UNIVERSITÀ
di VERONA

Department of Computer Engineering for Robotics and Smart Industries

Anomaly Detection and Classification Using Machine Learning Techniques on MVTec Dataset

Group Members:

1. Goitom Ybrah Hailu (Student ID: VR508738)
2. Yared Sereke Tedla (Student ID: VR509662)

Submitted To:

Professor Vittorio Murino
Professor Cigdem Beyan
Dr. Andrea Avogaro

A Project Report Submitted to Our Professors in Partial Fulfillment for Final Exam Project

June 24, 2024

Acknowledgements

First of all, we would like to express our sincere thanks to the **Almighty God** that allow us to be here and do this project with his grace.

Next, we would like to express our deepest gratitude to **Professor Vittorio Murino, Professor Cigdem Beyan** and **Dr. Andrea Avogaro**. They gave us insightful lectures and engaging laboratories, which have significantly enriched our learning experience, and they also provided us with necessary information and directions regarding our courses and projects.

Finally, we would like to thank **our parents and friends**, who were always interested, supportive, and cheering.

Contents

List of Abbreviations

AD	Anomaly Detection
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GANs	Generative Adversarial Networks
GMM	Gaussian Mixture Model
GNB	Gaussian Naive Bayes
HOG	Histogram of Oriented Gradients
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
ML	Machine Learning
MinMax	Minimum Maximum Scaling
MVTec AD	MVTec Anomaly Detection
PCA	Principal Component Analysis
RBF	Radial Basis Function
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
VGG16	Visual Geometry Group 16-layer network

1 Introduction

1.1 Motivation and Rationale

An anomaly, also referred to as an outlier, is an observation that deviates significantly from the rest in the distribution of provided data. In machine learning and statistical modelling, anomalies are critical observations that could change the interpretation and modelling of complex systems, which are used to predict classification scores or make statistical inferences about population parameters. The detection and appropriate management of such anomalies is essential for greater accuracy and reliability in predictive models and statistical investigates.

Humans can often, quickly, detect an anomaly and an irregular pattern in datasets due to years of evolutionary training, which helps distinguish the signal from the noise. However, there is still a long way for intelligent systems to work effectively on different anomalies and unknown patterns within the carried information. This presents the challenge for anomaly detection in detecting such outlier observations and reasoning the properties that make them anomalies. Now, anomaly detection is a pressing issue in the modern world since different industries and business organizations are developing systems to be robust against anomalies in incoming data. Such systems will not only have to detect anomalies but also classify them as outliers and issue an alert to machine-learning experts for proper care of [1].

The project plans to implement anomaly detection based on MVTec AD dataset. The dataset contains high-quality images mostly used for anomaly detection in research areas. Anomalies are detected in many industrial applications that can prevent large financial loss, ensures quality of product and safety.

The increasingly dominant advanced ML-based methods are gradually replacing traditional anomaly detection, which relies heavily on rule-based approaches. The MVTec AD dataset broadly covers several highly detailed and diverse anomaly images across a range of categories, making it potentially an excellent benchmark to test and validate such models. The increasing need for automated, efficient, and accurate anomaly detection systems prompts the project that follows.

Thus, an exploration of unsupervised and supervised ML algorithms will reveals on strengths and limitations, which is expected to provide valuable insights through the project. This helps establish the most appropriate techniques to be applied for different scenarios of anomaly detection, which is very much crucial in adopting solutions for practical application. In addition, this project will further advance the state of the art by evaluating and refining algorithms meant for such purposes. Moreover, the insights gained from the project could be the motivation needed to inform future developments toward more robust and more adaptable models employed across a broader range of applications.

After which, it will give some valuable recommendations to the researcher and practitioners in general through a systematic performance evaluation of different algorithms on the MVTec dataset. These recommendations will guide future efforts at research and practical implementations on how best anomaly detection methods should be used effectively and efficiently.

2 State of the Art

Research into the different algorithms and methodologies for anomaly detection has formed an essential part of the domain of ML. Traditional techniques include statistical methods and rule-based systems, which are foundational but not very flexible and adaptable to modern applications. Therefore, ML-based approaches have emerged as powerful alternatives capable of handling high-dimensional data and learning the complicated patterns directly from the data [2].

