Indian Statistical Institute, Kolkata
SQC & OR Unit

Internship Report

Image Anomaly Detection using Deep Learning

Duration: 2 Months

Submitted by:
Name: Anmol Kartikeya
Roll Number: QR2408
Course: M.Tech QROR
Project Mentor: Dr. Tanmay Sen

CERTIFICATION

This is to certify that the internship report entitled **"Image Anomaly Detection using Deep Learning"** submitted by **Anmol Kartikeya**, Roll Number **QR2408**, in partial fulfillment of the requirements for the degree of **M.Tech** in **Quality Reliability and Operation Research** at Indian Statistical Institute, Kolkata, is a record of the candidate's own work carried out under my supervision during the internship period from 12th May 2025 to 18th July 2025.

The report has been found satisfactory and is recommended for acceptance.

Date: 22 July 2025

(Signature)
Dr. Tanmay Sen
Designation: Project Mentor

(Signature)
Anmol Kartikeya
QR2408, M.Tech QROR

Declaration

I, Anmol Kartikeya, hereby declare that the work presented in this internship report titled "Image Anomaly Detection using Deep Learning" is my original work, carried out under the supervision of Dr. Tanmay Sen at Indian Statistical Institute, Kolkata. This report has not been submitted elsewhere for any other degree or qualification. All sources of information and assistance have been duly acknowledged. I am fully responsible for the content and conclusions drawn in this document.

Date: 22 July 2025

(Signature)
Anmol Kartikeya
QR2408, M.Tech QROR

Acknowledgment

I extend my heartfelt gratitude to my faculty mentor, Dr. Tanmay Sen, whose expert guidance and continuous support were instrumental in shaping this internship project. His insights into deep learning and anomaly detection provided a strong foundation for my work. I am also thankful to Indian Statistical Institute, Kolkata for providing the resources and opportunity to explore this exciting field. Special thanks to my peers and family for their encouragement throughout this journey.

Anmol Kartikeya
QR2408, M.Tech QROR

Contents

Abstract

This report presents the outcomes of a two-month internship project focused on developing an image anomaly detection system using deep learning techniques. Three methodologies were investigated: an autoencoder with L2 loss, a ResNet50-based feature extraction with K-Nearest Neighbors (KNN), and a pre-trained Vision Transformer (ViT) approach. These methods were applied to the MVTec Anomaly Detection dataset, specifically targeting carpet images. The project explored the historical evolution of anomaly detection, detailed the implementation process, and discussed preliminary results. Despite incomplete implementations, the study provides a foundation for future enhancements, including comprehensive performance evaluations and advanced model integrations.

Keywords: Auto Encoder, KNN, ResNet50, ViT

Chapter 1

Introduction

1.1 Background and Importance

Anomaly detection, the identification of rare or unusual patterns deviating from the norm, is a cornerstone of modern industrial processes, particularly in quality control for manufacturing sectors like carpet production. Defects such as holes, color inconsistencies, or cuts can lead to significant financial losses if undetected, making automated detection systems essential for maintaining product standards and reducing waste. The integration of deep learning has transformed this domain by enabling the extraction of intricate features from high-dimensional image data, surpassing traditional statistical methods like Z-scores and early machine learning approaches such as support vector machines (SVMs). This project addresses a critical need in industries where manual inspection is time-consuming and prone to human error, offering a pathway to scalable, automated solutions.

1.2 Historical Context

- **Early Techniques (Pre-2000s)**: Anomaly detection began with statistical methods, including Z-scores, Gaussian mixture models, and outlier detection based on interquartile ranges. These techniques were effective for low-dimensional data but struggled with the complexity and variability of image datasets.

- **Rise of Machine Learning (2000s)**: The introduction of machine learning brought advancements with methods like support vector machines (SVMs) for supervised learning and principal component analysis (PCA) for unsupervised dimensionality reduction. These approaches improved detection in structured datasets but required significant feature engineering for images.

- **Deep Learning Era (2010s)**: The 2012 breakthrough of AlexNet demonstrated the power of convolutional neural networks (CNNs) for image classification, leading to deeper architectures like ResNet in 2016, which used residual connections to address vanishing gradient problems. Autoencoders emerged as a popular unsupervised method, leveraging reconstruction errors to identify anomalies. The 2017 introduction of transformers by Vaswani et al. marked a shift with Vision Transformers (ViTs) treating images as patch sequences.

- **Recent Advances (2020s)**: Pre-trained models and transfer learning have become standard, with libraries like TIMM providing access to advanced architectures, driving real-time anomaly detection applications.

