Federated Learning for Industrial Anomaly Detection

Stage 1: Baseline Development & Independent Client Setup

AI for Trustworthy Decision Making

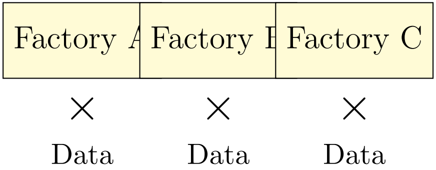
AutoVI Dataset Project

November 26, 2025

The Challenge: Privacy in Industrial Quality Control

Current Problem:

- Manufacturing data is proprietary
- Different factories = isolated data
- Regulatory restrictions on data sharing
- Cannot train shared models easily



Solution: Federated Learning

✓ Train models **locally**, share only **model updates**

✓ No raw data leaves the facility

Dataset: Automotive Visual Inspection (AutoVI)

Real industrial data from Renault Group

6 Product Categories:

- engine_wiring (285 train)
- pipe_clip (195 train)
- pipe_staple (191 train)
- tank_screw (318 train)
- underbody_pipes (161 train)
- underbody_screw (373 train)

Dataset Statistics:

Metric	Value
Total Images	3,950
Training Images	1,523
Test Images	2,399
Categories	6
Defect Types	10

% TODO: Insert image grid (6 categories)

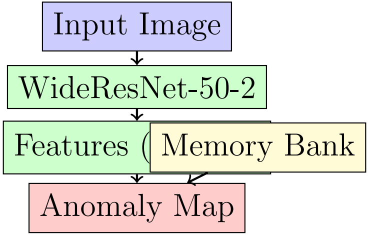
- **Unsupervised setup**: Train on "good" images only
- **Real world**: Lighting variation, authentic defects
- **Pixel annotations**: Ground truth masks for localization

Baseline Model: PatchCore Architecture

Memory Bank-Based Anomaly Detection

1. **Feature Extraction**
   - Pre-trained WideResNet-50-2 backbone
   - Multi-scale features (Layer 2 + Layer 3)
   - Output: 1536-dim patch embeddings
2. **Memory Bank (Coreset)**
   - Greedy selection (10% of patches)
   - Represents "normal" feature space
   - Computationally efficient
3. **Anomaly Scoring**
   - Distance to nearest bank feature
   - Upsampled to pixel resolution

% TODO: Architecture diagram



Federated Architecture: 5 Clients with IID Partitioning

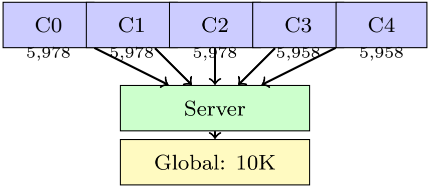
Balanced data distribution across clients

Client	Samples	Patches	Coreset (10%)
Client 0	305	59,780	5,978
Client 1	305	59,780	5,978
Client 2	305	59,780	5,978
Client 3	304	59,584	5,958
Client 4	304	59,584	5,958
Total	**1,523**	**298,508**	**29,850**

IID Partitioning: Each client receives samples from all 6 categories

- **Balanced distribution**: ∼305 samples per client
- Each client has **mixed categories** (engine_wiring, pipe_clip, etc.)
- Local coreset selection: **10%** of extracted patches

Federated Memory Bank Aggregation



Local Processing:

- Extract 784 patches/image
- Coreset: 10% selection
- Total: 29,850 patches

Server Aggregation:

- Weighted 2x oversampling
- Diversity selection
- Final: 10,000 patches

Compression: 298K → 30K → 10K (**96.6% reduction**)

Aggregation Strategies

Strategy	Description	Trade-off	Used
Federated Coreset	Weighted + oversampling + diversity	Fairness + diversity	✓
Simple Concatenate	Direct merge	Fast, no fairness	
Diversity Preserving	Min. 100 patches/client	Small client voice	

Federated Coreset Algorithm:

1. Weight contributions by sample count
2. Oversample 2x from each client
3. Apply greedy coreset selection

Client Weights (IID): ∼0.20 each

Only embeddings shared – raw images never leave clients

Experimental Setup & Training Statistics

Configuration:

- Clients: 5 (IID)
- Backbone: WideResNet50-2
- Features: layer2 + layer3
- Dimension: 1536
- Image: 224×224

Results:

- Time: 173 seconds
- Rounds: 1 (one-shot)
- Local coreset: 10%
- Global bank: 10,000
- Strategy: federated_coreset

Data Flow:

Stage	Patches	Kept
Raw extraction	298,508	100%
Local coreset	29,850	10%
Global coreset	10,000	3.4%

Evaluation Metrics: AUC-sPRO and AUC-ROC

AUC-sPRO (Localization)

- Pixel-level accuracy
- Multiple FPR thresholds:
  - @0.01 (strict)
  - @0.05 (intermediate)
  - @0.1 (moderate)
  - @0.3 (permissive)
- Saturated to prevent over-crediting

AUC-ROC (Classification)

