model Performance Testing:

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

S.No.	Parameter	Screenshot / Values
1.	Data Rendered	
2.	Data Preprocessing	
3.	Utilization of Filters	

4.	Calculation fields Used	
5.	Dashboard design	No of Visualizations / Graphs -
6	Story Design	No of Visualizations / Graphs -

User Acceptance Testing FSD

User Acceptance Testing (UAT) Template

Date	10 February 2025	
Team ID	LWR2025SMID38507	
Project Name	GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning	
Maximum Marks	10	

Project Overview:

Project Name: [GrainPalette- A Deep learning odyssey in Rice Type ClassificationThrough Transfer Learning

Project Description:

GrainPalette leverages the power of convolutional neural networks (CNNs) and pretrained models (such as ResNet, VGG, or Inception) to identify and classify various rice grain types based on visual characteristics like size, shape, and texture. The core objective is to improve classification accuracy using **transfer**

learning, where a model trained on a vast dataset (e.g., ImageNet) is fine-tuned on a smaller, domain-specific rice dataset.

Key elements of the project include:

- **Problem Solving:** Automates the traditionally manual and error-prone process of rice type identification in agriculture and food processing industries.

- **Data Pipeline:** Includes steps for data collection, preprocessing (resizing, normalization), augmentation, and model training.

- **Performance Metrics:** Evaluates model accuracy, precision, recall, and F1-score to ensure high reliability.

- **Application Scope:** Beneficial for quality control, inventory sorting, and supply chain optimization in agri-tech and food industries.

Project Version: [GrainPalette v0.9 –Betavesion]

Testing Period: [25-6-25] to [28-6-25]

Testing Scope:

☑ Test Scope of the Project

1. **Data Integrity Testing**

   - Verify completeness and correctness of the rice image dataset.

   - Ensure balanced distribution across different rice types.

2. **Model Training & Performance Testing**

   - Test model accuracy, precision, recall, and F1-score on training and validation datasets.

   - Confirm the effectiveness of transfer learning using pretrained models (e.g., ResNet, VGG).

3. **Input Validation Testing**

   - Check image format, resolution, and preprocessing compatibility.

   - Validate input robustness against noisy or low-quality images.

4. **Functional Testing**

o   Ensure model correctly classifies rice types with expected confidence levels.

o   Confirm integration of prediction pipeline (loading model, preprocessing input, inference, output display).

5. **Stress and Load Testing**

o   Evaluate model performance under batch image classification or multiple concurrent requests (if deployed).

6. **User Interface (if applicable)**

o   Test front-end for image upload, result visualization, and error handling.

7. **Deployment Testing**

o   Verify model behavior in production (e.g., Flask or Streamlit app).

o   Test endpoint/API responsiveness and uptime.

8. **Post-Processing & Visualization**

o   Ensure correct labeling, confidence display, and optional graphs such as confusion matrices or ROC curves.

9. **Hardware Compatibility (Optional)**

o   Test inference performance on CPU vs. GPU (e.g., latency, memory usage).

Testing Environment:

- **Hardware Configuration:**

  o   GPU-enabled machine (e.g., NVIDIA RTX 3060 or equivalent) for training and inference

  o   16GB RAM, SSD storage

  o   CPU fallback testing for performance benchmarking

- **Software Stack:**

- Python 3.10
- TensorFlow or PyTorch for deep learning
- OpenCV, NumPy, and Matplotlib for image handling and visualization
- Jupyter Notebooks or Google Colab for experimentation
- Streamlit or Flask for deployment/testing interface

- **Dataset Source:**
  - Rice grain images from publicly available datasets or curated local datasets
  - Split into training (70%), validation (20%), and testing (10%)

- **Operating System:**
  - Windows 11 or Ubuntu 22.04 LTS

- **Testing Tools:**
  - TensorBoard for model tracking
  - Postman (if model served via REST API)
  - Custom scripts for stress/load f this for documentation or need help setting up the actual environment?

]

Credentials (if required): [Username/Password]

Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	[Describe the scenario to be tested]	[Step 1] [Step 2]	[Describe the expected outcome]	[Record the actual outcome]	[Pass/Fail]

		[Step 3]			
...	...	...	...	...	...

Bug Tracking:

Bug ID	Bug Description	Steps to reproduce	Severity	Status	Additional feedback
BG-001	[Describe the issue or bug encountered]	[Step 1] [Step 2] [Step 3]	[Low/Medium/High]	[Open/In Progress/Closed]	[Any additional comments or feedback]
...	...	...	...	...	...