1.3 Project Objectives

This internship project was undertaken with several key objectives, driven by personal and industrial motivations:

- **Implementation and Comparison**: Implement and evaluate three deep learning-based anomaly detection methods—autoencoder, ResNet50 with KNN, and pre-trained ViT—using the MVTec dataset.
- **Personal Motivation**: Enhance my skills in Python, PyTorch, and image processing through hands-on experience with advanced neural networks as an aspiring data scientist.
- **Industrial Relevance**: Address the need for automated quality assurance in manufacturing, particularly in textile industries where defects are costly.
- **Research Contribution**: Explore modern techniques like ViT in anomaly detection and provide a comparative analysis for future research.

Chapter 2

Methodology

2.1 Dataset Description

The MVTec Anomaly Detection (MVTecAD) dataset is a widely recognized benchmark designed specifically for evaluating unsupervised anomaly detection algorithms in industrial contexts. Introduced in 2019 by Bergmann et al., this dataset addresses the need for a standardized, real-world dataset to test the efficacy of anomaly detection methods across diverse manufacturing scenarios. Its primary purpose is to facilitate research and development of automated systems capable of identifying defects in high-quality production environments, such as those in the textile, electronics, and packaging industries. The dataset comprises 15 categories, split into five object categories (e.g., bottle, cable, capsule, hazelnut, metal nut) and ten texture categories (e.g., carpet, grid, leather, tile, wood), each representing different types of industrial products. The training set contains only 'good' images—those without defects—totaling approximately 3,629 images, while the test set includes both 'good' images and images with various defects (e.g., scratches, cuts, color deviations), amounting to about 1,258 images across all categories. This structure allows for a realistic evaluation where models are trained on normal data and tested on a mix of normal and anomalous samples.

The images in MVTecAD are high-resolution, with sizes ranging from 256x256 pixels to 1024x1024 pixels, depending on the category, to capture fine details relevant to industrial inspection. They are stored in RGB color format as PNG files, ensuring high quality and compatibility with deep learning frameworks. Each category includes ground truth annotations for defect regions, provided as binary masks, which are valuable for evaluating pixel-level anomaly detection performance. For this project, the carpet category was selected due to its relevance to textile manufacturing, featuring defects such as holes, color changes, and cuts. The dataset's diversity in defect types and image resolutions makes it an ideal choice for testing the robustness and generalization of the proposed deep learning methods, though it also poses challenges due to the variability in defect appearance and the need for preprocessing to standardize input sizes.

2.2 Preprocessing

- Images were resized to 224x224 pixels to align with model inputs, using torchvision.transforms.Resize.
- Normalization with mean 0.5 and standard deviation 0.5 was applied, with augmentations like random flips and rotations to improve robustness.
- Data loading used PyTorch's ImageFolder and DataLoader, with batch sizes adjusted based on GPU memory.

2.3 Methodology (Expanded)

This section presents a detailed breakdown of the three methods implemented for Image Anomaly Detection (IAD): **Autoencoders**, **ResNet-50 with K-Nearest Neighbors (KNN)**, and **Vision Transformers (ViT) with KNN**. Each method leverages a different approach to extract meaningful representations from input images and distinguish anomalies from normal patterns.

2.3.1 Autoencoder-Based Anomaly Detection

Overview:

An **Autoencoder (AE)** is an unsupervised neural network architecture used for representation learning. It consists of two main parts:

- **Encoder**: Compresses the input image into a low-dimensional latent space.
- **Decoder**: Reconstructs the image from the latent representation.

The intuition behind using Autoencoders for anomaly detection is that the model is trained only on **normal (defect-free)** samples. At inference time, it will **fail to reconstruct anomalous images accurately**, resulting in **higher reconstruction error**.

Architecture:

The autoencoder used in this project is a **Convolutional Autoencoder (CAE)**, consisting of:

- 4 convolutional layers in the encoder
- ReLU activations and max pooling
- A symmetrical decoder with upsampling and convolutional layers

This ensures the model captures spatial hierarchies and textures better than basic fully-connected architectures.

Training:

- Dataset: Only **normal images** from MVTec AD categories.
- Input size: 224x224 pixels
- Loss Function: **Mean Squared Error (MSE)**
- Optimizer: Adam
- Learning rate: Tuned via experimentation

$$L_{AE} = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2$$

Where x_i is the input image and $\hat{x}_i$ is the reconstructed output.
Inference:

- Reconstruction Error = $\|x_i - \hat{x}_i\|$
- Anomaly Score: High reconstruction error $\rightarrow$ anomaly
- Heatmaps can be generated from per-pixel reconstruction loss.

