Surface Defect Detection and Classification from Images

Raza Ahmad
Department of Electrical Engineering
Syed Babar Ali School of Science
Engineering, LUMS
Lahore, Pakistan
22060004@lums.edu.pk

Ahmed Waheed
Department of Computer Science
Syed Babar Ali School of Science
Engineering, LUMS
Lahore, Pakistan
22030009@lums.edu.pk

Abstract

Surface defect detection is of paramount importance in a variety of industrial applications, including quality assurance and product inspection. In this study, we aim to compare a range of machine learning and deep learning algorithms for the detection and classification of surface defects on paper reels. The dataset consists of 1500 images, classified into three categories: defect-free paper, paper with a hole, and paper with a pen mark. 80% of the dataset is allocated for training and the remaining 20% for testing. The Histogram of Oriented Gradients (HOG) feature extractor is employed to convert images into feature vectors, which serve as input for our classifiers.

Various algorithms were explored in strive for determining the optimal hyper parameters. Using a grid search method, the hyper parameters of the classifiers were optimized. The best parameters for the Support Vector Machine (SVM) with an RBF kernel are found to be 'C': 5 and 'gamma': 0.1. The performance of several classifiers were compared, including SVM, Naive Bayes, Logistic Regression, and Random Forest, based on their accuracy in defect detection. The SVM classifier achieves an accuracy of 79%, while Naive Bayes, Logistic Regression, and Random Forest obtain accuracies of 53.48%, 48.52%, and 37.87%, respectively.

Additionally, we investigated the performance of pre-trained deep learning models such as VGG16, ResNet18, ResNet50, and GoogleNet. The accuracy achieved by these models are 95% for VGG16, 95.67% for ResNet18, 98.67% for ResNet50, and 98.33% for GoogleNet.

"Segment Anything" model, a powerful image segmentation technique, was also employed to improve the performance of our models further. Findings of this paper highlights the potential of machine learning techniques in surface defect detection and classification, while emphasizing the importance of selecting appropriate algorithms and features for specific applications.

1 Introduction

In recent years, there has been a growing demand for artificially intelligent solutions that can improve the robustness of machine vision systems. This technology involves using computer algorithms and image processing techniques to detect and classify defects on the surface of a material, and it has become increasingly important in various industries such as manufacturing, quality control, and automotive. By automating the defect detection process, machine vision can significantly improve production efficiency, reduce human error, and save time and resources. Furthermore, this technology can provide valuable data that can be used to optimize production processes and prevent future defects from occurring. Overall, machine vision has proven to be a highly effective tool for surface defect detection, and it is expected to continue playing a vital role in the production process.

Feature extraction is a critical process in applications that require computer vision to recognize objects in an image. However, extracting features robustly can be computationally intensive, making it challenging to achieve real-time performance using pure software. To address this challenge, feature extraction algorithms are often used, but they require significant computational resources. Despite this, feature extraction remains an essential component of computer vision systems and is crucial for accurately identifying and classifying objects in images (2).

The Support Vector Machine (SVM) (3) is a popular supervised learning technique employed for various tasks in Machine Learning, including Classification and Regression. It is primarily used to address Classification problems. The main objective of the SVM algorithm is to develop a decision boundary, known as a hyperplane, which effectively divides an n-dimensional space into distinct classes. This allows for the accurate categorization of future data points. The hyperplane is constructed by utilizing the most critical vectors or extreme points, called support vectors. Consequently, the SVM algorithm is often referred to as the Support Vector Machine.
K-nearest neighbors is a non-parametric, supervised learning classifier that employs proximity to make predictions or classifications about individual data points. Although it can be utilized for both regression and classification tasks, it is primarily employed as a classification algorithm. The KNN algorithm operates on the principle that similar points are often located in close proximity to each other, enabling it to accurately categorize data points. The Naïve Bayes algorithm is a popular supervised learning technique that leverages Bayes theorem to solve classification problems. It is frequently used in text classification scenarios that involve extensive training data sets with high-dimensional features.

1.1 Literature Review

The use of Machine Learning and deep learning techniques for surface defect detection has gained significant attention in recent years. One of the researcher surveyed various effort in the literature for surface defect detection using deep learning techniques. They classified the literature in three ways: defect detection context, learning techniques, and defect localization and classification method. They also identified the limitations of traditional image processing techniques in handling noise, variations in lighting conditions, and backgrounds with complex textures. They highlighted the potential of deep learning for defect detection in manufacturing applications(4).