Unsupervised learning algorithms that are promising include K-means, DBSCAN, and Gaussian Mixture Models, amongst others. These methods do not require label data from the anomalies. K-means clustering forms clusters out of data so that similar points end up in the same group. In that respect, anomalies will be found far off from the centroid of any group. DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise; it is efficient at identifying different shapes or sizes of clusters, but ironically, it has an ability for identifying those abandoned noise points as anomalies. GMM is a generative model that considers data as a mixture of some Gaussian distributions. It provides a probabilistic framework for anomaly detection by checking the likelihood of data points [2].

Supervised-learning algorithms, for example, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Gaussian Naïve Bayes (GNB), utilize labeled data to train classification models for new observations into normal or anomalous ones. SVM builds a hyperplane that gives the maximum margin between normal and anomalous data points, which is effective in high-dimensional spaces. KNNs detect anomalies as some distance to the nearest neighbors; hence, the number of neighbors an outlier has is less or those neighbors are distant. GNB is a probabilistic classifier based on Bayes' theorem assuming feature independence; it calculates the posterior probability as another step in classifying anomalies [3].

The MVTec AD dataset is a generally accepted benchmark for industrial anomaly detection that contains a range of object categories with annotated anomalies. Its use has demonstrated the efficiency of unsupervised as well as supervised approaches. An example is the elaborated work of Bergmann et al. [4], where an assessment was done on unsupervised methods using the MVTec AD dataset, showing that techniques such as autoencoders and one-class SVMs are well-equipped to detect subtle anomalies. Furthermore, supervised techniques have also been effectively used with deep learning models, like convolutional neural networks (CNNs), which have shown their superiority in tasks of image-based anomaly detection [5].

This project will build on these foundational studies by systematically comparing the performance of unsupervised and supervised algorithms on the MVTec AD dataset. The results will highlight an improved understanding of trade-offs between approaches and will make practical recommendations to be able to deploy effective anomaly detection in a variety of industrial settings.

3 Objectives

In an attempt to explore and design an effective anomaly detection model using machine learning algorithms, the general and specific objectives of the study are formulated as follows.

3.1 General Objective

The general objective of this project report is to develop and implement an effective anomaly detection model on MVTec dataset images using different machine learning algorithms. The developed model has a lot of importance to various industrial applications and advancement of machine learning techniques for anomaly detection.

3.2 Specific Objectives

The specific objectives of this project report work are:

- To evaluate the performance of unsupervised learning algorithms (K-means, DBSCAN, GMM) on the MVTec datasets.
- To assess and evaluate the effectiveness of supervised learning algorithms (SVM, KNN, GNB) for anomaly detection on the MVTec dataset images.
- To use hyperparameter tuning techniques to optimize the performance of both supervised and unsupervised models.
- To compare the performance of the supervised and unsupervised models and determine which approach is more suitable for anomaly detection in the given context.
- To provide insights and recommendations for future research and practical applications in the field of anomaly detection.

4 Methodology

The following methodology was followed to achieve the objective of the project:

4.1 Dataset Collection

The most important step for creating a robust and accurate anomaly detection model with different machine learning algorithms starts with obtaining suitable datasets. To train and build our model, we obtain images that illustrate the normal and abnormal scenarios. The MVTec AD dataset from Bergmann et al.[4] is utilized for training and testing our model. This is a very comprehensive benchmark dataset used to evaluate anomaly detection algorithms in industrial scenarios, including rich-content and high-resolution images related to samples for both objects and textures on normal and abnormal images.

For this study, five of the 15 different categories in the MVTec dataset were selected: bottles, pills, screws, grid and wood. These categories represent a diverse range of object types, textures, and possible defects one might find in industrial inspection environments. This dataset contains a distinct set of normal images for training and a labeled set of normal and abnormal images for testing.