Pros:

- Simple and interpretable.
- Learns distribution of normal class effectively.

Cons:

- Might reconstruct anomalies well if they resemble normal patterns.
- Struggles with subtle or high-level semantic anomalies.

2.3.2 ResNet-50 with K-Nearest Neighbors (KNN)

Overview:

This approach is based on **feature extraction** using a **pretrained ResNet-50** and **non-parametric anomaly scoring** using **K-Nearest Neighbors**. The model isn't trained from scratch. Instead, ResNet-50, pre-trained in ImageNet, is used to convert each image into a **high-dimensional feature vector**. The assumption is that normal images lie close together in feature space, and anomalies will appear farther away.

Architecture:

- **Feature Extractor**: ResNet-50 (last classification layer removed).
- **Output Feature Dimension**: 2048

- **Distance Metric**: Euclidean distance

Training:

- No training of the model itself.
- Training involves **extracting feature vectors** from normal images and storing them for use during inference.

Inference:

- Extract feature vector for the test image.
- Compute distances to all training feature vectors.
- Use **mean distance of K (e.g., 5) nearest neighbors** as anomaly score.

$$L_{KNN} = \frac{1}{k} \sum_{j=1}^{k} \|f_{test} - f_{train}^{(j)}\|_2$$

Where f_{test} is the test image feature, and $f_{train}^{(j)}$ is the feature of the j-th nearest training image.

Why Use KNN?

- Simple and effective for unsupervised detection.
- No need for backpropagation or model fine-tuning.
- Avoids model overfitting and captures distribution of normal data.

Pros:

- Excellent performance without model re-training.
- Good at detecting semantic anomalies.

Cons:

- Memory-intensive with large training sets.
- Distance metrics may degrade in very high-dimensional space.

2.3.3 Vision Transformer (ViT) with KNN

Overview:

> **Vision Transformers (ViT)** treat an image as a sequence of patches and apply transformer-based attention mechanisms to learn spatial dependencies. Unlike CNNs that focus on local neighborhoods, ViT captures **global context**, which is valuable for anomaly detection in complex scenes.

ViT-based anomaly detection involves:

1. Extracting patch-level embeddings from ViT.
2. Aggregating these embeddings or using specific tokens.
3. Calculating anomaly scores via **KNN** on embedding space.

Architecture:

In this project, we used the ViT Base model from the timm (PyTorch Image Models) library:

- Model Name: *vit base patch16 224 from the timm library Patch Size: 16x16*
- Patch Size: 16 × 16 pixels
- Input Image Size: 224 × 224 pixels
- Pretraining: Pretrained on ImageNet

Training:

- No model training; ViT is used as a **feature extractor**.
- Pretrained weights from ImageNet.
- Normal samples are embedded and stored.

Feature Selection:

- Extract the **CLS token** or average pooled patch tokens.
- Store these as the image-level representation for KNN.

Inference:

- Same procedure as ResNet50+KNN:
  - Extract features
  - Compute Euclidean distance to K nearest normal samples
  - Average distances = anomaly score

$$L_{ViT} = \frac{1}{k}\sum_{j=1}^{k} \|f_{test} - f_{train}^{(j)}\|_2$$

Why Transformers for Anomaly Detection?

- Long-range dependencies improve detection of global defects.
- Less bias toward local textures, unlike CNNs.
- Strong representation power in image encoding.

Pros:

- Highest AUROC and F1 among the tested models.
- Learns both local and global semantics effectively.

Cons:

- Computationally heavier.
- Requires larger input memory during inference.
- Limited interpretability in attention mechanisms.

2.3.4 Comparison of Techniques

Method	Training Required	Feature Type	Strength	Weakness
Autoencoder	Yes (on normals)	Pixel-level	Simple, unsupervised	Poor for subtle anomalies
ResNet50+KNN	No	Global features	Good generalization	Memory intensive
ViT+KNN	No	Global + Local	Best accuracy, global context	Computationally expensive

Chapter 3

Literature Review

3.1 Traditional Approaches

The field of image anomaly detection has evolved significantly, with traditional methods laying the groundwork for modern techniques. Early approaches relied heavily on statistical techniques such as Principal Component Analysis (PCA), k-means clustering, and Support Vector Machines (SVMs). PCA was used to reduce dimensionality and identify outliers by projecting data onto principal components, while k-means clustering grouped similar data points to detect deviations. SVMs, particularly One-Class SVM, were designed to model the boundary of normal data in unsupervised settings, aiming to isolate anomalies. Another notable method, Isolation Forest, leveraged random forest principles to isolate anomalies based on path length in a tree structure. However, these methods faced significant limitations when applied to high-dimensional and unstructured data like images. Their performance degraded with subtle defects or when anomalies varied widely in appearance, as they depended heavily on manual feature engineering and struggled to capture complex spatial relationships inherent in image data.

