**Performance Testing**

**1. Objective**

Performance testing was conducted to assess the responsiveness, reliability, and efficiency of the Smart Sorting system. The primary goal was to ensure that the model can handle image classification tasks smoothly and quickly, even under repeated or simultaneous usage.

**2. Testing Setup**

| Parameter | Description |
| --- | --- |
| Platform | Localhost (Flask server on Windows 10) |
| Hardware | Intel Core i5, 8 GB RAM, SSD |
| Model | MobileNetV2 (Transfer Learning – Keras/TensorFlow) |
| Input Image Size | Average 100–200 KB (JPG/PNG formats) |
| Number of Test Samples | 30 fruit images (mixed fresh and rotten, 3 categories) |
| Browser Used | Google Chrome |

**3. Key Performance Metrics**

| Scenario | Measured Result | Remarks |
| --- | --- | --- |
| Image Upload to Result Time | ~1.8 seconds | Fast and consistent across test cases |
| Model Prediction Time (Single Image) | ~0.65 seconds | Lightweight model ensures quick inference |
| Peak Memory Usage | ~450 MB | No memory overflow observed |
| Concurrent Requests (2–3 Users) | ~2.3 seconds per image | Slight increase but stable performance |
| UI Responsiveness | <200 ms for progress updates | Immediate user feedback |

**4. Stress & Load Testing**

* The system was tested with continuous uploads (up to 10 images per minute).
* It maintained smooth and uninterrupted performance.
* The server handled light concurrency without errors or delays.
* Memory consumption stayed consistent during prolonged testing.

**5. Observations**

* The application is well-suited for real-time usage at small scale (e.g., classroom, retail demo).
* No crashes or slowdowns were encountered during repeated usage.
* The current setup (CPU-based) performs well for its intended use.
* Optimization through GPU deployment or background task queuing could improve performance further.

**6. Summary**

The Smart Sorting application delivers efficient performance with fast response times and stable behavior. Even under repeated use and light concurrent access, it processes predictions smoothly and reliably. The current architecture is ideal for demos, prototypes, and pilot programs, with potential to scale through cloud or hardware enhancements.