4.2 Data Preprocessing

To balance the computational efficiency while trying to keep as much detail as possible, the images were resized to 128x128 pixels; subsequently, grayscale conversion was applied to maintain texture and shape features as they are, without color information. Then, we applied shuffling on the dataset in such a manner that it would result in a random sequence of data points and avoid the possibility that bias may occur while training and testing the models.

Features extracted using the Histogram of Oriented Gradients (HOG) feature extraction method that captured edge and gradient structures characteristic of local shape information with computation steps:

1. Calculate the magnitude and direction of the gradients at each pixel of the input image.
2. Divide the image into cells of the same size.
3. Group the gradient directions of all pixels in each cell into a specified number of orientation bins.
4. Group the cells into blocks of same size.
5. Normalize the cell histogram according to the other cells in the block to enhance differences in illumination and noise.

6. Acquire HOG features of all the blocks into one feature vector.

In addition to the HOG feature extraction method, we applied VGG16, which is a pre-trained convolutional neural network (CNN) model mainly used to extract features from images deeply. This VGG16 model is one of the most powerful used feature extractors for various computer vision tasks, including anomaly detection. Also, it was trained on the ImageNet dataset and it learned rich representations of visual patterns.

The extracted features were normalized according to the scaled Min-Max approach to ensure a predetermined range of values, mainly in the closed interval [0,1], and avoid dominance from features with much larger scales.

4.3 Dimensionality Reduction

We applied Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to reduce the dimensionality of our feature space during development of the model.

Principal Component Analysis (PCA): transforms the dataset to a new coordinate system so that the most significant variance projected on the coordinate, lies in the first coordinate. This new coordinate is the first principal component, with the second most significant variance in the second coordinate, and so on.

Linear Discriminant Analysis (LDA): maximizes class separability by projecting the data onto a lower-dimensional space. Its main objective is to maximize the ratio of the between-class variance to the within-class variance in any particular dataset for ensuring the maximum separability.

After feature dimensionality reduction, we transformed our dataset into various configurations to explore their impact on the model's performance. Specifically, we created variations consisting of:

- Original features without dimensionality reduction (no PCA/LDA)
- Features reduced using PCA
- Features reduced using LDA

4.4 Model Design

After performing the data preprocessing and dimensionality reduction, we moved towards the task of designing and developing an anomaly detection model using both supervised and unsupervised learning algorithms.

Model Designing using Supervised Learning Approach

Supervised learning algorithms have been chosen because they are most helpful in leveraging the labeled data to tell the difference between normal and anomaly images. This project performed experiments with three kinds of algorithms in the classification process such as Support Vector Machine, K-Nearest Neighbors, and Gaussian Naive Bayes.

- **Support Vector Machine (SVM):** known for its effectiveness in high-dimensional spaces and its ability to handle cases where the number of dimensions exceeds the number of samples.
- **K-Nearest Neighbors (KNN):** simple and non-parametric classification algorithm that assigns a class to a sample based on the majority class of its k-nearest neighbors.
- **Gaussian Naive Bayes (GNB):** probabilistic classifier based on Bayes' theorem with the assumption that the features are conditionally independent given the class label.

Model Designing Using Unsupervised Learning Approach

For the unsupervised learning approach, we used three different clustering algorithms such as k-means clustering, DBSCAN, and Gaussian Mixture Models (GMM).

- **K-means Clustering:** a centroid-based clustering algorithm that is used to partition the data into k clusters and each cluster is represented by the mean of the data points in that cluster. The working principle allows for every data point to be iteratively assigned to the nearest cluster center and then updates the centers so that they become the mean of assigned points. Convergence is assured by continuous repetition of this process.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** is a density-based clustering algorithm that groups data points that are closely packed together, marking points in low-density regions as outliers and also works by identifying core points (with a minimum number of neighbors within a specified radius), forming clusters from core points and their neighbors, and identifying points that do not belong to any cluster as outliers.
- **Gaussian Mixture Models (GMM):** is a probabilistic model that assumes that the data is generated from a mixture of several Gaussian distributions with unknown parameters.