Detecting surface defects using images is crucial in various industries, including paper manufacturing. Traditional image processing techniques have limitations in detecting surface defects under complex industrial conditions. To overcome these limitations, deep learning models have been explored, and this paper proposes a new method for automatically detecting paper surface defects using images. The proposed method utilizes a convolutional neural network (CNN) that accurately detects paper defects. The CNN architecture is designed to learn the features of defect images and classify them according to their severity. The experimental results demonstrate that the proposed method achieves high accuracy and robustness in detecting paper surface defects. This novel approach provides a promising solution to the challenges faced by traditional methods and contributes significantly to research on surface defect detection using images(5).

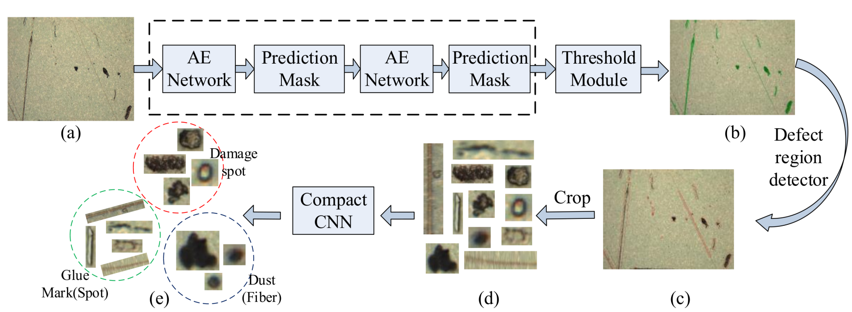

Figure 1: Metal Surface Defect Detection.

Similarly, many other applications such as rail defect detection is a crucial task in the maintenance and safety of railways. Traditional methods for rail defect detection using image processing techniques have limitations in detecting defects under specific conditions. One of a research proposes a novel method using convolutional neural networks (CNNs) trained on a database of photometric stereo images of metal surface defects. The proposed method uses differently colored light sources illuminating the rail surfaces from different directions to make cavities in the rail surface visible. The CNN architecture is designed to classify surface defects and non-defects accurately, and the impact of regularization methods such as unsupervised layer-wise pre-training and training data-set augmentation is explored. The experimental results demonstrate that the proposed method using classical CNNs outperforms the model-based approach and that regularization methods further improve the performance(6).

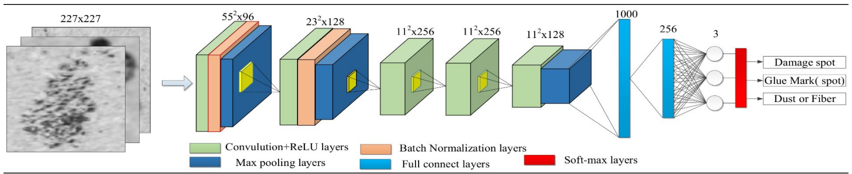

Figure 2: Surface Defect Detection using CNN.

One of a research uses automated surface defect detection. It is an important aspect of quality management in manufacturing processes. Traditional methods have limitations in dealing with imbalanced and severely rare images with defects. In this paper, a learning-based approach for automatic defect detection based on small image datasets is proposed. The approach employs Wasserstein generative adversarial nets (WGANs), feature-extraction-based transfer learning techniques, and a multi-model ensemble framework to successfully detect defects and reduce false negative rates (FNR). The experimental results demonstrate that the proposed approach achieves high accuracy in defect detection on decorative sheets and welding joints, with FNR results much lower than traditional vision methods used in production lines. This paper presents a significant contribution to the research on surface defect detection and offers a promising solution for the challenges faced by traditional methods(7).

2 Mathematical Formulation

2.1 SVM - Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning method that is widely employed for classification and regression tasks. Its main objective is to identify an optimal hyperplane that maximizes the margin between two classes while minimizing classification errors. This approach ensures accurate predictions and generalization to unseen data points, making SVM a robust and effective tool for various applications.