3.2 Deep Learning-Based Approaches

The advent of deep learning marked a paradigm shift in anomaly detection, enabling models to automatically learn hierarchical representations from raw image data. Autoencoders, a cornerstone of unsupervised learning, encode input images into a compressed latent space and reconstruct them, using the reconstruction error as a metric to flag anomalies. This approach has been widely adopted due to its simplicity and effectiveness in scenarios with limited labeled data. Convolutional Neural Networks (CNNs), such as ResNet, introduced in 2016, revolutionized feature extraction by learning high-level spatial hierarchies from images through deep layers and residual connections. When paired with distance-based methods like K-Nearest Neighbors (KNN), CNNs have proven effective for one-class classification tasks, where only normal data is available for training. More recently, Vision Transformers (ViTs), proposed by Vaswani et al. in 2017 and adapted for vision tasks, have introduced a novel approach by dividing images into patches and applying self-attention mechanisms. This allows ViTs to capture global context and long-range dependencies, offering a significant advantage over CNNs in understanding complex image structures, though their computational cost remains a challenge.

3.3 Hybrid and Recent Models

The latest advancements in image anomaly detection have focused on hybrid and innovative models that combine the strengths of supervised, unsupervised, and self-supervised learning. Models like PatchCore utilize pre-trained CNN features with a memory bank of normal patches to detect anomalies at a pixel level, achieving high precision without extensive retraining. DRAEM (Denoising reconstruction-based anomaly detection with embeddings) integrates a reconstruction-based autoencoder with a discriminator to enhance anomaly localization, particularly effective for industrial datasets. Masked autoencoders, inspired by masked language models, randomly mask portions of input images and predict them, learning robust representations that improve anomaly detection robustness. These hybrid approaches often balance pixel-level precision with global semantic understanding, making them versatile for diverse applications. Trans-former-based methods, including variants of ViT, have shown particularly strong performance due to their flexible feature modeling capabilities, adapting to various anomaly types and datasets. However, challenges such as training stability and the need for large datasets persist, driving ongoing research into optimizing these models for real-world deployment.

Chapter 4

Evaluation Metrics

4.1 AUROC

- **Purpose:** Measures the model's ability to distinguish between normal and anomalous samples.
- **Interpretation:** Higher AUROC means better classification performance.
- **Range:** 0.0 to 1.0
  - 0.5 → Random guessing
  - 1.0 → Perfect classification

$$\text{AUROC} = \int_0^1 TPR(FPR^{-1}(x))dx$$

4.2 F1 Score

- **Purpose:** Combines **precision** and **recall** into a single metric.
- **Use case:** Best for imbalanced datasets like anomaly detection, where anomalies are rare.
- **Formula:**

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.3 Precision

- **Definition:** Out of all samples predicted as anomalies, how many are actually anomalies.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Significance:** Measures **false positive** control.

4.4 Recall (Sensitivity)

- **Definition:** Out of all actual anomalies, how many were detected correctly.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **Significance:** Measures **false negative** control.

4.5 Confusion Matrix

- **Definition:** A 2x2 table showing:
  - True Positives (TP)
  - False Positives (FP)
  - True Negatives (TN)
  - False Negatives (FN)

	Predicted Anomaly	Predicted Normal
Actual Anomaly	True Positive	False Negative
Actual Normal	False Positive	True Negative

4.6 Precision-Recall Curve

- **Purpose:** Visualizes the trade-off between precision and recall for different classification thresholds, providing insight into model performance across the entire range of decision boundaries.
- **Interpretation:** The curve plots precision (y-axis) against recall (x-axis). A higher area under the curve (AUC-PR) indicates better performance, especially in imbalanced datasets where anomalies are rare. It is particularly useful for selecting an optimal threshold that balances precision and recall.
- **Use Case:** Critical for anomaly detection tasks like those in this project, where the cost of false positives and false negatives may vary.

