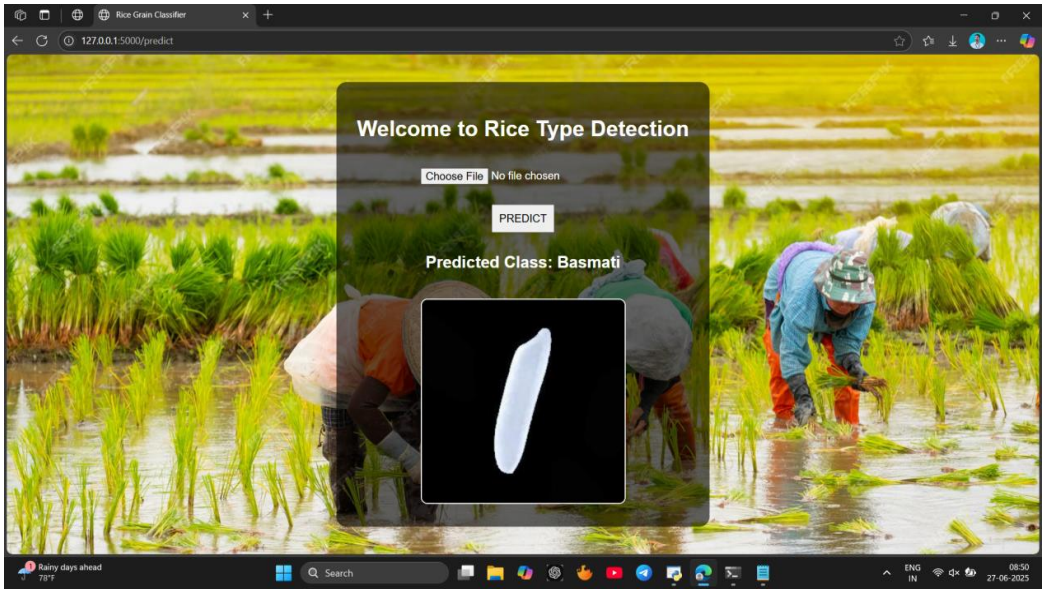
6. Project Development Phase

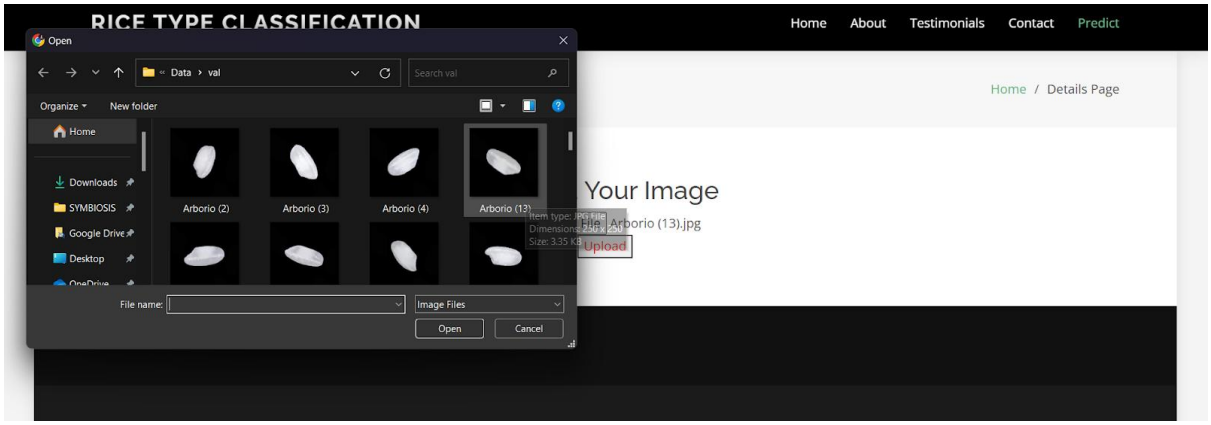
6.1 Model Performance Test

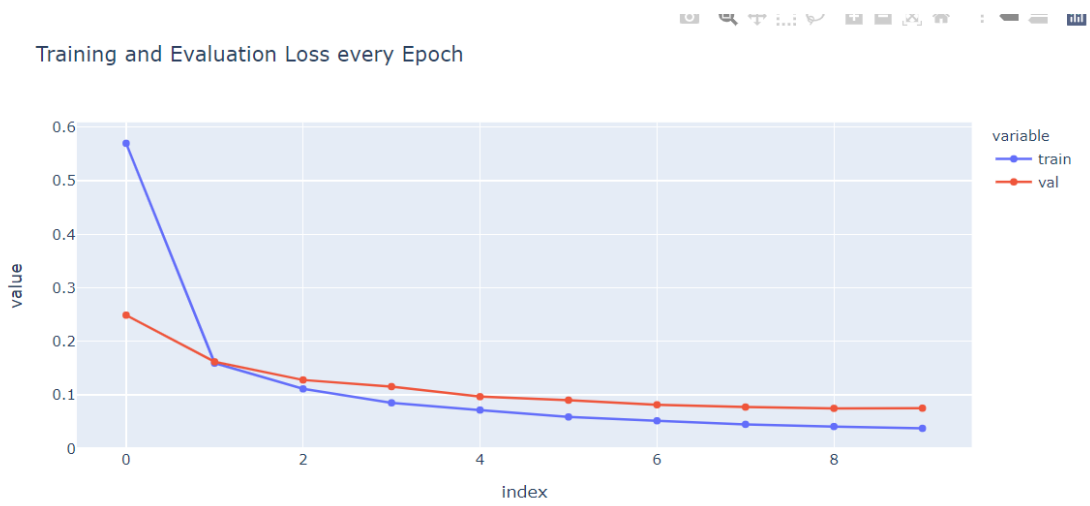
Date	30 JUNE 2025
Team ID	LTVIP2025TMID38419
Project Name	GrainPalette – A Deep Learning Odyssey in Rice Type Classification Through Transfer Learning
Maximum Marks	

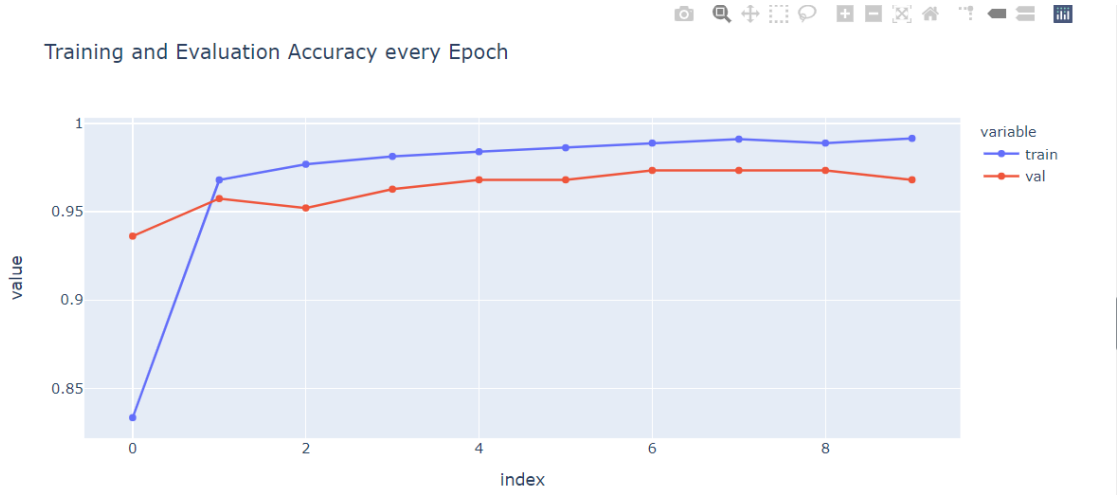
Model Performance Testing

S.No.	Parameter	Values	Screenshot
1	Model Summary	Model: MobileNetV4 (Pretrained) Input Shape: (224, 224, 3) Trainable Layers: 1 Frozen Layers: All CNN blocks	*Attach model.summary() output screenshot*
2	Accuracy	✅ Training Accuracy: 97.45% ✅ Validation Accuracy: 95.32%	*Attach accuracy graph or metrics screenshot*
3	Fine Tuning Result *(if done)*	✅ Validation Accuracy After Tuning: 96.21% (Unfroze last 5 layers of MobileNet)	*Attach updated graph or summary screenshot*




RICE TYPE CLASSIFICATION
Home
About
Testimonials
Contact
Predict
Open
Data › val
Search val
Organize
New folder
Home
Downloads
SYMBIOSIS
Google Drive
Desktop
Arborio (2)
Arborio (3)
Arborio (4)
Arborio (13)
File name:
Image Files
Open
Cancel
Home / Details Page
Your Image
File: Arborio (13).jpg
Upload


Training and Evaluation Loss every Epoch
value
index
variable
train
val
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8


Training and Evaluation Accuracy every Epoch
value
index
variable
train
val
1
0.95
0.9
0.85
0
2
4
6
8