4.5 Tools

We used a Python programming language for developing the prototype and achieving the project objectives. The Python was chosen because of its extensive libraries, ease of use, and strong community support. We also used Jupyter Notebook as the development environment since it provides an interactive interface ideally suited to data analysis, iterative development, and visualization. Besides that, we utilized different types of Python libraries, including scikit-learn for machine learning algorithms, OpenCV for image processing, NumPy for numerical computing, Pandas for data manipulation, Matplotlib, and Seaborn for visualization.

4.6 Evaluation

The evaluation of the models was performed using several key metrics to ensure a comprehensive assessment of their performance:

- **Accuracy:** is the ratio of correctly predicted instances to the total instances.
- **Precision:** is the ratio of correctly predicted positive instances to the total predicted positives.
- **Recall:** is the ratio of correctly predicted positive instances to the total actual positives.
- **F1 Score:** is the harmonic mean of precision and recall.
- **Confusion Matrix:** is a matrix or table showing the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions.
- **Classification Report:** A summary of precision, recall, F1 score, and support (number of true instances for each label).

5 Experiments and Results

5.1 Experimental Setup

The anomaly detection models were evaluated on the MVTec dataset images, which contains 15 different object categories with both good and defect samples. But for this study, we selected five categories which are bottle, pills and screws from object types; and grid and woods from the textures type.

Various preprocessing methods were applied for preparing the data for the machine learning algorithms. Initially, all images were resized into a fixed resolution of 128x128 pixel size to ensure the consistency and reduction of computational complexity. Then, a feature extraction was applied using the Histogram of Oriented Gradients (HOG) technique. HOG is an extremely popular feature descriptor that captures distribution information about edge orientations and intensities within an image; thus, it gives a compact representation of the image's local structure. In addition to the HOG features, deep features were also extracted using the pre-trained VGG16 model for the unsupervised machine learning approaches to compare performance with both feature sets. Various dimensionality reduction techniques have been applied after feature extraction to reduce this high-dimensional feature space and mitigate the curse of dimensionality.

In this project work, for supervised learning algorithms, both Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were employed. While the PCA is an unsupervised technique, which involves finding directions of maximum variance in the data and then projecting them onto a lower-dimensional subspace, LDA is supervised method, maximizing the separability between classes. In unsupervised learning algorithms, only PCA was used for dimensionality reduction. Finally, the extracted features were normalized using different scaling techniques such as standard scaling, robust scaling and Minmax scaling; but we utilized the Minmax scaling technique which gives as good results in our project. Minmax scaling transforms the features onto a given range, usually [0, 1], ensuring that all features are escalated to the same scale for learning process and prevents features with larger magnitudes from dominating the others.

Each algorithm was tested in three different scenarios: without dimensionality reduction, with PCA on all supervised and unsupervised algorithms, followed by LDA on supervised algorithms only. The anomaly detection models were developed using both supervised and unsupervised machine learning algorithms. For supervised learning algorithms, we just splitted the dataset into 80% for training and 20% for testing to ensure a robust evaluation of the models.

The supervised techniques involved were SVM, KNN, and GNB, while K-means clustering, DBSCAN, and GMM were applied as unsupervised methods. For SVM, the regularization parameter C was tuned across the values 0.001, 0.01, 0.1, 1, 10, and 100, and different kernel types: linear, radial basis function (RBF), polynomial, and sigmoid were evaluated to identify the optimal configuration. The value of C is controls the tradeoff between achieving a good fit and minimizing misclassifications, and the kernel type defines the similarity function used in the SVM formulation. Another experiment was the KNN classifier, the number of neighbors (n_neighbors) was tuned, evaluating values ranging from 3 to 9 (3, 5, 7, 9) using cross-validation to select the optimal value. These n_neighbors parameters defines how many of the closest training examples will be considered in determining the class of a test instance.

The Gaussian Naive Bayes (GNB) classifier was applied by using its default parameters through scikit-learn for straightforward evaluation.

In an unsupervised learning approach, the K-means clustering algorithm with the number of clusters K=2 representing for normal samples and anomalies was applied. The DBSCAN algorithm was also applied with epsilon values equal to 0.1, 0.5, and 1, and min_samples values equal to 5, 10, and 20, to find dense regions as normal samples and mark low-density regions as anomalies. And finally, the third unsupervised learning approach the GMM was also applied with the number of components equals to 2, which correspondence to the normal and anomaly classes.

Supervised models were trained on a combined set of good and anomalous images; mapping for 0 were used to represent good images, while 1 was used for the anomalous images. The unsupervised learning algorithms were trained with only the good images since this is a realistic scenario because, in real-world applications, it has only good data during the training.

For each algorithm, the tuning of hyperparameters was done by grid search along with cross-validation to find that combination of parameters, which led to better model performances. Grid search is one such systematic method for optimizing hyperparameters, where it tries all exhaustive subsets of a provided parameter space for the best combination. Grid search allows all possible combinations between a defined set of parameters to evaluate and find the best values that most optimize models' performance. K-fold cross-validation is a technique designed to check how generalizable the model might be. In this approach, the dataset is divided into k subsets where the model would train on k-1 folds and be validated on the remaining fold. This process is repeated k times, with each fold being the validation set once. In this way, this gives a more robust averaged performance across the k iterations for estimating the model's performance and avoids the risk of overfitting to any subset in particular. In this project we utilized grid search with k=5 cross-validation to ensure that the selected hyperparameters contribute to a well-generalizing model with unseen data; thus, it would have improved predictive accuracy and reliability.

Evaluation metrics such as accuracy, precision, recall, F1-score, and the confusion matrix, gave insights into each approach's strengths and weaknesses. Accuracy is a metric that measures overall correctness, whereas precision and recall deals with correctly identifying anomalies. The F1-score which is the harmonic mean of precision and recall, was used for model comparison because it gives a balanced measure of model performance, particularly important in anomaly detection where the distribution between classes can be highly imbalanced. The confusion matrix spells out true positives, true negatives, false positives, and false negatives, giving the overall view of how the training model is performing. The formulas are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

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

where as the confusion matrix provides a detailed breakdown of the predictions versus the actual labels in a tabular format.

The anomalous detection system of machine learning algorithms proposed on the MVTec AD dataset shall be thoroughly evaluated by the above-mentioned experimental setup. This project is intended to show the efficacy of these algorithms for anomaly detection by using different preprocessing techniques and tuning hyperparameters against standard evaluation metrics that provide insights.

5.2 Results and Discussion

This section delves into the performance of supervised and unsupervised learning algorithms for anomaly detection in industrial image data. We evaluated five categories (Bottle, Pill, Screw, Grid, Wood) using three algorithms (SVM, KNN, GNB) for supervised learning and three clustering algorithms (K-Means, DBSCAN, GMM) for unsupervised learning. Dimensionality reduction techniques (PCA and LDA) were applied to supervised algorithms and PCA only for unsupervised algorithms in order to assess their impact.

The performance metrics of accuracy and F1-score were calculated for each category, algorithm, and dimensionality reduction combination. These metrics provide valuable insights into the algorithms' ability to correctly classify instances as normal or anomalous, while accounting for imbalances in the dataset. The following tables provide a detailed breakdown of the performance metrics (Accuracy, F1-Score) for each category, algorithm, and dimensionality reduction combination.

Supervised Learning Performance:

Category	Algorithm	Accuracy	Acc_PCA	Acc_LDA	F1_score	F1_PCA	F1_LDA
Bottle	SVM	0.90	0.88	0.90	0.89	0.88	0.89
	KNN	0.83	0.83	0.90	0.79	0.79	0.89
	GNB	0.93	0.88	0.90	0.93	0.88	0.89
Pill	SVM	0.80	0.83	0.71	0.80	0.81	0.70
	KNN	0.77	0.76	0.70	0.75	0.74	0.69
	GNB	0.77	0.70	0.69	0.77	0.65	0.68
Screw	SVM	0.98	0.95	0.82	0.98	0.95	0.81
	KNN	0.75	0.75	0.82	0.72	0.73	0.81
	GNB	0.81	0.76	0.82	0.82	0.67	0.81
Grid	SVM	0.99	0.90	0.84	0.97	0.88	0.77
	KNN	0.84	0.84	0.83	0.77	0.77	0.76
	GNB	0.50	0.84	0.84	0.57	0.77	0.77
Wood	SVM	1.0	1.0	0.90	1.0	1.0	0.89
	KNN	0.82	0.82	0.86	0.74	0.74	0.87
	GNB	0.79	0.88	0.86	0.81	0.85	0.86

TABLE 1: Supervised Learning Performance

Unsupervised Learning Performance with Vgg16 & HOG features:

Category	Algorithm	Acc(no PCA)		Acc_PCA		F1_with no PCA		F1_with_PCA	
		HOG	Vgg16	HOG	Vgg16	HOG	Vgg16	HOG	Vgg16
Bottle	K-Means	0.52	0.60	0.42	0.40	0.46	0.55	0.39	0.37
	DBSCAN	0.76	0.76	0.76	0.76	0.43	0.43	0.43	0.43
	GMM	0.46	0.46	0.24	0.58	0.44	0.41	0.21	0.55
Pill	K-Means	0.51	0.41	0.66	0.57	0.45	0.36	0.44	0.50
	DBSCAN	0.84	0.84	0.84	0.84	0.46	0.46	0.46	0.46
	GMM	0.84	0.41	0.16	0.34	0.49	0.36	0.13	0.32
Screw	K-Means	0.48	0.33	0.51	0.66	0.43	0.32	0.47	0.51
	DBSCAN	0.74	0.74	0.74	0.74	0.43	0.43	0.43	0.43
	GMM	0.49	0.64	0.49	0.54	0.45	0.50	0.45	0.48
Grid	K-Means	0.44	0.54	0.44	0.46	0.42	0.51	0.42	0.44
	DBSCAN	0.73	0.73	0.73	0.73	0.42	0.42	0.42	0.42
	GMM	0.42	0.50	0.60	0.45	0.41	0.49	0.56	0.44
Wood	K-Means	0.52	0.65	0.53	0.65	0.48	0.53	0.44	0.53
	DBSCAN	0.76	0.76	0.76	0.76	0.43	0.43	0.43	0.43
	GMM	0.34	0.27	0.76	0.24	0.34	0.26	0.43	0.19

TABLE 2: Unsupervised Learning Performance

Summary of unsupervised learning performance with Vgg16 and HOG:

Category	Best_Unsupervised_Algorithm (F1-score)	Feature extraction method
Bottle	K-Means no PCA (0.55)	Vgg16
Pill	GMM no PCA (0.49)	HOG
Screw	K-Means with PCA (0.51)	Vgg16
Grid	GMM no PCA (0.56)	HOG
Wood	K-Means no PCA (0.53)	Vgg16

TABLE 3: Best unsupervised algorithms performance for each category

As shown in Table 2 & 3, deep features extracted from the pre-trained VGG16 model were better than the traditional HOG features in some classes and algorithms of the unsupervised approach. As an illustration, a direct application without dimensional reduction of Principal Component Analysis is that K-Means with VGG16 features achieved the classes Bottle (F1-score = 0.55) and Wood (F1-score = 0.53).

However, there were no consistent performance gains in all categories with VGG16. Best F1 scores were achieved using HOG features with GMM for Pill and Grid categories, which again underlines the need for careful evaluation of the fit of feature extraction methods for each specific dataset or application.

Summary of the selected supervised and unsupervised algorithms performance:

Category	Best_Supervised_classifier (F1_score)	Best_Unsupervised_classifier(F1_score)
Bottle	GNB no PCA (0.93)	K-Means no PCA (0.55) with Vgg16
Pill	SVM with PCA (0.81)	GMM no PCA (0.49) with HOG
Screw	SVM no PCA (0.98)	K-Means with PCA (0.51) with Vgg16
Grid	SVM no PCA (0.97)	GMM no PCA (0.56) with HOG
Wood	SVM no PCA (1.0)	K-Means no PCA (0.53) with Vgg16

TABLE 4: Best selected algorithms performance for each category

Finally as shown in Table 4, the overall performance of the unsupervised learning algorithms was poorer compared to the counterpart of the supervised ones. However, these results suggest that deep learning techniques can be a crucial means to enhance anomaly detection capabilities in an unsupervised framework via transfer learning using pre-trained architectures like VGG16.

Of the various supervised algorithms used in this work, SVM has shown a better performance than KNN and GNB under most categories. The reason for its better performance lies in what an SVM can do with high-dimensional and nonlinear data, often with industrial image data.

Because of the kernel trick, SVM can produce an optimal decision boundary in higher-dimensional feature space, leading to better class separation and improved anomaly detection. Perhaps due to this fact, the exceptional F1-scores attained by SVM without dimensionality reduction were at high levels: 0.97 (with polynomial(poly) kernel type and C=1), 0.98 (with polynomial(poly) kernel type and C=0.1), and 1.0 (with linear kernel type and C=0.1) for Grid, Screw, and Wood, respectively. The choice of SVM kernel types and C values are different for each category and this indicated that the kernel (linear, polynomial) determines the nature of the decision boundary SVM can create, while C controls the regularization strength, influencing the trade-off between achieving a good fit and minimizing misclassifications. Beside that,using various dimensionality reduction techniques, the performance of different algorithms was greatly affected under different categories. In the case of the Pill category, PCA improved the F1-score of SVM, thus showing that this dimensionality reduction step was helpful for extracting relevant features. On the contrary, for the Grid and Wood categories, PCA lowered the F1-scores, which proves that there was information of importance for the task of anomaly detection within the original feature space. It was observed that the choice of technique either PCA or LDA for dimensionality reduction impacted the performance of supervised algorithms. LDA method did not show an absolute win over PCA or an original feature space in any case—a fact which may underline the need for carefully looking into the appropriateness of each dimensionality reduction technique to a specific problem.

In the unsupervised learning domain, K-Means and Gaussian Mixture Models (GMM) generally outperformed DBSCAN and across multiple categories in terms of F1-score when we applied the VGG16 and HOG features, as evident from Table 2 and Table 3. Among unsupervised methods, the K-Means algorithm with a Vgg16 feature extractor achieved the best F1-scores for the Bottle, screw, and Wood categories: 0.55 without PCA, 0.51 with PCA, and 0.53 without PCA, respectively. Without PCA, GMM combined with HOG features performed best on both the Pill and Grid categories and achieved an F1-score of 0.49 and 0.56, respectively. Although DBSCAN is quite well-known for automatically identifying clusters of any shape while efficiently managing noise and outliers, its performance in this study was surpassed by K-Means and GMM for a several categories. The results show that deep learning techniques, for instance, VGG16, may have potential superiority in the increment performance of unsupervised anomaly detection algorithms like K-means, as reflected in the higher F1-scores achieved using VGG16 features compared to HOG features for some particular classes.

It should better to note that the overall performance of unsupervised algorithms was entirely below that of supervised learning approaches. The reasons might be many, though key among them include no labeled data available, high dimensionality of image data, and presence of class imbalanced data.

No trends of performance improvement could be noted across all categories in applying PCA as a dimensionality reduction technique for unsupervised algorithms. In some cases, such as with GMM on the pill category and K-means on the bottle category, applying PCA actually lowered the F1-score compared to using the original feature space. These observations highlights that dimensionality reduction techniques should be very carefully chosen, and the most appropriate unsupervised algorithm for each category or dataset should be selected. The choice depends on the characteristics of the data and assumes that the algorithms will have to make a trade-off between performance and computational complexity.

Several strategies can improve the performance of unsupervised learning algorithms: First, the limitations of the individual algorithms can be mitigated by combining multiple unsupervised algorithms or ensemble techniques. Second, using deep learning architectures like auto-encoders or generative adversarial networks (GANs) capture complicated patterns and representations in high-dimensional image data, hence improving anomaly detection performance. Additionally, integrating supervised and unsupervised techniques within a hybrid framework could advance strength from these two paradigms to deliver better performance than either applied singly.

In summary, supervised learning algorithms, especially SVM, turned out to be the best performers in this study, the power of unsupervised techniques cannot easily be sidetracked. Provided care is dealt with challenges and limitations, and ensemble techniques are coupled with more advanced deep learning methodologies, unsupervised learning shall stand very strong in features for detecting anomalies in an image dataset, especially in cases where labeled data is scarce or hard to get.

