GrainPalette - AIML Project Documentation

1. Introduction
Project Title: GrainPalette - A Deep Learning Odyssey In Rice Type Classification

Through Transfer Learning Team Members

1.Sarvan Singh

2.S. Nithin Kumar

3. Shaik Asif

4.Shaik Rafi

GrainPalette is an advanced AI-powered image classification system focused on

identifying different types of rice grains.

Leveraging the power of Transfer Learning, this project aims to revolutionize the agricultural

sector by enhancing the speed, accuracy, and scalability of rice grain type identification.

2. Architecture

- Model: Transfer Learning with ResNet50 fine-tuned on a labeled dataset of rice types.
- Pipeline:
1. Image preprocessing (resizing, normalization)
2. Feature extraction
3. Classification via dense layers with softmax
3. Training and Inference

Training:

- Dataset split: training, validation, test
- Augmentation: rotation, zoom, flip
- Early stopping and model checkpointing used Inference:
- Accepts an image, processes it, and outputs rice type with confidence score
4. Setup Instructions Setup

Instructions:

GrainPalette - AIML Project Documentation

- Prerequisites: Python 3.7+, TensorFlow/Keras, OpenCV, NumPy, Flask or CSS-

based UI - Steps:

1. Clone repo
2. Create virtual env
3. pip install -r requirements.txt
4. Prepare dataset
5. Run training script
6. Launch app using: open index.html (CSS UI) or python app.py
7. Github Link: https://github.com/Sarvansingh/grainpalette---a-deep-learning-odyssey-in-rice-type-classification.git

5. Folder Structure Folder

Structure:

- /dataset: Labeled rice images
- /model: Training scripts and models
- /app: UI and API logic (CSS or Flask)
- /utils: Helper functions - requirements.txt, README.md

6. Running the Application

Running the Application:

1.Install dependencies

2.Activate environment

3. Launch:

- CSS UI: open index.html
- Flask: python app.py

7. API

Documentation API Documentation:

- POST /predict: Accepts image, returns rice type
- GET /model-info: Returns model details
- POST /retrain: (Future scope)

8. Authentication and UI

Authentication:

- None in current version- Future: JWT login, admin data upload

User Interface:
- Built using CSS for responsive layout
- Upload image component
- Displays predictions

9. Testing Testing:

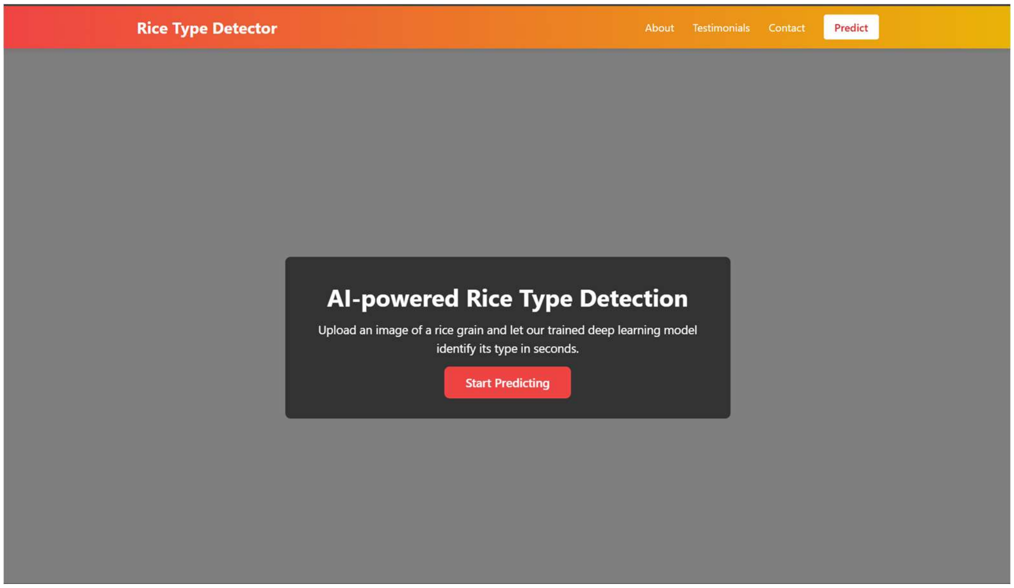
- Manual testing on diverse images
- Metrics: Accuracy, precision, recall, F1
- Confusion Matrix used - Future: CI/CD with pytest

10.Screenshot or Demo

Screenshots and Demo:

Below are UI screenshots of the GrainPalette rice classification system built using HTML + CSS.
Demo Video: https://drive.google.com/file/d/1Txt3TpLmYikynRLa2zR35P9j3jI-EADb/view?usp=sharing

Screenshot 1: Home and Model Info

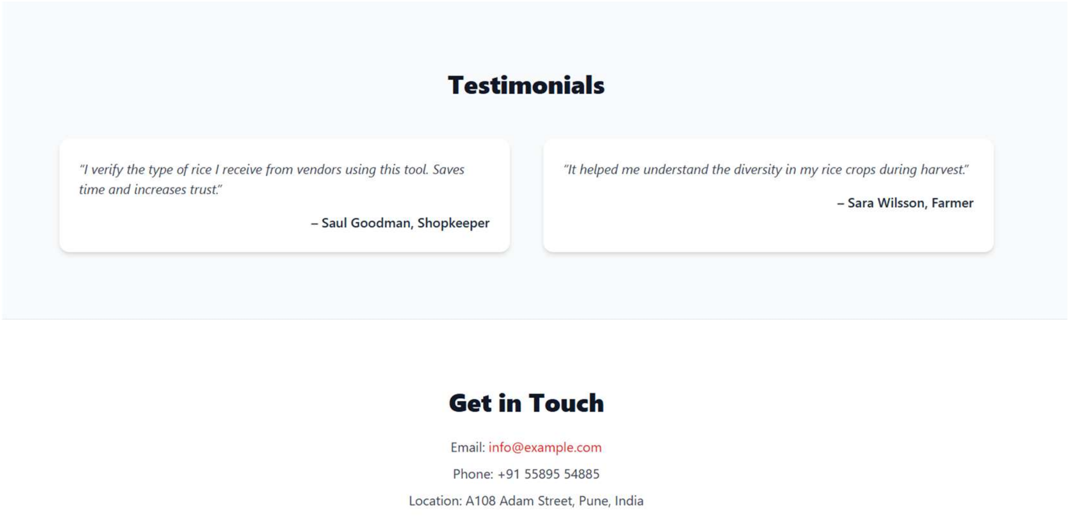


AI-powered Rice Type Detection

Upload an image of a rice grain and let our trained deep learning model identify its type in seconds.

Screenshot 2: Contact

Page



Testimonials

"I verify the type of rice I receive from vendors using this tool. Saves time and increases trust."

– Saul Goodman, Shopkeeper

"It helped me understand the diversity in my rice crops during harvest."