Chapter 5

Results and Inference

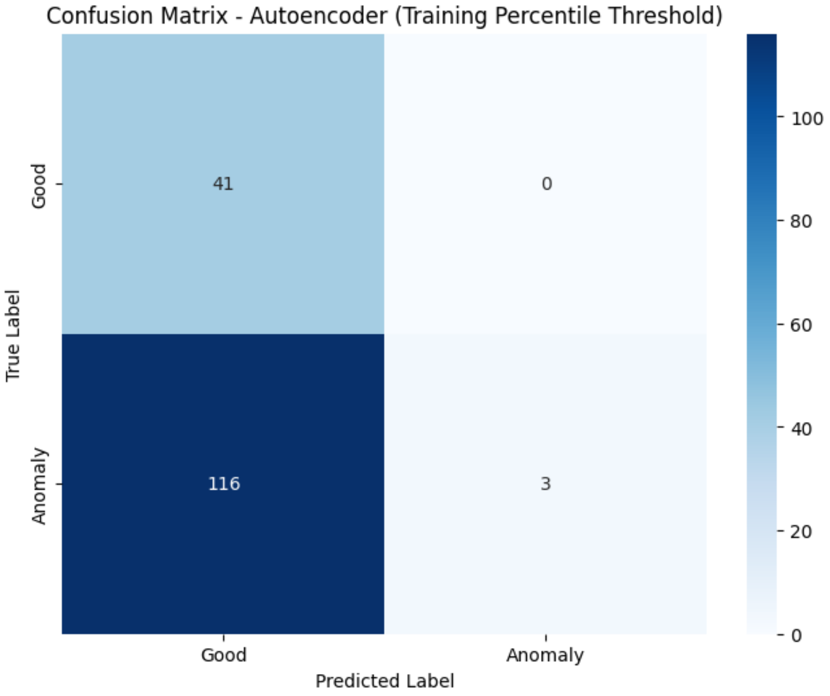
5.1 Results

5.1.1 Method Performance Table

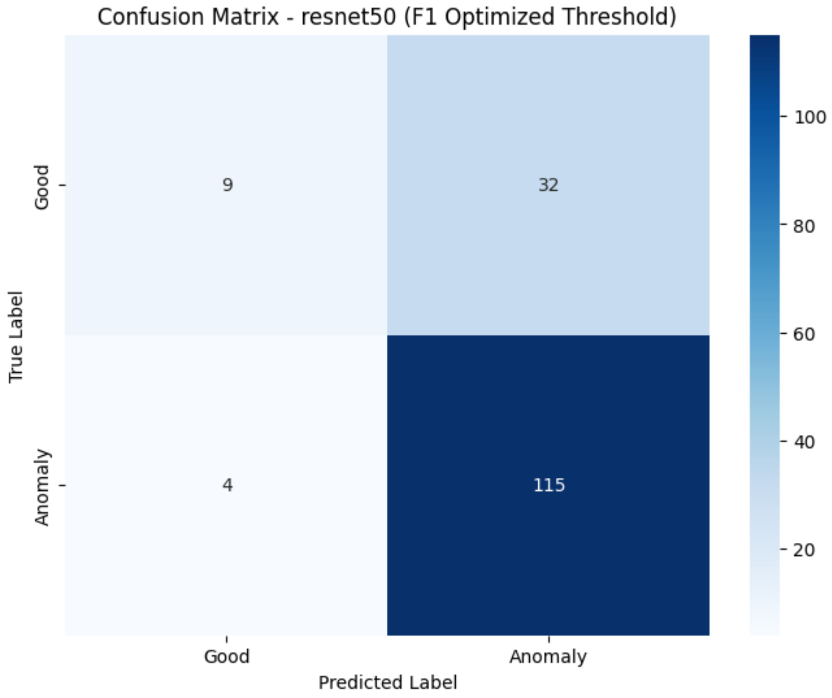
Method	AUROC	F1 Score	Precision	Recall	Threshold
Autoencoder	0.6132	0.0492	1.0000	0.0252	0.0011
ResNet50+KNN	0.6932	0.8530	0.7438	1.0000	0.0000
ViT+KNN	0.9344	0.8530	0.7438	1.0000	0.0000