The primal optimization problem for SVM with a soft margin can be defined as:

$$min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \quad \xi_i \geq 0, \quad i = 1, \ldots, n$$

Here, $\xi \in \mathbb{R}^n$ represents a vector of non-negative slack variables, $C > 0$ is a regularization parameter controlling the trade-off between maximizing the margin and minimizing classification error, and the constraints ensure that all training examples are correctly classified with a margin of at least $1/\|\mathbf{w}\|$. The dual optimization problem corresponding to the primal problem can be written as:

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j s.t. \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

In this formulation, $\alpha \in \mathbb{R}^n$ denotes a vector of non-negative dual variables. The decision function can be represented in terms of the optimal dual variables as:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b\right)$$

The bias term b can be obtained from any support vector $\mathbf{x}_i$ that satisfies the condition $0 < \alpha_i < C$. The dual problem is a quadratic programming problem that can be efficiently solved using various optimization algorithms, such as the Sequential Minimal Optimization (SMO) algorithm.

2.2 Naive Bayes

Given a dataset of n observations with m features, where each observation is represented as a vector $x_i = (x_{i1}, x_{i2}, ..., x_{im})$ and a corresponding class label y_i, the goal of Naive Bayes algorithm is to predict the class label for a new observation $x_{new} = (x_{new1}, x_{new2}, ..., x_{newm})$.

The algorithm works by computing the posterior probability $P(y|x_{new})$ for each possible class label y, and then selecting the class label with the highest probability as the predicted label for x_{new}. The posterior probability is calculated using Bayes' rule:

where $P(x_{new}|y)$ is the likelihood of the new observation given the class label y, $P(y)$ is the prior probability of the class label y, and $P(x_{new})$ is the marginal probability of the new observation.

Naive Bayes algorithm assumes that the features x_{ij} are conditionally independent given the class label y, which means that:

where $P(x_{newj}|y)$ is the probability distribution of the j-th feature given the class label y.

To compute the probabilities $P(x_{newj}|y)$ and $P(y)$, Naive Bayes algorithm typically makes use of certain assumptions about the distributions of the features. For example, for binary features, it might use a Bernoulli distribution, while for continuous features, it might use a Gaussian distribution.

Once the probabilities $P(x_{new}|y)$ and $P(y)$ are estimated from the training data, they can be used to compute the posterior probability $P(y|x_{new})$ for each possible class label y, and the class label with the highest probability can be selected as the predicted label for x_{new}.

2.3 k-Nearest Neighbour (KNN)

Given a training dataset $T = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where x_i is a d-dimensional feature vector and y_i is the corresponding label, and a test sample x_q, the k-NN algorithm works as follows:

Compute the distances between x_q and all the samples in T: $\mathrm{d}_i = ||x_q - x_i||_2, \quad for \quad i = 1, 2, ..., n$

Select the k nearest neighbors to x_q from the training dataset T based on the distances computed in step 1.

Predict the label of x_q based on the majority label of the k nearest neighbors: $\hat{y_q} = \arg\max_{y_j} \sum_{i=1}^{k} [y_i = y_j]$

where $[y_i = y_j]$ is an indicator function that equals 1 if $y_i = y_j$ and 0 otherwise.

Note that the value of k is a hyperparameter that needs to be set before applying the k-NN algorithm.

3 Feature Extraction

Feature extraction is a vital step in machine learning, as it involves identifying and extracting essential features from raw data to be used as input for classifiers. Several algorithms can be employed for feature extraction, such as Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), Color Histogram, and Spatial Binning. In this study, we have chosen to use SIFT and HOG as our feature extraction methods. However, for the current iteration, we have focused on the HOG feature extractor.

3.1 Histogram of Oriented Gradient-HOG

The Histogram of Oriented Gradients, commonly abbreviated as HOG, is a feature descriptor that is widely used in computer vision applications, particularly for object detection. This technique involves the calculation of the occurrence of gradient orientation events in a particular area or region of interest within an image.

The following is the result of HOG feature extraction on our dataset.

