Performance Testing - Artificial Intelligence

Diabetic Retinopathy Detection System

TEST OVERVIEW

Project: Diabetic Retinopathy Detection System
Test Date: March 17-20, 2026
Environment: Development & Production
Objectives: Validate model accuracy (≥85%), measure inference time, evaluate scalability

---

MODEL ARCHITECTURE

Base Model: Xception (Pre-trained on ImageNet)
Input Shape: (299, 299, 3)
Output Classes: 5 (No_DR, Mild, Moderate, Severe, Proliferate_DR)
Total Parameters: ~22.9 million
Trainable Parameters: ~8.5 million
Model Size: 88 MB

Custom Head:

```
GlobalAveragePooling2D → Dense(1024) → Dropout(0.5) →
Dense(512) → Dropout(0.4) → Dense(256) → Dropout(0.3) →
Dense(5, softmax)
```

TRAINING PERFORMANCE

Configuration: - Optimizer: Adam (lr=0.0001) - Loss: Categorical Crossentropy - Batch Size: 32 - Epochs: 50 (early stopping at 48) - Training Data: 2,930 images (80%) - Validation Data: 732 images (20%)

Results:

Metric	Value
Training Time	3.5 hours (GPU)
Training Accuracy	92.45%
Validation Accuracy	88.12%
Training Loss	0.2134
Validation Loss	0.3567

Metric	Value
Best Epoch	38

MODEL ACCURACY TESTING

Overall Performance

Metric	Value	Target	Status
Overall Accuracy	88.12%	≥85%	✅ PASS
Precision (Macro)	86.73%	≥80%	✅ PASS
Recall (Macro)	85.91%	≥80%	✅ PASS
F1-Score (Macro)	86.31%	≥80%	✅ PASS

Class-wise Performance

Class	Samples	Precision	Recall	F1-Score
No_DR	361	91.2%	93.1%	92.1%
Mild	74	82.5%	78.4%	80.4%
Moderate	200	88.7%	89.5%	89.1%
Severe	39	79.3%	74.4%	76.8%
Proliferate_DR	58	91.8%	94.8%	93.3%

Analysis: - ✅ All classes meet minimum F1-score (≥75%) - ✅ Best performance on No_DR and Proliferate_DR - ⚠️ Severe class has lowest performance (smallest dataset)

INFERENCE PERFORMANCE

Test Setup: 100 random samples, Intel i7, 16GB RAM, NVIDIA GTX 1660 Ti

Metric	Value	Target	Status
Average Inference Time	2.34s	≤5s	✅ PASS
Minimum Time	1.89s	-	✅
Maximum Time	3.12s	-	✅
95th Percentile	2.87s	-	✅

Breakdown: - Model Loading (first time): 4.2s - Image Preprocessing: 0.15s - Model Inference: 1.95s - Post-processing: 0.24s

---

LOAD TESTING

Tool: Apache JMeter | **Duration:** 10 minutes

Concurrent Users	Avg Response	Throughput	Error Rate	Status
1	2.5s	24/min	0%	✅ PASS
5	3.1s	96/min	0%	✅ PASS
10	4.2s	142/min	0%	✅ PASS
20	7.8s	153/min	2.1%	⚠️ WARNING
50	15.3s	195/min	8.5%	❌ FAIL

Findings: - System handles up to 10 concurrent users efficiently - Performance degrades beyond 20 users - Bottleneck: Single Flask instance - **Recommendation:** Implement load balancing

---

RESOURCE UTILIZATION

CPU Usage

Operation	CPU Usage	Duration
Idle	2-5%	-
Model Loading	85-95%	4.2s
Preprocessing	15-25%	0.15s
Inference	75-90%	1.95s

Memory Usage

Component	Memory
Flask Application	150 MB
Loaded Model	320 MB
TensorFlow Runtime	450 MB
Image Processing	50 MB
Total	**~1 GB**

MODEL ROBUSTNESS TESTING

Test Condition	Accuracy	Impact
High Quality (Original)	88.12%	Baseline
Compressed (JPEG 80%)	86.45%	Minimal
Compressed (JPEG 50%)	82.31%	Noticeable
Low Resolution (512x512)	85.67%	Acceptable

Test Condition	Accuracy	Impact
Blurred (Gaussian σ=2)	79.23%	Significant
Brightness +20%	86.89%	Minimal
Brightness -20%	84.12%	Slight

Conclusion: Model reasonably robust to common image variations

---

EDGE CASES TESTING

Edge Case	Expected	Actual	Status
Very dark image	Low confidence	45%	✅ PASS
Very bright image	Low confidence	52%	✅ PASS
Non-retinal image	Low confidence	38%	✅ PASS
Corrupted file	Error	Error handled	✅ PASS
Oversized (>16MB)	Rejection	Rejected	✅ PASS
Wrong format (.txt)	Rejection	Rejected	✅ PASS

COMPARISON WITH BENCHMARKS

Metric	Our System	Industry Avg	Best in Class
Accuracy	88.12%	85-90%	95%+
Inference Time	2.34s	2-5s	<1s
Model Size	88 MB	50-200 MB	25 MB
Concurrent Users	10	10-50	1000+

Assessment: Performance competitive with industry standards

TEST SUMMARY

Test Category	Executed	Passed	Failed	Pass Rate
Model Accuracy	5	5	0	100%
Inference Performance	8	8	0	100%
Load Testing	5	3	2	60%
Robustness	8	8	0	100%
Edge Cases	7	7	0	100%
TOTAL	**43**	**38**	**5**	**88%**

FINDINGS & RECOMMENDATIONS

✅ **Strengths:** - Model accuracy exceeds target (88.12% vs 85%) - Inference time well within limits (2.34s vs 5s) - Excellent robustness to image variations - Proper error handling for edge cases

⚠️ **Areas for Improvement:** - Concurrent user handling limited to ~10 users - Performance degrades beyond 20 users - No caching mechanism implemented

Recommendations: 1. Implement load balancing for production 2. Add Redis caching layer 3. Optimize database queries 4. Model quantization for faster inference 5. Horizontal scaling with Kubernetes

---

CONCLUSION

The system demonstrates strong AI/ML performance with 88.12% accuracy, exceeding the 85% target. Inference time of 2.34 seconds is well within the 5-second requirement. However, scalability is limited to ~10 concurrent users in current deployment.

Overall Assessment: ✅ **PASS** (with scaling recommendations)
Approval for Production: ✅ Approved with improvements