

Binary Image Classification MLOps Pipeline

Cats vs Dogs Classification for Pet Adoption Platform

This project implements an end-to-end MLOps pipeline for binary image classification (Cats vs Dogs) using open-source tools. The pipeline covers model development, experiment tracking, containerization, CI/CD, and monitoring.

🏗️ Project Structure

```
└── .github/
    └── workflows/          # GitHub Actions CI/CD pipelines
    ├── data/
    │   ├── raw/             # Original dataset
    │   ├── processed/       # Preprocessed 224x224 images
    │   └── .gitkeep
    ├── src/
    │   ├── data/            # Data processing scripts
    │   ├── models/           # Model architectures
    │   ├── training/         # Training scripts
    │   ├── inference/        # Inference service
    │   └── utils/            # Utility functions
    ├── tests/               # Unit and integration tests
    ├── notebooks/           # Jupyter notebooks for exploration
    ├── configs/              # Configuration files
    ├── models/               # Saved model artifacts
    ├── mlruns/                # MLflow tracking
    ├── k8s/                  # Kubernetes manifests
    ├── monitoring/           # Prometheus/Grafana configs
    ├── Dockerfile             # Container definition
    ├── docker-compose.yml     # Local deployment
    ├── requirements.txt        # Python dependencies
    ├── dvc.yaml               # DVC pipeline
    └── .dvc/                  # DVC configuration
```

🚀 Quick Start

Prerequisites

- Python 3.10+
- Docker & Docker Compose
- Git & DVC
- Kubernetes (minikube/kind) - optional

Installation

```
# Clone the repository
git clone
https://github.com/vishalvishal099/BinaryImageClassification_For_A_Pet_Adoption_Platform.git
cd BinaryImageClassification_For_A_Pet_Adoption_Platform

# Create virtual environment
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Pull data from Dagshub
dvc pull # Downloads ~1.4GB of images from Dagshub
```

Data Preparation

```
# Download dataset from Kaggle
kaggle datasets download -d bhavikjikadara/dog-and-cat-classification-dataset
unzip dog-and-cat-classification-dataset.zip -d data/raw/

# Run preprocessing pipeline
python src/data/preprocess.py

# Version data with DVC
dvc add data/processed
git add data/processed.dvc
git commit -m "Add processed dataset"
```

Model Training

```
# Train the model with MLflow tracking
python src/training/train.py --config configs/train_config.yaml

# View experiments in MLflow UI
mlflow ui --port 5000
```

Running the Inference Service

```
# Using Docker
docker build -t cats-dogs-classifier:latest .
docker run -p 8000:8000 cats-dogs-classifier:latest

# Using Docker Compose
docker-compose up -d

# Test the API
curl http://localhost:8000/health
curl -X POST -F "file=@test_image.jpg" http://localhost:8000/predict
```

Running Tests

```
# Run all tests
pytest tests/ -v

# Run with coverage
pytest tests/ --cov=src --cov-report=html
```

🔧 Configuration

Training Configuration ([configs/train_config.yaml](#))

- Batch size, learning rate, epochs
- Model architecture selection
- Data augmentation settings

Deployment Configuration

- Kubernetes: [k8s/deployment.yaml](#), [k8s/service.yaml](#)
- Docker Compose: [docker-compose.yml](#)

📊 API Endpoints

Endpoint	Method	Description
/health	GET	Health check endpoint
/predict	POST	Predict cat/dog from image
/metrics	GET	Prometheus metrics

🛠 CI/CD Pipeline

GitOps Flow (CI → CD → ArgoCD)



Service URLs

Service	URL	Notes
FastAPI	http://localhost:8000	Inference API
API Docs	http://localhost:8000/docs	Swagger UI
MLflow	http://localhost:5001	Experiment tracking
Prometheus	http://localhost:9090	Metrics scraping
Grafana	http://localhost:3000	ML monitoring dashboard
Grafana Dashboard	http://localhost:3000/d/pet-adoption-ml-v2	16-panel ML dashboard
ArgoCD UI	https://localhost:9443	GitOps CD (admin/ <code>wbYZNPLoKU4h-a0d</code>)

Service	URL	Notes
GitHub Repo	https://github.com/vishalvishal099/BinaryImageClassification_For_A_Pet_Adoption_Platform	Source
Dagshub	https://dagshub.com/vishalvishal099/BinaryImageClassification_For_A_Pet_Adoption_Platform	DVC + MLflow remote

One-command startup

```
bash start_all.sh
```

Starts: MLflow · FastAPI · Prometheus · Grafana · Metrics pusher

Monitoring

- **Prometheus** (port 9090): Scraps metrics from `/metrics` on ports 8000 and 8081
- **Grafana** (port 3000): 16-panel ML dashboard — requests, predictions, accuracy, latency, errors, day/time metrics
- **MLflow** (port 5001): Experiment tracking, model registry, artifact storage
- **Metrics server (`scripts/push_metrics.py`)**: Exposes Prometheus-format ML metrics on port 8081