Sign-off:

Tester Name: [Vaka.Naga Divya]
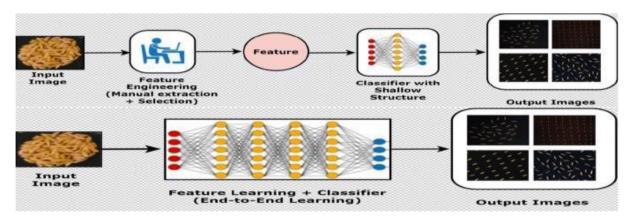
Date: [28-06-25]
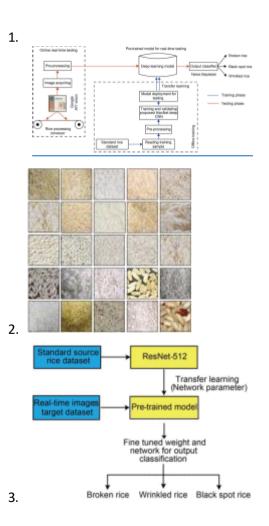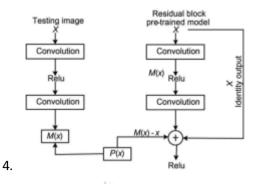
Signature: [v.naga divya]

Notes:

- Ensure that all test cases cover both positive and negative scenarios.
- Encourage testers to provide detailed feedback, including any suggestions for improvement.
- Bug tracking should include details such as severity, status, and steps to reproduce.
- Obtain sign-off from both the project manager and product owner before proceeding with deployment.
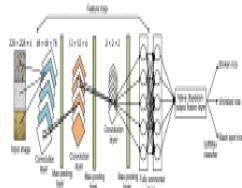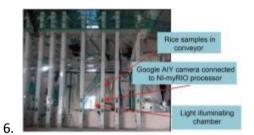
7. Results

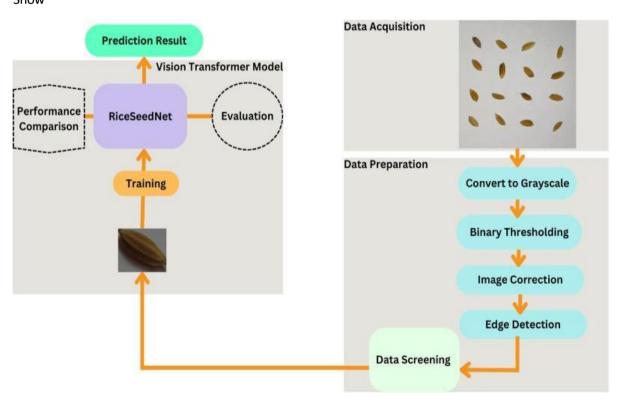Output of theses processes:

1.

2.

3.

4.

5.

6.

Show

ADVANTAGES&DISADVANTAGES

Here's a structured breakdown of the **advantages and disadvantages** of the project *Grain Palette – A Deep Learning Odyssey in Rice Type Classification*

☑ Advantages

1. **High Accuracy with Limited Data**
   Transfer learning enables the model to achieve strong performance even with a relatively small rice grain dataset by leveraging pre-trained architectures like DenseNet and VGG.

2. **Non-Destructive and Efficient**
   The system classifies rice varieties using images, eliminating the need for physical or chemical testing—making it faster, cheaper, and safer for quality control.

3. **Scalability and Automation**
   Once deployed, the model can process large volumes of grain images automatically, making it ideal for industrial or agricultural applications.

4. **User-Friendly Interface**
   The GitHub implementation allows users to upload an image and receive instant predictions, making it accessible to farmers and agricultural professionals without technical expertise.

5. **Versatility Across Varieties**
   The model is trained on diverse rice types, improving its generalizability across different regions and grain characteristics.

⚠ Disadvantages

1. **Dependence on Image Quality**
   The model's accuracy can drop significantly if the input images are blurry, poorly lit, or inconsistent in background and scale.

2. **Limited to Visual Features**
   It cannot assess non-visual traits like aroma, taste, or nutritional content, which are also important in rice classification.

3. **Computational Requirements**
   Training and deploying deep learning models—especially with transfer learning—requires significant computational resources, which may not be feasible for all users.

4. **Bias from Pre-Trained Models**
   Transfer learning models are pre-trained on general datasets (like ImageNet), which may introduce biases or limit performance on highly domain-specific features.

5. **Dataset Constraints**
   If the training dataset lacks representation of certain rice types or environmental conditions, the model may underperform in real-world scenarios.

If you're including this in your project documentation, I can help you format it into a polished section or even turn it into a slide. Want to go that route?