6 Conclusions

In this project, we have explored traditional machine learning algorithms for anomaly detection on the MVTec AD dataset. The primary objective has been to build a model of anomaly detection using supervised machine learning algorithms and comparing the efficiency of supervised learning algorithms (SVM, KNN & GNB) with unsupervised learning algorithms (K-means, DBSCAN & GMM) under both the categories of texture and object in the problem domain of anomaly detection. Furthermore, we tested the impact of dimensionality reduction techniques, PCA and LDA, on these algorithms.

The results of the experiment demonstrate that, in general, with respect to the F1-score, the supervised learning group of algorithms consistently outperformed the unsupervised approaches, proving significant separation ability between normal and anomalous samples. Among the supervised algorithms themselves, SVM did exceptional well, achieving F1-scores as high as 0.97 (with polynomial(poly) kernel and C=1), 0.98 (with polynomial(poly) kernel type and C=0.1), and 1.0 (with linear kernel type and C=0.1) for Grid, Screw, and Wood categories, respectively, without dimensionality reduction. Dimensionality reduction techniques, however, had a varied impact across categories and algorithms. Although PCA improved some by using, for instance, the Pill category with SVM, it also decreased performance on other models like Grid and Wood categories with SVM. However, the choice of SVM kernel types and C values differ for each category, and it indicates that while the kernel (linear,polynomial,etc) determines the nature of the decision boundary one is building, C controls the strength of regularization, thus influencing where to strike a good balance in the tradeoff between getting a good fit versus merely not being misclassified.

In contrast, the performance exhibited by unsupervised learning algorithms is relatively poorer compared to their supervised version. The best F1-score that an unsupervised algorithm obtained was 0.56 for GMM with HOG feature extraction method and without PCA in the Grid category. Poor performances by unsupervised algorithms are attributed to several factors, including a lack of labeled data and high dimensionality, among others, plus in some cases, imbalanced datasets with quite a few anomalous instances.

In this project the best F1-score observed was 1.0 for the SVM Algorithm with linear kernel type and C=0.1. Further, at the level of unsupervised anomaly detection techniques, the best value obtained for the F1-score is only 0.56 by GMM. It clearly shows that much more work is needed in developing more advanced and robust unsupervised anomaly detection techniques.

Among the directions that future work can take are the integration of traditional ML algorithms with Deep Learning techniques, investigation of the usage of ensemble methods for the effective combination of strengths of many algorithms, and extension of the analysis to a more general range of texture and object categories from the MVTec AD dataset and other industrial image datasets.

In conclusion, this project showed that the potential use of classical machine learning algorithms, especially supervised techniques such as SVM, in detecting anomalies within the MVTec image data. Techniques of dimensionality reduction can be applied to improve any performance parameters of such algorithms. However, the relatively lower performance of the unsupervised algorithms shows that the necessity of further research and development in this domain. Such deficiencies shall be addressed, and advanced techniques pursued to make anomaly detection systems more robust and find their practical applications. Both supervised and unsupervised techniques have their strengths and limitations, and a combination of approaches, coupled with ongoing research efforts, is necessary to achieve optimal results in this challenging and important field.

Bibliography

[1] S. V. T. R. H. S. ,. L. Vincent Wilmet, ”A Comparison of Supervised and Unsupervised Deep Learning Methods for Anomaly Detection in Images”.

[2] A. B. V. K. Varun Chandola, ”Anomaly Detection: A Survey,” 2009.

[3] K. P. Murphy, ”Machine Learning: A Probabilistic Perspective.,” MIT Press,, 2012.

[4] M. F. D. S. a. C. S. P. Bergmann, ”MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,” IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR, pp. 9592-9600, 2019.

[5] L. R. e. al., ”A Unifying Review of Deep and Shallow Anomaly Detection,” IEEE, vol. 109, pp. 756-795, May 2021.

[6] G. M. W. a. H. Hirsh, ”Learning to predict extremely rare events,” 2000.