– Sara Wilsson, Farmer

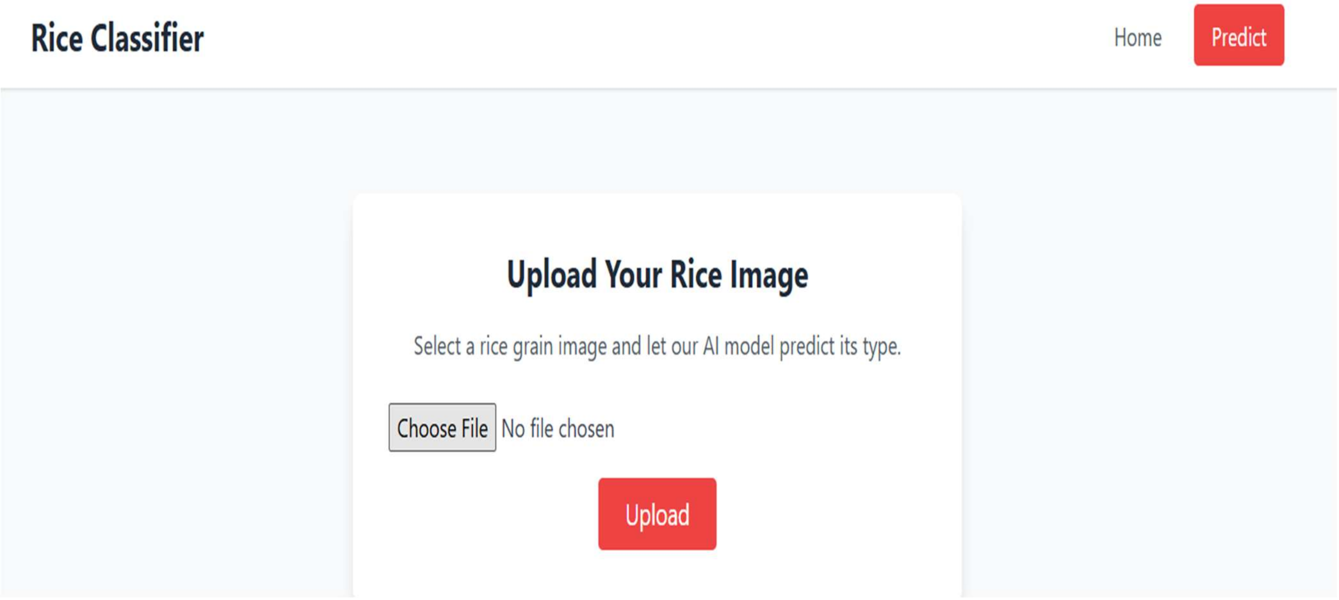
Get in Touch

Email: info@example.com

Phone: +91 55895 54885

Location: A108 Adam Street, Pune, India

Screenshot 3: Upload Interface

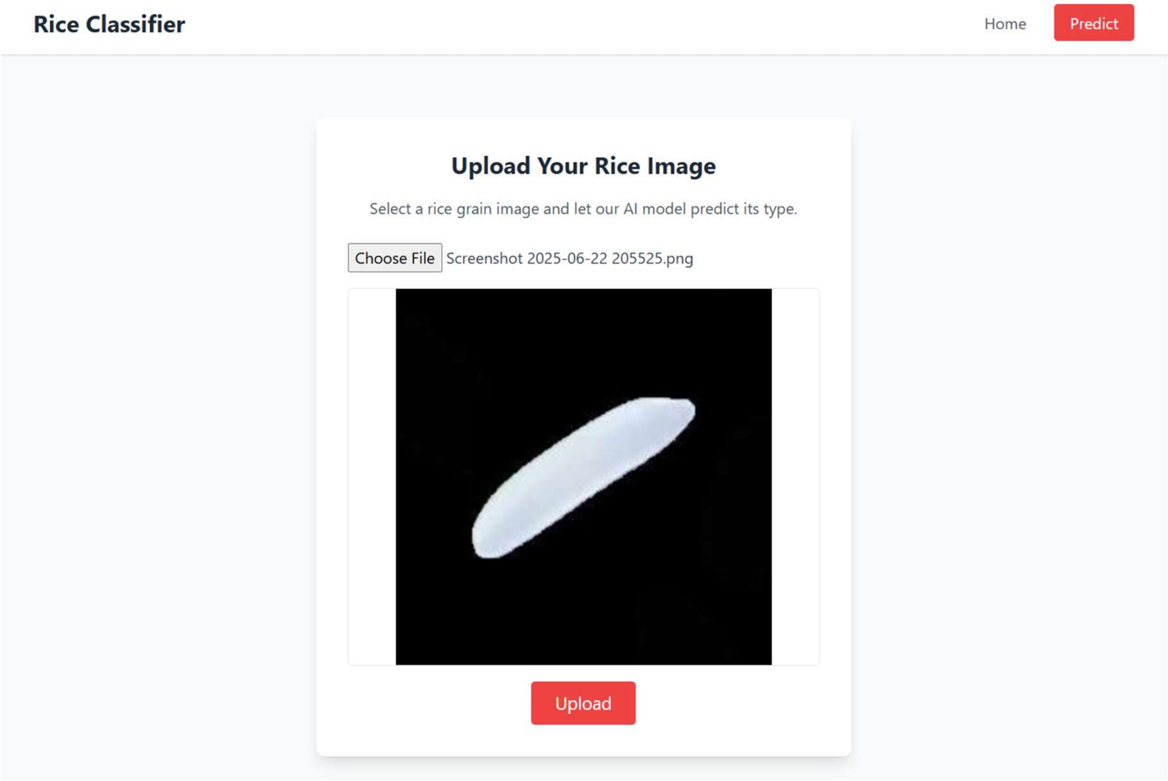


Upload Your Rice Image

Select a rice grain image and let our AI model predict its type.

No file chosen

Screenshot 4: Image Preview

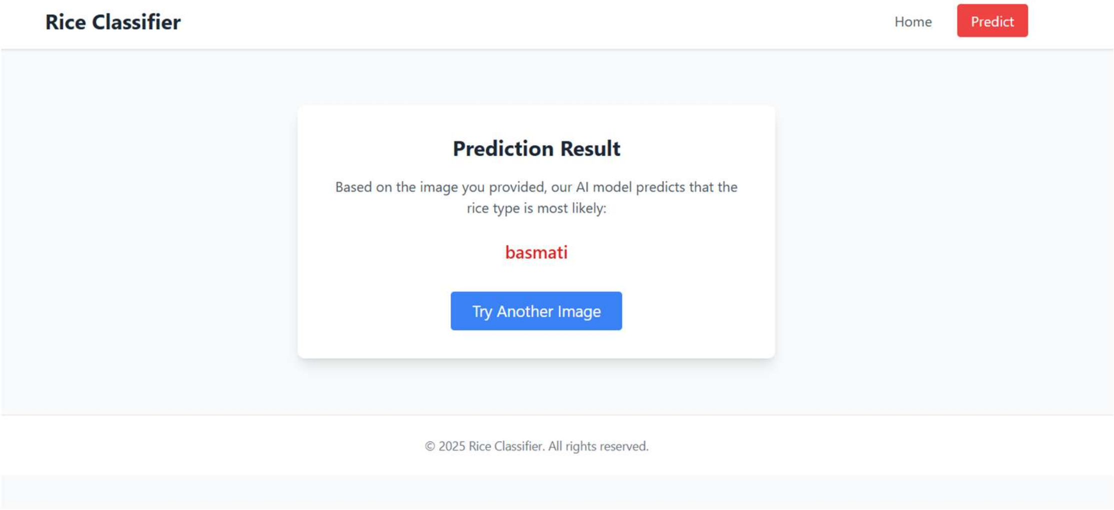


Upload Your Rice Image

Select a rice grain image and let our AI model predict its type.

Choose File Screenshot 2025-06-22 205525.png

Screenshot 5: Prediction Result



Prediction Result

Based on the image you provided, our AI model predicts that the rice type is most likely:

basmati

© 2025 Rice Classifier. All rights reserved.

10.KnownIssues and Future Enhancements

GrainPalette - AIML Project Documentation

Known Issues:

- Sensitive to lighting, blur
- Accuracy drops with mixed grains- Limited by pre-trained model rice types

Future Enhancements:

- Larger dataset
- Multi-type detection
- Mobile app with TensorFlow Lite
- Real-time webcam integration
- Multi-language UI