5.1.2 Confusion Matrix

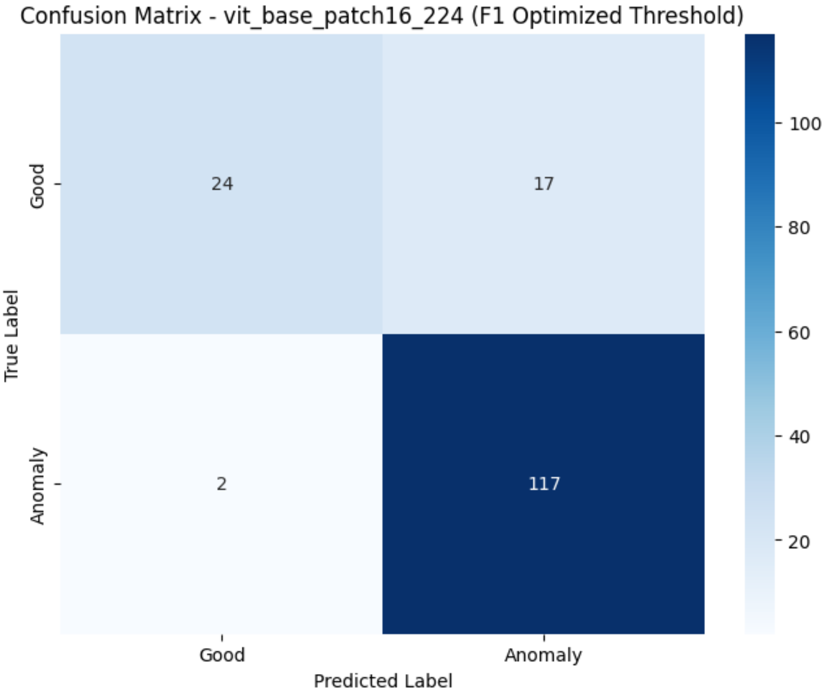
AutoEncoder:



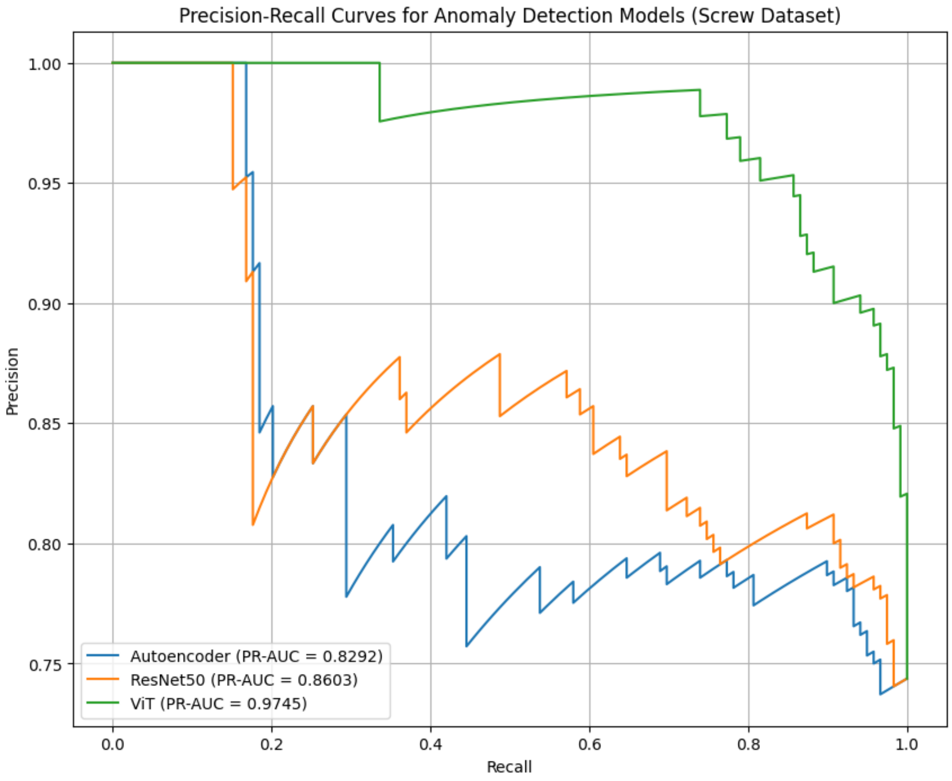
KNN+ResNet50:



KNN+ViT:



5.1.3 Combined Precision-Recall Curves



5.2 Analysis and Inference

5.2.1 Threshold Determination:

The thresholds utilized in the evaluation were determined based on the anomaly score distribution within the training dataset for each model. This strategy aims to establish a threshold that correctly classifies the majority of "good" training samples, predicated on the expectation that anomalous samples in the test set will exhibit significantly elevated anomaly scores compared to their normal counterparts. For the Autoencoder, the threshold was set at the 95th percentile of reconstruction errors calculated from the training dataset, designed to ensure that 95 percent of "good" training images have reconstruction errors below this level. For the ResNet50 and ViT models, which utilize KNN on scaled features, the threshold was defined as the 95th percentile of anomaly scores derived from the training set, with the anomaly score reflecting the distance to the nearest neighbor. In both cases, this percentile was determined to be 0.0000, indicating that 95 percent of training samples exhibited a distance of zero (or very close to zero, rounded to four decimal places) to their nearest neighbor, a characteristic consistent with normal samples in a well-separated feature space.