- Image-level detection
- Binary: Good vs Anomalous
- Standard metric
- Range: [0, 1]
- Higher is better

% TODO: Figure – FPR-sPRO curves for different categories

Expected visualization: Multiple curves comparing categories

- Performance varies by product complexity

Stage 1 Results: Federated Training Complete

Training Statistics (Completed):

Client	Samples	Patches	Coreset	Weight
Client 0	305	59,780	5,978	0.200
Client 1	305	59,780	5,978	0.200
Client 2	305	59,780	5,978	0.200
Client 3	304	59,584	5,958	0.200
Client 4	304	59,584	5,958	0.200
Total/Avg	**1,523**	**298,508**	**29,850**	**1.000**

Evaluation Metrics (TODO):

Category	@FPR=0.01	@FPR=0.05	@FPR=0.1	@FPR=0.3	AUC-ROC
engine_wiring	TODO	TODO	TODO	TODO	TODO
pipe_clip	TODO	TODO	TODO	TODO	TODO
pipe_staple	TODO	TODO	TODO	TODO	TODO
tank_screw	TODO	TODO	TODO	TODO	TODO
underbody_pipes	TODO	TODO	TODO	TODO	TODO
underbody_screw	TODO	TODO	TODO	TODO	TODO

Key Achievement: Global memory bank (10,000 patches) trained in

Stage 1 Implementation Status

Component	Status	Notes
Data Loader	✓ Complete	All 6 categories loaded
Data Partitioning	✓ Complete	5 clients, IID distribution
PatchCore Model	✓ Complete	WideResNet50-2 backbone
Federated Training	✓ Complete	173s, global bank ready
Server Aggregation	✓ Complete	federated_coreset strategy
Baseline Evaluation	**In Progress**	AUC metrics pending

Completed Outputs:

- Global memory bank: 10,000 patches (1536-dim)
- Training logs and statistics saved
- Checkpoint for reproducibility

Next Steps:

Stage 2 Preview: Trust-Focused Enhancements

Building on Stage 1 baselines...

Privacy Enhancement

- Differential Privacy (DP-SGD)
- Formal privacy guarantees
- Privacy budgets: $\varepsilon = 1, 5, 10$
- Measure privacy-utility trade-off

Aggregation Strategy

- Memory bank pooling (1 communication round)
- Weighted coreset selection
- Fairness-aware weighting

Fairness Enhancement

- Address data imbalance
- Reduce performance variance
- Cross-category equity
- Client contribution weighting

Analysis

- Statistical significance testing
- Trade-off Pareto frontiers
- Recommendations by use case

Key Takeaways & Project Roadmap

Stage 1 Achievements:

- ✓ Complete data infrastructure (6 categories, 3,950 images)
- ✓ Federated architecture (5 clients, IID partitioning)
- ✓ PatchCore model with federated aggregation complete
- ✓ Global memory bank trained (10,000 patches, 173s)

Stage 2 Objectives:

- Add privacy guarantees (Differential Privacy)
- Implement fairness mechanisms
- Demonstrate one-round efficient aggregation
- Provide trade-off analysis

Final Deliverables:

- Technical Report (18-20 pages)

Questions?

Dataset: doi.org/10.5281/zenodo.10459003

Code: [GitHub Repository]

Contact information for team members

Backup: Detailed Dataset Statistics

Category	Train	Test Total	Test Good	Test Anom	Defect Types	Size
engine_wiring	285	607	285	322	4	400×400
pipe_clip	195	337	195	142	2	400×400
pipe_staple	191	305	188	117	1	400×400
tank_screw	318	413	318	95	1	1000×750
underbody_pipes	161	345	161	184	3	1000×750
underbody_screw	373	392	374	18	1	1000×750
Total	**1,523**	**2,399**	**1,521**	**878**	**10**	-

- Small images (400×400): 671 train, 1,249 test
- Large images (1000×750): 852 train, 1,150 test
- Total defect types: 10 (mix of structural and logical)

Backup: PatchCore Algorithm Details

Greedy Coreset Selection:

1. Start with all patches extracted from training images
2. Randomly select first patch
3. Iteratively add patch maximizing minimum distance to selected set
4. Continue until target size (10% of total patches)

Memory Bank Construction:

- Coreset represents normal feature distribution
- Size: 10% of all extracted patches
- Stored as feature vectors (1536-D)
- Enables efficient nearest-neighbor search (FAISS)

Inference:

- Extract patches from test image

Backup: Federated Aggregation Strategy (Stage 2)

Why Memory Bank Aggregation?

- Unlike gradient FL: Only 1 communication round needed
- Not iterative: Feature banks, not parameters
- Efficient: Total 450 MB communication

Weighted Coreset Selection:

1. Weight contributions by local dataset size (fairness)
2. Oversample from each client ($2\times$ target allocation)
3. Apply global greedy coreset selection
4. Ensures diverse representation + balance

Client Imbalance Handling:

- Small clients: pipe_clip (195 images)
- Large clients: underbody_screw (373 images)