Technology Stack

Diabetic Retinopathy Detection System

ARCHITECTURE OVERVIEW

Three-tier web application combining deep learning with cloud storage: - **Presentation Tier:** Web-based UI (HTML5, CSS3, Bootstrap, JavaScript) - **Application Tier:** Flask backend with ML model - **Data Tier:** IBM Cloudant database and file storage

---

MACHINE LEARNING & AI

Deep Learning Framework

TensorFlow 2.15.0 with Keras - Industry-standard framework with excellent documentation - Keras high-level API for rapid development - GPU acceleration support - Used for model development, training, and inference

Pre-trained Model

Xception Architecture - State-of-the-art image classification performance - Pre-trained on ImageNet (1.4M images) - Input size: 299x299 pixels - Last 20 layers unfrozen for fine-tuning - Custom classification head with dropout layers

---

BACKEND TECHNOLOGIES

Web Framework

Flask 2.3.0 - Lightweight Python framework - Easy ML model integration - RESTful API support - Features: Route handling, session management, file upload, template rendering

Programming Language

Python 3.8+ - Native ML/AI library support - Rich data science ecosystem - Key libraries: NumPy, Pillow, Werkzeug

---

FRONTEND TECHNOLOGIES

HTML5 - Semantic markup, form validation
CSS3 - Responsive styling, animations

JavaScript (ES6+) - Client-side interactivity
Bootstrap 5 - Responsive UI components, mobile-first design

DATABASE & STORAGE

Cloud Database

IBM Cloudant - Managed NoSQL database - JSON document storage with RESTful API - Automatic scaling and high availability - Collections: Users (authentication), Predictions (results)

File Storage

Local File System - `/uploads` - User-uploaded images - `/model` - Trained model files (.h5) - `/static` - Static assets (CSS, JS, images) - Secure filename handling, file type validation, 16MB size limit

---

DATA PROCESSING

Image Processing

TensorFlow Keras Preprocessing - Rescaling (0-1 normalization) - Data augmentation: rotation (±20°), flips, zoom (20%), shift, brightness

Pillow (PIL) - Image loading and manipulation - Format conversion and resizing

---

MODEL TRAINING

Training Infrastructure: Local/Cloud GPU with CUDA support
Optimizer: Adam (lr=0.0001)
Loss Function: Categorical crossentropy
Callbacks: ModelCheckpoint, EarlyStopping (patience=10), ReduceLROnPlateau (patience=5)
Training Time: 2-4 hours with GPU

SECURITY & AUTHENTICATION

Flask Sessions - Secure session cookies with encryption
Werkzeug Security - Secure filename handling, file validation
Input Validation - File type/size checking, form data sanitization

---

DEPLOYMENT OPTIONS

1. **IBM Cloud** - Cloud Foundry with integrated Cloudant
2. **Local Server** - On-premise with full control
3. **Docker Container** - Portable containerized deployment

DATASET

Diabetic Retinopathy Dataset - Total: 3,662 fundus images (PNG format) - Distribution: - No_DR: 1,805 (49.3%) - Moderate: 999 (27.3%) - Mild: 370 (10.1%) - Proliferate_DR: 294 (8.0%) - Severe: 193 (5.3%) - Split: 80% training, 20% validation

---

CORE DEPENDENCIES

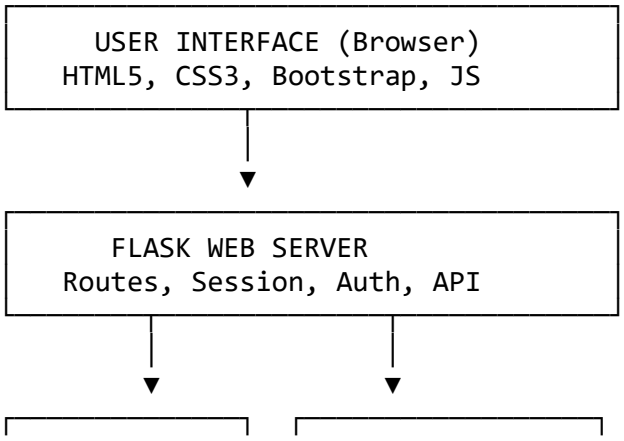
```
flask==2.3.0
tensorflow==2.15.0
numpy==1.24.3
pillow==10.0.0
werkzeug==2.3.0
cloudant==2.15.0
```

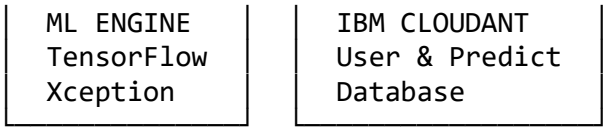
SYSTEM REQUIREMENTS

Minimum: - OS: Windows 10, macOS 10.14+, Linux (Ubuntu 18.04+) - RAM: 8GB (16GB for training) - Storage: 10GB free space - Python: 3.8+ - Browser: Chrome, Firefox, Safari, Edge (latest)

Recommended for Training: - GPU: NVIDIA with CUDA support - RAM: 16GB+ - Storage: SSD

ARCHITECTURE DIAGRAM





TECHNOLOGY SELECTION RATIONALE

Proven Technologies: Industry-standard with extensive documentation
Rapid Development: High-level APIs, pre-built components
Scalability: Cloud-native, horizontal scaling
Cost-Effective: Open-source frameworks, free tier services
Performance: GPU acceleration, efficient architecture (<5s inference)

FUTURE ENHANCEMENTS

- Model quantization (TensorFlow Lite)
- RESTful API development
- Docker containerization
- CI/CD pipeline
- Redis caching
- Nginx load balancing
- PostgreSQL for relational data