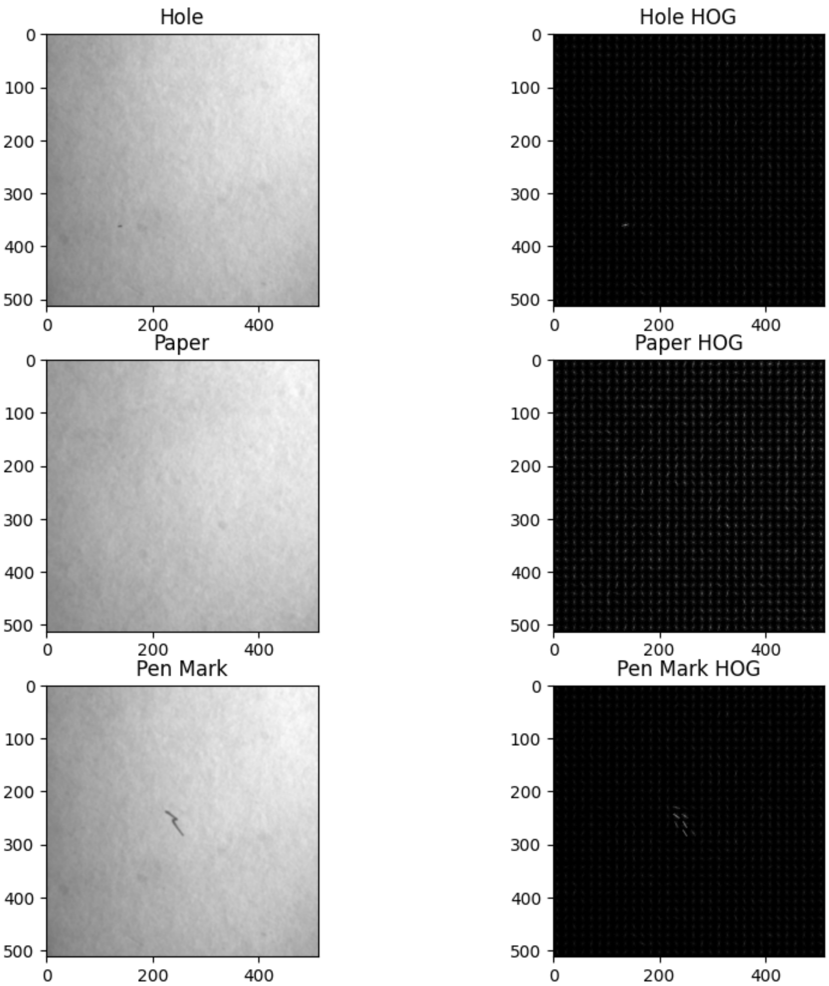

Figure 3: HOG Feature extraction.

4 Implementation

In the context of surface defect detection in images, there are several machine learning algorithms that can be used for image classification tasks, including Support Vector Machines (SVM) and Naive Bayes. Both algorithms can be trained on a dataset of labeled images, where each image has been manually annotated as either containing a defect or not. SVM is a popular machine learning algorithm used for classification tasks, which learns to identify patterns in the image features that correspond to the presence of defects. On the other hand, Naive Bayes is a probabilistic algorithm that assumes each feature in an image is independent of all other features and can also provide useful results in practice. Both algorithms involve extracting features from the images, such as color histograms, texture features, or edge detectors, which can then be used as inputs to the classifier. Overall, the implementation of both SVM and Naive Bayes on a dataset of images for surface defect detection involves training the algorithm on a labeled dataset and using it to predict the presence of defects in new, unseen images based on their extracted features.

4.1 Support Vector Machine-SVM

Support Vector Machines (SVMs) are a popular machine learning algorithm used for classification tasks. In the context of surface defect detection in images, an SVM can be trained on a dataset of labeled images, where each image has been manually annotated as either containing a defect or not. The SVM then learns to identify patterns in the image features that correspond to the presence of defects. When presented with a new, unlabeled image, the SVM can make a prediction about whether or not it contains a defect based on these learned patterns. To use SVM for image classification, it is common to extract features from the images, such as color histograms, texture features, or edge detectors, which can then be used as inputs to the SVM. Overall, the implementation of SVM on a dataset of images for surface defect detection involves training the SVM on a labeled dataset and using it to predict the presence of defects in new, unseen images.

We implemented the Support Vector Machine (SVM) classifier using various kernel functions and hyper parameters to evaluate its performance on the surface defect detection problem. The results for SVM with different Kernals are mentioned below:

- SVM (with RBF) -> Accuracy: 79%

- SVM (with "linear" kernel) -> Accuracy: 46.17%

- SVM (with linear with 'HoG') -> Accuracy: 43.52%

4.2 Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm that can also be used for image classification tasks. In the context of surface defect detection in images, a Naive Bayes classifier can be trained on a dataset of labeled images, where each image has been manually annotated as either containing a defect or not. Naive Bayes assumes that each feature in an image is independent of all other features, which is often not true in image datasets. However, despite this limitation, Naive Bayes can still provide useful results in practice. A Bayesian classifier is a type of statistical classifier that is capable of determining the probability of class membership for a given sample. This is achieved by leveraging Bayes' theorem, which allows for the estimation of the probability of a particular event occurring based on prior knowledge of related events. In contrast, naive Bayesian classifiers make the simplifying assumption that the influence of a given attribute value on a specific class is independent of the values of other attributes. (8) The algorithm learns the probability distribution of the features in the dataset and uses this information to predict the presence of defects in new, unseen images. To use Naive Bayes for image classification, it is common to extract features from the images, such as color histograms, texture features, or edge detectors, which can then be used as inputs to the classifier. Overall, the implementation of Naive Bayes on a dataset of images for surface defect detection involves training the classifier on a labeled dataset and using it to predict the presence of defects in new, unseen images based on their extracted features.

5 Results

In this study, we evaluated the performance of four classifiers for surface defect detection and classification: Support Vector Machine (SVM), Random Forest, k-Nearest Neighbors (k-NN), Naive Bayes and CNN. The classifiers were tested on a dataset containing 1500 images of paper reels with three different defect types: hole, paper (defect-free), and pen mark. Gaussian blur filter of 5x5 is applied to all the images for smoothing. We used the HOG feature extractor for feature identification and extraction.

First, we performed multiple experiments to find the best kernel for SVM.

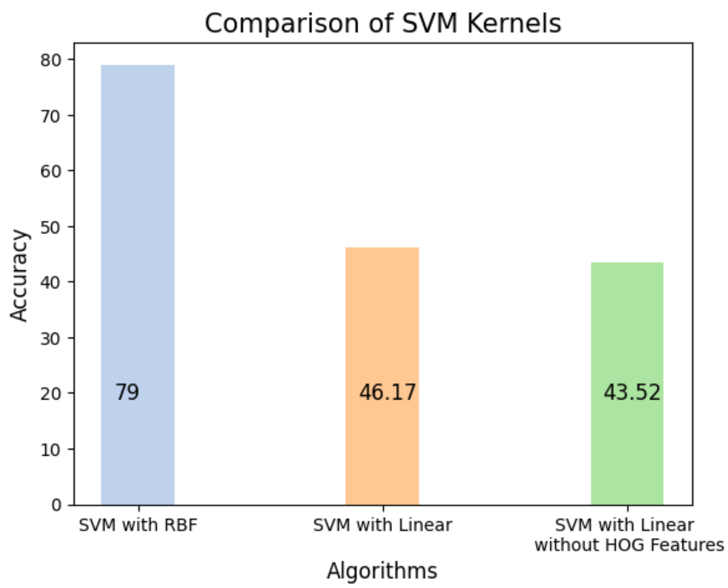

Figure 4: Comparison of SVM Kernels

SVM performed best with the RBF kernel. Then we performed further experiments to find out the best hyper parameters, like different values of C and gamma.

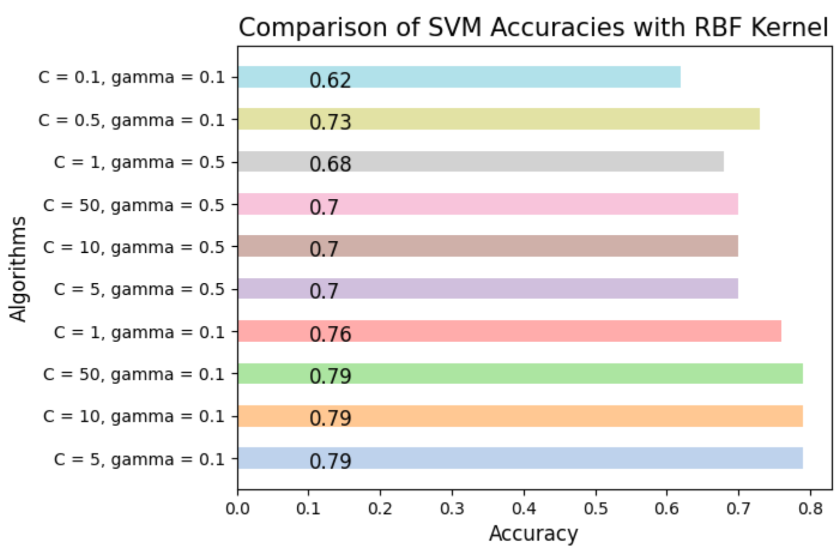

Figure 5: Comparison of SVM Accuracies with RBF Kernel

Then we compared the Best SVM with Naive Bayes, Logistic Regression and Random Forest.

Here are the F1, precion and recall scores. Class 0: hole, class 1: paper, class 2: penMark

Table 1: Model Comparison

Model	Class	Precision	Recall	F1-Score	Support
Decision Tree	0	0.32	0.23	0.27	98
	1	0.36	0.51	0.42	102
	2	0.38	0.32	0.34	101
Random Forest	0	0.33	0.37	0.35	98
	1	0.40	0.41	0.40	102
	2	0.36	0.31	0.33	101
Logistic Regression	0	0.41	0.37	0.39	98
	1	0.46	0.81	0.58	102
	2	0.88	0.28	0.42	101
SVM	0	0.81	0.47	0.59	98
	1	0.78	0.98	0.87	102
	2	0.80	0.91	0.85	101
Naive Bayes	0	0.38	0.43	0.40	98
	1	0.52	0.65	0.58	102
	2	0.84	0.52	0.65	101
Neural Network	0	0.33	0.03	0.06	98
	1	0.40	1.00	0.57	102
	2	0.83	0.29	0.43	101

Here we have used different convolutional neural networks to compare the accuracy. Figure 6 shows the comparison between vgg16, Resnet18, Resnet50 and googlenet.

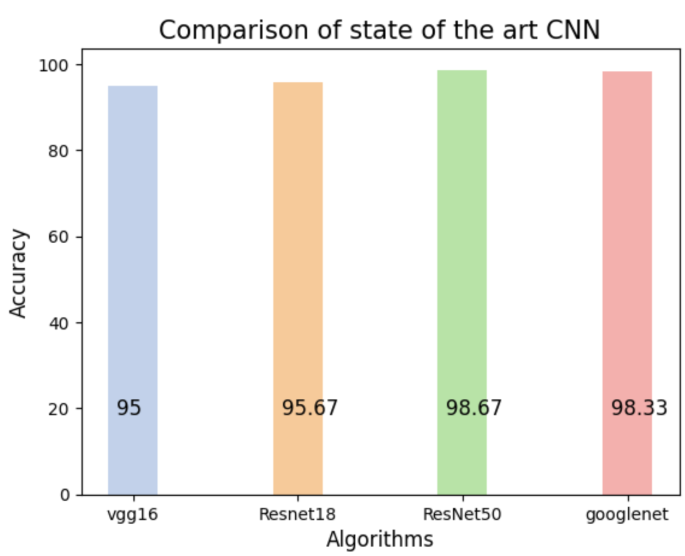

Figure 6: Comparison of different CNNs

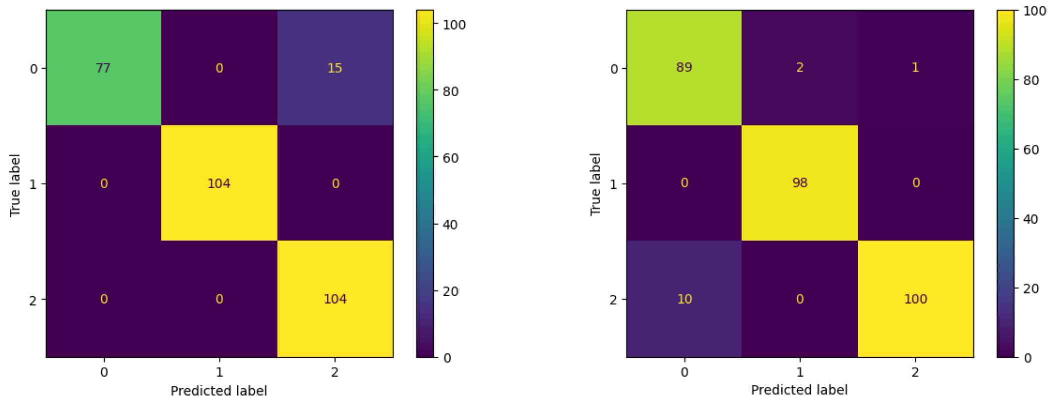

Figure 7: vgg16 & Resnet18 Confusion Matrix

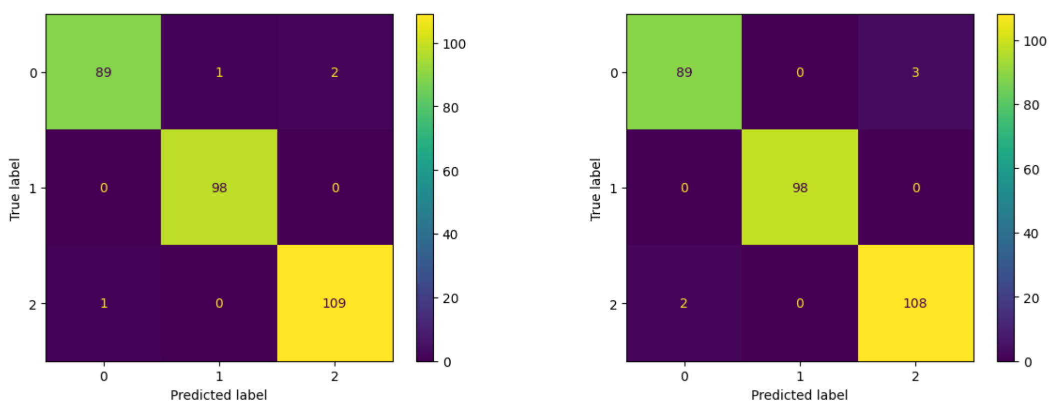

Figure 8: Resnet50 & googlenet Matrix

6 Defect Localization

6.1 Defect Localization using sliding window

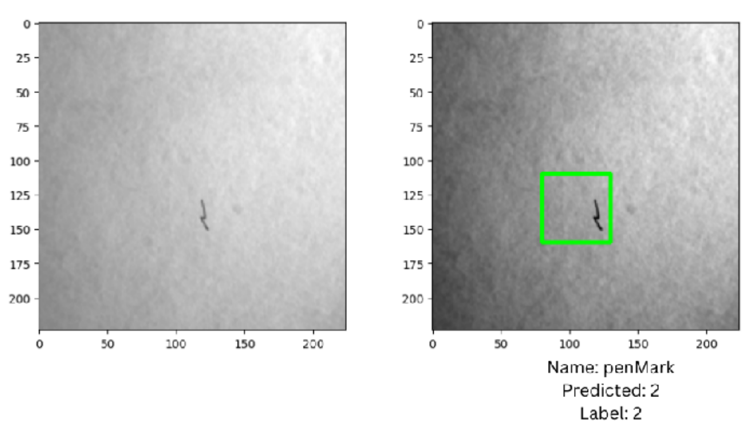

Figure 9: Defect Localization using sliding window

6.2 Defect Localization using SAM

The Segment Anything Model (SAM) is a groundbreaking development in the field of image segmentation, providing a more flexible and efficient solution compared to previous approaches.

Traditionally, there were two main approaches to image segmentation: interactive and automatic segmentation. Interactive segmentation involves a user iteratively refining a mask to guide the segmentation process. This approach allowed the segmentation of any object but required significant manual effort. In contrast, automatic segmentation focused on pre-defined object categories, requiring extensive amounts of training data and resources for each category.

SAM combines the best aspects of both approaches to create a single, versatile model capable of performing both interactive and automatic segmentation. SAM's prompt interface allows for flexible usage in various segmentation tasks by tailoring the prompt to the task at hand, such as user clicks, boxes, or text. This flexibility makes it applicable to a wide range of segmentation scenarios.

Lets look at a few examples

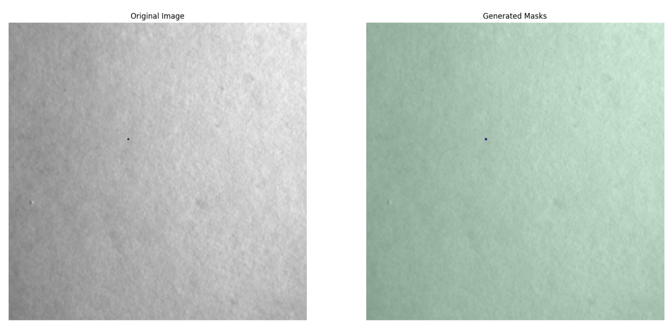

Figure 10: Hole detection using SAM

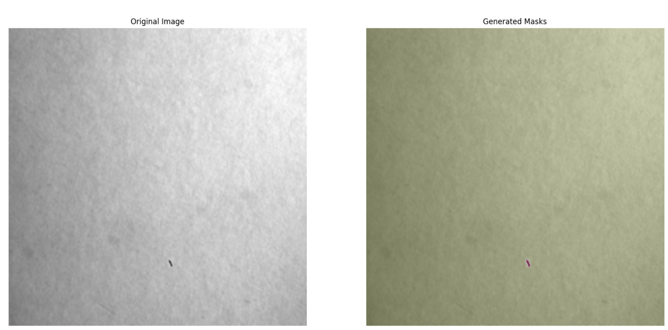

Figure 11: Example 2: Hole detection using SAM

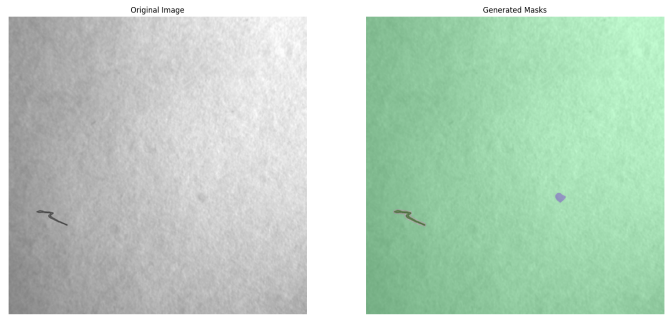

Figure 12: Example 3: penMark detection using SAM

7 Conclusions and Future work

In conclusion, our study demonstrates the efficacy of various machine learning algorithms in detecting and classifying surface defects on paper reels. The results indicate that the choice of algorithms and features plays a critical role in the performance of these models. Among the classifiers we tested, SVM with an RBF kernel achieved the highest accuracy of 79%, followed by Naive Bayes, Logistic Regression, and Random Forest. Furthermore, the pre-trained deep learning models, including VGG16, ResNet18, ResNet50, and GoogleNet, demonstrated impressive performance, with ResNet50 achieving an accuracy of 98.67%.

The use of the "Segment Anything" model, an advanced image segmentation technique, further underscores the potential of machine learning in surface defect detection and classification. However, there is still room for improvement, and our study serves as a foundation for further exploration in this area.

In future work, we plan to investigate the impact of data augmentation techniques on the performance of our models. Data augmentation can help increase the diversity and size of our dataset, potentially leading to more robust models that generalize better to new, unseen data. Moreover, we intend to explore other feature extraction techniques, in addition to HOG, to determine their impact on classification performance. Finally, we aim to test the efficacy of other state-of-the-art machine learning and deep learning algorithms, as well as investigate potential ensemble methods, to enhance the accuracy and robustness of our models further.

By refining our models and incorporating these planned enhancements, we hope to contribute to the development of more accurate and reliable surface defect detection systems, ultimately benefiting various industrial applications, such as quality assurance and product inspection.

References

[1] Scale-invariant feature transform. (2023, February 28). In Wikipedia. https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[2] Huang, Feng-Cheng, et al. "High-performance SIFT hardware accelerator for real-time image feature extraction." IEEE Transactions on Circuits and Systems for Video Technology 22.3 (2011): 340-351.

[3] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

[4] Bhatt, P. M., Malhan, R. K., Rajendran, P., Shah, B. C., Thakar, S., Yoon, Y. J., & Gupta, S. K. (2021). Image-based surface defect detection using deep learning: A review. Journal of Computing and Information Science in Engineering, 21(4).

[5] Tao, X., Zhang, D., Ma, W., Liu, X., & Xu, D. (2018). Automatic metallic surface defect detection and recognition with convolutional neural networks. Applied Sciences, 8(9), 1575.

[6] Soukup, D., & Huber-Mörk, R. (2014). Convolutional neural networks for steel surface defect detection from photometric stereo images. In Advances in Visual Computing: 10th International Symposium, ISVC 2014, Las Vegas, NV, USA, December 8-10, 2014, Proceedings, Part I 10 (pp. 668-677). Springer International Publishing.

[7] Le, X., Mei, J., Zhang, H., Zhou, B., & Xi, J. (2020). A learning-based approach for surface defect detection using small image datasets. Neurocomputing, 408, 112-120.

[8] Leung, K. M. (2007). Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering, 2007, 123-156.

Detailed experiments. `https://docs.google.com/document/d/e/2PACX-1vSshq5QNHUieAwhOf9EqBoojPLEpO_pMtHfCdedF9sgpS9K_YOcqw9d_cxjY9TjpV7Dp_PCB5rE7Y60/pub`