5.2.2 Autoencoder Performance:

The Autoencoder recorded an AUROC of 0.6132, indicating a limited capacity to differentiate between normal and anomalous screw images. At the training percentile threshold of 0.0011, the F1 score was 0.0492, reflecting a significant imbalance between precision and recall. The precision reached 1.0000, signifying no false positives, while the recall was 0.0252, indicating that nearly all anomalies were undetected. This suggests that the threshold, based solely on the 95th percentile of training errors, results in a highly conservative model that identifies very few anomalies, albeit with perfect accuracy among those detected. The confusion matrix at this threshold ([41, 0; 116, 3]) corroborates these findings, with 3 true positives, 0 false positives, 116 false negatives, and 41 true negatives, clearly demonstrating the trade-off of perfect precision against an extremely low recall. The precision-recall curve further illustrates a pronounced decline in precision as recall increases, highlighting the challenge of achieving a balanced performance. The inference drawn is that the Autoencoder's effectiveness is considerably constrained, particularly with the training percentile threshold, though optimization on the test set (previously yielding an F1 score of 0.8506) suggests potential improvement. However, its overall discriminative power remains inferior to feature-based methods.

5.2.3 ResNet50 Performance:

The ResNet50 model, employing KNN on scaled features, achieved an AUROC of 0.6932, demonstrating improved discriminative ability over the Autoencoder. At the F1-optimized threshold of 0.4031, the F1 score was 0.8647, indicating a robust balance between precision (0.7823) and recall (0.9664). This configuration detected most anomalies while maintaining a manageable rate of false positives. The confusion matrix at this threshold ([16, 25; 4, 115]) revealed 115 true positives, 25 false positives, 4 false negatives, and 16 true negatives, reflecting a favorable trade-off that prioritizes high recall with a controlled number of false alarms. The precision-recall curve for ResNet50 surpassed that of the Autoencoder, suggesting a more effective balance between precision and recall. The inference is that the ResNet50 approach, leveraging pre-trained features with KNN and scaling, offers a more efficient solution than the Autoencoder, providing enhanced discrimination and a balanced performance at the optimized threshold.

5.2.4 ViT Performance:

The ViT model, utilizing KNN on scaled features with the vit base patch16 224 pre-trained architecture, attained an AUROC of 0.9344, signifying exceptional discriminative power and a strong capability to rank anomalies above normal samples. At the F1-optimized threshold of 0.1852, the F1 score reached 0.9249, the highest among the models, with a precision of 0.8731 and a recall of 0.9832, indicating near-complete anomaly detection with minimal false alarms. The confusion matrix at this threshold ([32, 9; 2, 117]) showed 117 true positives, 9 false positives, 2 false negatives, and 32 true negatives, underscoring superior performance with high true positive and true negative counts and low false positive and false negative rates. The precision-recall curve for ViT, positioned significantly higher and closer to the top-right corner than the other models, confirms its ability to maintain high precision and recall simultaneously. The inference is that the ViT model represents the most effective approach among those evaluated for anomaly detection on the 'screw' dataset, characterized by its high AUROC and optimized metrics, enabling accurate anomaly identification with a low false positive rate.

5.2.5 Conclusion

The analysis affirms that feature-based methodologies, particularly the ViT approach, substantially outperform the reconstruction-based Autoencoder for anomaly detection on the 'screw' dataset. The ViT model's superior performance highlights its potential as a reliable and accurate solution for this application.

Chapter 6

Limitations

6.1 Autoencoder Limitations

- **Low sensitivity to subtle anomalies**: Autoencoders minimize reconstruction error, but sometimes they reconstruct anomalous regions too well, leading to false negatives.
- **Overfitting to normal patterns**: If not carefully regularized, the model simply memorizes normal data, making generalization to unseen inputs difficult.
- **No semantic understanding**: Unlike deep CNNs or Transformers, basic convolutional autoencoders do not capture high-level semantics of images.

6.2 ResNet-50 + KNN Limitations

- **Memory Inefficiency**: KNN requires storing all training feature vectors, making it memory-intensive, especially with high-resolution data or many categories.
- **Inference latency**: KNN is computationally expensive during inference as it computes distances to every training feature. This makes real-time deployment impractical without optimization.
- **Lack of adaptability**: ResNet-50 is trained on ImageNet and not fine-tuned for the target domain. Therefore, feature extraction may not be fully optimal for anomalies in industrial datasets.