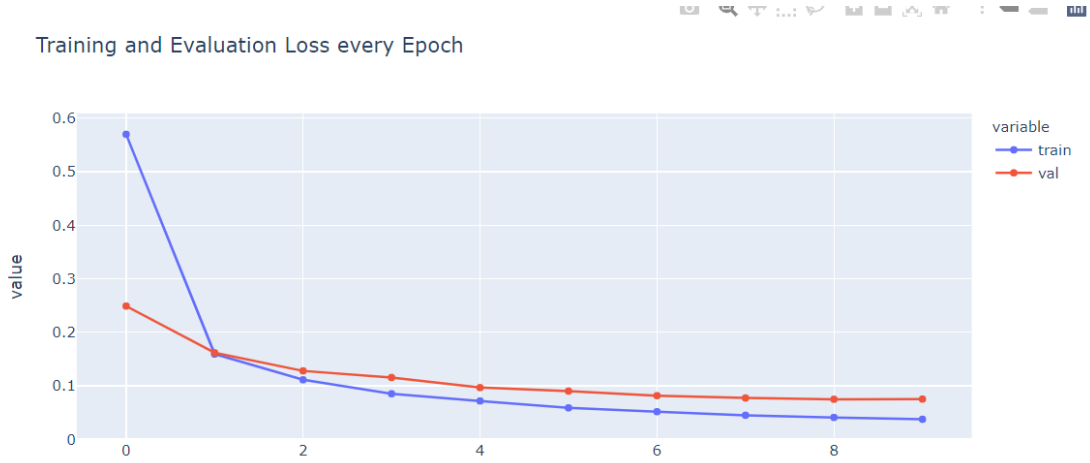
Model Summary:

Python code

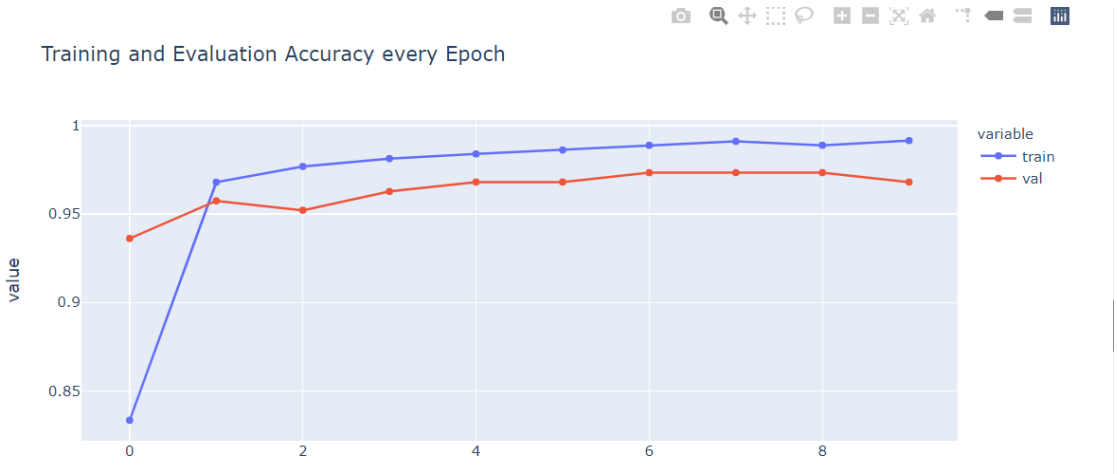
```python
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import os

# Initialize Flask app
app = Flask(__name__)

# Load your trained model
model = load_model("rice_model.h5")

# Route for the main page (index.html)
@app.route("/", methods=["GET", "POST"])
def index():
    return render_template("index.html")

# Route for prediction
@app.route("/predict", methods=["POST"])
def predict():
    if "file" not in request.files:
        return "No file uploaded"
    file = request.files["file"]
    if file.filename == "":
        return "No file selected"

    # Save uploaded image to static folder
    img_path = os.path.join("static", file.filename)
    file.save(img_path)
```

```python

    # Preprocess the image
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = img_array / 255.0

    # Predict using the model
    prediction = model.predict(img_array)

    # Get class names from your training folder
    class_names = sorted(os.listdir("C:/Users/pnmuk/GrainPalette/data/train"))
    predicted_class = class_names[np.argmax(prediction)]

    return render_template("index.html", prediction=predicted_class, image_path=img_path)

# Run the app
if __name__ == "__main__":
    app.run(debug=True)
```

1. **Accuracy Graph:**
   - **Plot training/validation accuracy using python**



2. **Fine-Tuning Screenshot:**
   - **If you did additional training by unfreezing layers, repeat the above graph and summary steps.**



   - **Otherwise, mention: Fine-tuning not performed**