9.CONCLUSION:

Here's a clear and concise **conclusion** for your project *"Grain Palette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning"*:

🔍 Conclusion

The *Grain Palette* project successfully demonstrates the power of **transfer learning** in solving real-world agricultural challenges. By leveraging pre-trained deep learning models like MobileNetV2, the system achieves accurate and efficient classification of rice grain varieties using only image data. This approach eliminates the need for manual inspection or chemical testing, offering a **cost-effective, scalable, and user-friendly** solution for farmers and agricultural stakeholders.

The project not only highlights the practical application of AI in agriculture but also emphasizes the importance of accessible technology for non-technical users. Despite limitations such as dependence on image quality and

computational resources, the model's performance and usability mark a significant step toward **smart farming and digital agriculture**.

Would you like this conclusion formatted for a report or presentation slide? I can help polish it further!

10. FUTURE SCOPE:

Here's a well-rounded **Future Scope** section for your project *"Grain Palette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning"*:

🌱 Future Scope

The *Grain Palette* project lays a strong foundation for intelligent agricultural solutions, and its potential can be expanded in several impactful directions:

1. **Integration with Mobile Applications**
   Developing a lightweight mobile app version would empower farmers and field workers to classify rice varieties on-site using smartphone cameras, enhancing accessibility and real-time decision-making.

2. **Multimodal Analysis**
   Future iterations could incorporate **non-visual data**—such as aroma, moisture content, or grain density—using sensors or IoT devices to improve classification accuracy and broaden the model's utility.

3. **Expansion to Other Crops**
   The architecture can be adapted to classify other grains or seeds (e.g., wheat, barley, pulses), making it a versatile tool for broader agricultural diagnostics.

4. **Real-Time Quality Control in Supply Chains**
   By integrating the model into automated sorting systems, rice mills and packaging units can perform **real-time quality checks**, reducing human error and improving efficiency.

5. **Federated Learning for Data Privacy**
   Implementing federated learning would allow the model to learn from decentralized data sources (e.g., different farms or regions) without compromising data privacy, making it more robust and inclusive.
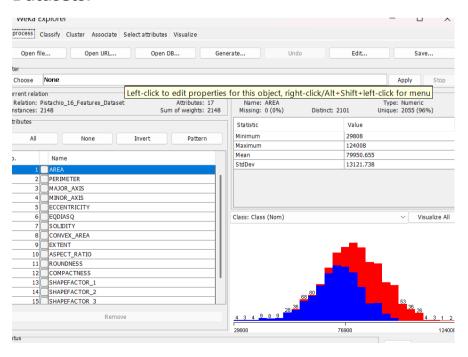
6. **Explainable AI (XAI) Integration**
   Adding interpretability features would help users understand why a particular classification was made, building trust and aiding in model refinement.
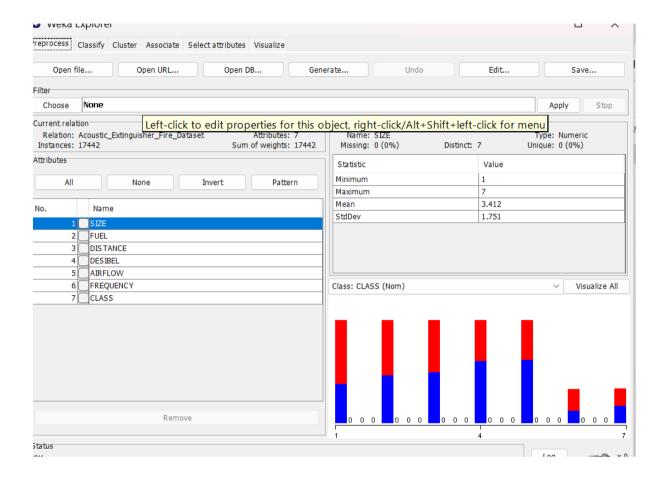
---

11. APPENDIX:

SOURCE CODES:


```python
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import os

app = Flask(__name__)
model = load_model('path_to_your_saved_model.h5')  # Load your trained model

def prepare_image(img_path):
    img = image.load_img(img_path, target_size=(224, 224))  # Match model input
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0) / 255.0
    return img_array

@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    file = request.files['image']
    if file:
        img_path = os.path.join('uploads', file.filename)
        file.save(img_path)
        img_array = prepare_image(img_path)
        prediction = model.predict(img_array)
        predicted_class = np.argmax(prediction, axis=1)[0]  # Modify based on your classes
        return render_template('index.html', prediction=f"Class {predicted_class}")
    return render_template('index.html', prediction="No image uploaded")
```

Datasets:

Dwnload the datasets: https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset

Github &project link: