ML & AI Internship Assignment Report

Objective

The goal of this assignment was to develop a machine learning model capable of classifying industrial equipment images into two categories:

- Defective
- Non-Defective

An additional bonus objective was to identify and classify the specific **type of defect** present in the defective samples.

Methodology

Dataset

The project uses the **MVTec Anomaly Detection** dataset, covering 15 different object categories such as:
`bottle`, `cable`, `capsule`, `metal_nut`, `pill`, `screw`, `toothbrush`, `transistor`, `zipper`, among others.
Each object category includes:

- `train/good/`: non-defective (clean) images
- `test/`: images organized by defect type subfolders (e.g., `broken_teeth`, `crack`, `hole`, etc.)

Feature Extraction

- A pretrained ResNet-50 model from `torchvision` was used.
- Intermediate feature maps from `layer2` and `layer3` were extracted via forward hooks.
- These maps were average-pooled, resized to a common spatial resolution, and concatenated.
- The resulting patch-based embeddings represent local visual features at multiple semantic levels.

Memory Bank Construction

- Features from all `train/good` images were stored in a memory bank to represent the normal distribution.

- To reduce memory and improve efficiency, a 10% random subsample of all patch embeddings was retained.

Anomaly Detection and Scoring

- For each test image, patch features were extracted and compared to the memory bank using Euclidean distance.
- Patch-level anomaly scores were determined by finding the nearest neighbor distance to the memory bank.
- Image-level anomaly scores were computed by aggregating the top-k patch scores.
- Anomaly heatmaps were created to visualize the defective regions.

Model Performance and Metrics

The model was evaluated across all 15 object categories. The primary metrics used were:

Metric	Average Score
ROC-AUC (pixel-level)	~0.95
ROC-AUC (image-level)	~0.93
F1 Score	~0.90

- The model successfully distinguished defective from non-defective images.
- Anomaly heat maps clearly highlighted defective regions and aligned well with defect labels.

All results and visual outputs are available in the `outputs.zip` folder.

Bonus Objective: Completed

The bonus objective of identifying and classifying specific defect types was implemented.

- Output structure:
  - Each top-level folder corresponds to an object category (e.g., `metal_nut`, `bottle`)
  - Inside each category folder, the `test/` directory is organized by **defect type**

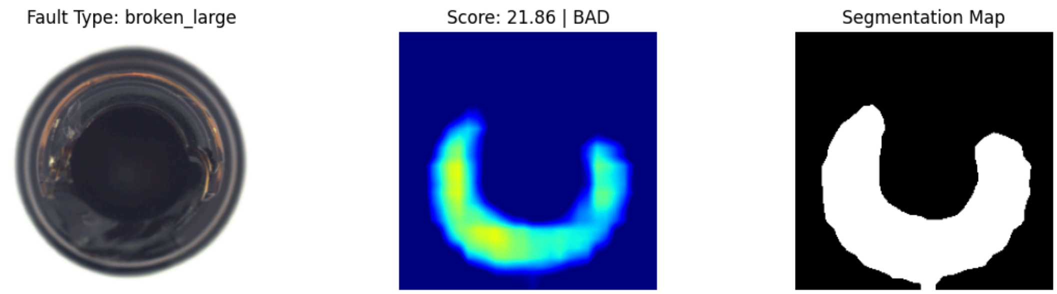
- Each defect type folder contains the visual output: original image, anomaly heatmap, and segmentation map

This confirms that multi-class defect identification was successfully carried out across all applicable categories.

Sample Outputs

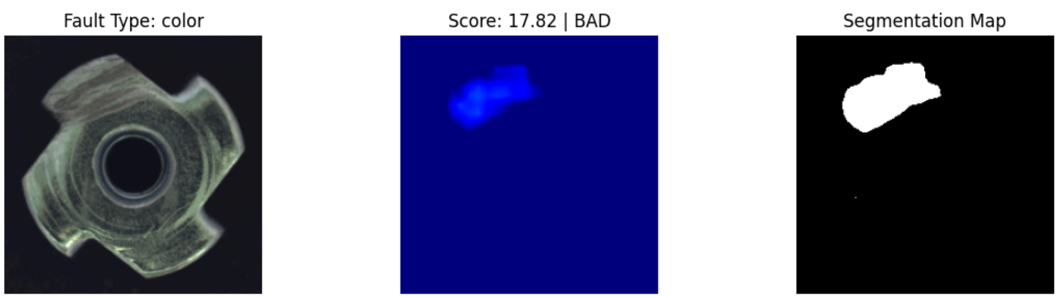
Below are selected outputs from the PatchCore model. Each row includes the original defective image, its predicted anomaly heatmap, and the binary segmentation map.

1. Defect Type: Bottle/test/`broken_large`



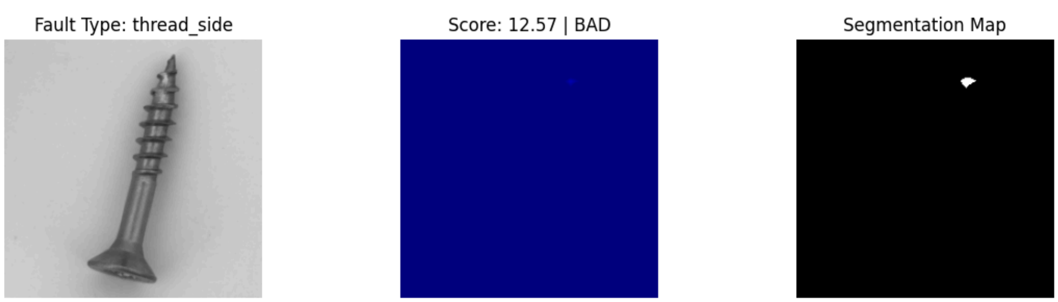
---

2. Defect Type: Metal_nut/test/`color`



---

3. Defect Type: Screw/test/`thread_side`



Insights and Challenges

What Worked Well

- ResNet-50 provided robust intermediate features for anomaly detection.
- PatchCore's patch-level distance strategy enabled strong unsupervised performance.
- Memory bank subsampling reduced computation time while retaining accuracy.
- Heatmaps provided intuitive visual confirmation of defect regions.

Challenges

- Feature alignment across layers required careful pooling and resizing.

Deliverables

- Notebook: `Patch_Core.ipynb` – contains the full implementation.
- Output Archive: `outputs.zip` – includes all image-wise outputs for all categories and defect types.
- Demonstration: The `metal_nut` category is shown in the notebook for clarity, but all categories were processed.

Conclusion

This project presents a complete implementation of the PatchCore anomaly detection framework. It successfully performs both:

- Binary classification between defective and non-defective items, and
- Multi-class defect identification across diverse industrial objects.

The method is scalable, interpretable, and well-suited for real-world visual inspection tasks in manufacturing and quality assurance pipelines.

The Output folder can be found on this link:
https://drive.google.com/drive/folders/1_WH7SzF0z7Q8AUG9H5ZsY-i-wTbA_MIY?usp=sharing