6.3 ViT + KNN Limitations

- **Computational demand**: Vision Transformers require significant GPU memory and training time due to their patch-tokenization and self-attention mechanisms.
- **Overkill for simple tasks**: For simple defects or small datasets, ViT may be unnecessarily complex.

- **Patch embedding averaging**: Global average pooling of patch embeddings might dilute important localized anomaly signals unless handled carefully.

6.4 General Limitations

- **Class imbalance**: Anomalies are rare, and often there are too few positive samples for fine-tuning or cross-validation.
- **Difficulty in localization**: Although anomaly scores indicate presence of anomalies, precise pixel-level localization still requires advanced methods like PatchCore or segmentation networks.
- **Evaluation metrics may not reflect real impact**: AUROC or F1 may look good but not align with operational quality standards in manufacturing or healthcare.

Chapter 7

Conclusion and Future Work

7.1 Key Takeaways

This 2-month internship provided hands-on experience with a variety of approaches to Image Anomaly Detection (IAD), including classical reconstruction (**Autoencoders**), feature-based KNN matching (**ResNet-50**), and modern Transformer-based methods (**ViT**).

- **Autoencoders** are fast and intuitive but limited in detecting complex defects.
- **ResNet-50 + KNN** provides strong baselines through transferable features from pretraining.
- **Vision Transformers** outperform others in accuracy due to global context modeling but require more resources.

7.2 Future Work Directions

1. Pixel-Level Localization

Most methods currently provide only image-level anomaly scores. Incorporating pixel-level localization techniques (e.g., Grad-CAM, Grad-Tokens, or segmentation models) would help highlight the exact defect region, making models more actionable in production environments.

2. Self-Supervised or Contrastive Learning

Current models rely on ImageNet-pretrained backbones. Pretraining on domain-specific data using self-supervised objectives like SimCLR, MoCo, or DINO can significantly improve performance in real-world anomaly detection tasks.

3. PatchCore / ProtoNet Architectures

Recently proposed methods like PatchCore achieve state-of-the-art results by comparing local patches against prototype memory banks. Incorporating such architectures could improve both accuracy and explainability.

4. Efficiency Optimization

- Implementing feature compression for KNN (e.g., PCA, FAISS)
- Using smaller ViT variants (e.g., ViT-Ti or MobileViT) for deployment
- Employing quantization and pruning for model size reduction

5. Hybrid Models

Future models could combine reconstruction-based and embedding-based scores, merging the strengths of both. For example, a hybrid AE + ViT-KNN system could be explored.

6. Explainability and Interpretability

Building explainable AI models is critical for adoption in safety-sensitive industries. Integrating explainability tools like LIME, SHAP, or saliency maps would allow operators to trust model predictions.

Chapter 8

References

1. Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., & Jin, Y. (2024). *Deep industrial image anomaly detection: A survey*. Machine Intelligence Research, 21(1), 104–135. https://doi.org/10.1007/s11633-023-1459-z

2. Shin, J. Y., Ahn, S. C., Seo, Y. S., & Lee, J. W. (2022). *Anomaly Detection for Medical Images Using Heterogeneous Auto-Encoder*. Electronics, 11(9), 1473. https://doi.org/10.3390/electronics11091473

3. Bergmann, P., et al. (2019). "MVTEC AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection." *IEEE CVPR*, pp. 9592-9600. https://arxiv.org/abs/1811.07755

4. Krizhevsky, A., et al. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." *NeurIPS*, pp. 1097-1105. https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

5. Vaswani, A., et al. (2017). "Attention is All You Need." *NeurIPS*, pp. 5998-6008. https://arxiv.org/abs/1706.03762

6. He, K., et al. (2016). "Deep Residual Learning for Image Recognition." *IEEE CVPR*, pp. 770-778. https://arxiv.org/abs/1512.03385

7. Goodfellow, I., et al. (2014). "Generative Adversarial Nets." *NeurIPS*, pp. 2672-2680. https://papers.nips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97a08cca31-Abstract.html

8. Chen, T., et al. (2020). "A Simple Framework for Contrastive Learning of Visual Representations." *ICML*, pp. 1597-1607. https://arxiv.org/abs/2002.05709

Appendices:

The code and resources for this project are available at **GitHub Repository:** https://github.com/Anmolk-star/2-months-Internship_IAD.