Machine Learning Operations Canvas (v1.1)

Product name:	Designed by:	Date:	Iteration:
AI/Real Image Classification	Group 75	09/01/2026	1

Problem

Background

The project develops a binary classification system to distinguish AI-generated (synthetic) images from authentic (real) photographs. This ML effort aims to improve the ability to verify image authenticity, supporting applications in content analysis and quality control.

Value Proposition

The solution reliably detects synthetic images, enhancing trust in visual content and aiding in quality assurance. Benefits: Supports content platforms in verification, helps users identify real vs. AI images, enables integration for automated checks.

Objectives

Goals: ≥95% test accuracy; compare ResNet variants and DIRE. Outcomes: Deployable robust classifier. Criteria: Outperform baselines, low FP/FN, via confusion matrices/ROC.

Data

Data Collection

Data sources include the Kaggle competition dataset, featuring authentic images from Shutterstock paired with AI-generated counterparts using state-of-the-art models. Methods involve downloading the dataset. Volume: 79,950 training images and 19,986 test images. Labeling process: Binary labels (0 = Real, 1 = AI-generated) provided in train.csv and test.csv. Frequency: One-time static collection.

Data Verification and Governance

Data management policies: Ensure quality through checks for corrupt files, label accuracy, and class balance (50/50 real/AI). Privacy and compliance: Adhere to dataset licenses (Shutterstock/Kaggle); no personal data involved. Mechanisms: Access controls via secure storage, automated quality scripts for validation, and compliance monitoring through version tracking.

Model

Modelling

Use ResNet models (e.g., ResNet50) pretrained on ImageNet, fine-tuned for binary classification. Also evaluate DIRE using diffusion model reconstruction errors. Features: Resize to 224x224, normalize, augment. Transfer learning for ResNet; DIRE as non-neural complement.

Metrics and Evaluation

Metrics: Accuracy, Precision, Recall, F1, AUC-ROC. Methods: 5-fold CV on train, test set eval. Effectiveness: Beat 50% baseline, aim >90% accuracy vs. benchmarks.

Model Governance

Process for managing versions: Use Weights & Biases for tracking experiments and artifacts. Conditions from staging to production: Pass validation thresholds. Procedures: Automated pipelines for retraining on new data, with rollback options for failed updates.

Operations

Inference

Deployment process: Package models into a service using Docker for API-based inference. Include details on preprocessing pipelines and batch handling.

Infrastructure: Cloud-based with GPU support for efficient processing.

Environment: Python with PyTorch, scalable to handle variable loads.

Decision

Predictions integrated: Output probability scores for AI-generated; threshold for binary decisions. Human oversight: In high-stakes scenarios, flag uncertain cases for manual review. Automated systems: Direct integration into workflows like content filters, with logging for audits.

Monitoring

Feedback

Since this is a academic project without live users, feedback is collected via: automated validation on hold-out sets and cross-validation folds, error analysis on misclassified samples periodic benchmarking against new generators or updated test splits, and qualitative review of failure modes . How used: Insights drive iterative improvements . In a future production extension, this would expand to logged production misclassifications and user-reported flags.

Lifetime

Outline after deployment: Continuous monitoring for drift via periodic re-evaluation on new data. Conditions for retraining: If metrics drop >5% or new generators emerge. For decommissioning: If accuracy falls irrecoverably or replaced by superior models, with graceful shutdown and data archival.

Risk

Fairness

Check biases across categories (humans vs. others); avoid errors from imbalanced data or biased generators. Use equalized odds metrics, augment data, perform audits.

Explainability

Use Grad-CAM for ResNet visuals; error maps for DIRE. Add SHAP/LIME. Provide explanations like "AI due to textures" for trust.

Security

Identify risks: Adversarial attacks, data breaches in storage. Adversarial attacks, system vulnerabilities: Model poisoning or inference-time exploits. Include measures: Adversarial training, input sanitization, secure APIs, and regular vulnerability scans to ensure robustness.

DTU

By Nicki Skafte Detlefsen nsde@dtu.dk
From DTU course 02476 Machine Learning Operations