Project Name: Surface Defect Detection on Wafer Samples

Team members : *Jorge Bonilla and Oluwafemi Adelegan*

Brief Description/ Title

Title: Surface Defect Detection

The semiconductor industry, a crucial linchpin of the modern digital revolution, is no stranger to the challenges and rewards of innovation. Central to the manufacturing process of this industry is the production of high-quality silicon wafers, which form the base substrate for microchips. These microchips, serving as the building blocks of diverse electronic devices, owe their functionality and reliability to the integrity of the underlying silicon wafers. Given the progressively shrinking size of semiconductor devices, the task of identifying and rectifying defects on these wafers has become an intricate and demanding process. Yet, with the advent of artificial intelligence, and more specifically machine learning, there is a new wave of technological transformation reshaping this landscape.

Machine learning, a powerful branch of artificial intelligence, harnesses the potential of algorithms to glean patterns, learn from historical data, and make predictive decisions, often surpassing human intervention. In the realm of semiconductor manufacturing, these capabilities are reshaping the way we approach defect detection and classification on silicon wafers.

Machine learning applications are being deployed to automate defect identification on silicon wafers, providing a more reliable, precise, and cost-effective alternative to traditional methods. Through leveraging large volumes of data, machine learning algorithms can detect and classify wafer defects with increased speed and accuracy, minimizing the chances of faulty microchips making their way to market. Moreover, machine learning's predictive capabilities can be harnessed to preemptively identify potential issues in the wafer fabrication process. These predictive models help to increase yield rates, reduce waste, and enhance overall operational efficiency, thereby directly contributing to the industry's bottom line.

By augmenting defect detection with machine learning, the semiconductor industry can not only ensure higher quality standards in microchip production but also lay the groundwork for continued technological advancements in the digital era. This comprehensive exploration outlines how machine learning applications are pioneering new methodologies in silicon wafer defect identification, thus empowering the semiconductor industry to navigate its digital transformation journey with confidence.

Description: Our objective is to identify production items with defects using images captured in a controlled industrial environment. Various types of defects were observed on different samples and these defects include scratches, minor spots, and surface imperfections. In this project, by harnessing the potential of different machine learning algorithms, we hope to extract intricate patterns and subtle abnormalities from captured images, allowing us to uncover defects that may otherwise go unnoticed by human inspectors in a high-volume manufacturing environment.

Problem Statement

In manufacturing processes, most especially in high-volume manufacturing, there is a need to quickly and accurately identify production items with surface defects. Humans are prone to errors due to different factors and this makes the reduction in human interaction with yield in manufacturing highly imperative.

Objective

Identify production items with defects using between 8 and 10 different machine learning algorithms on images (both defective and non-defective) captured in a controlled industrial environment.

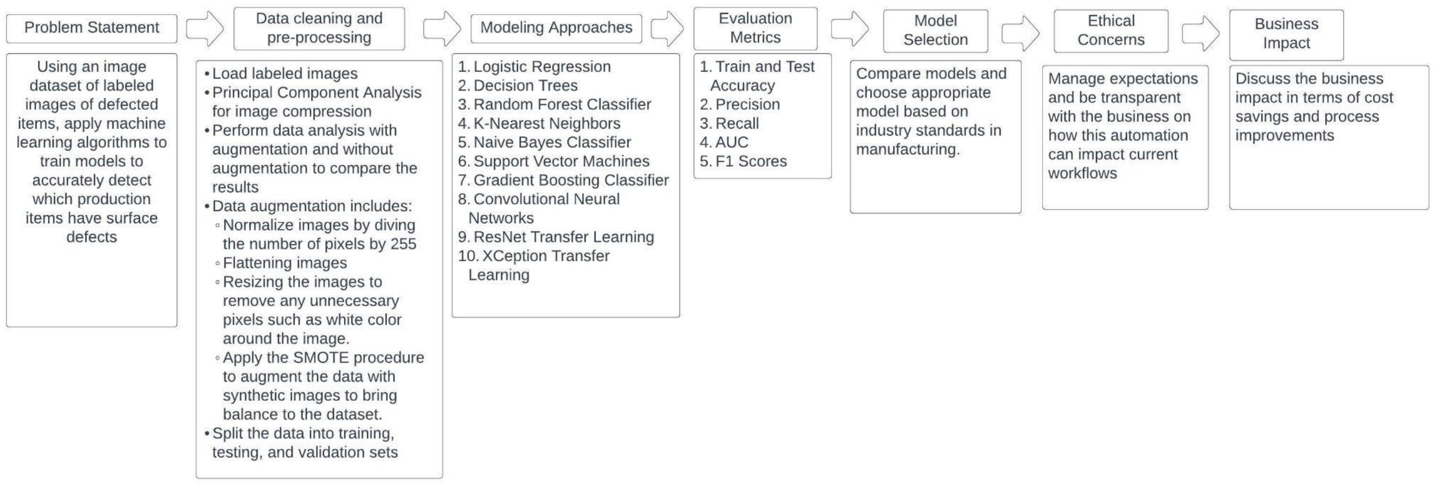
Approach/ Methodology

After performing the exploratory data analysis, image compression with Principal Component Analysis, and image augmentation, we divided the image data into train, test, and validation sets. The image augmentation consists of resizing the images to ensure consistent features, generalization, and computational efficiency. In order to have a balanced dataset, we applied Synthetic Minority Oversampling Technique (SMOTE) to the train set to the augmented data. As a result, we ran our machine learning algorithms on the non-augmented and augmented train sets.

Image classification algorithms implemented include:

Logistic Regression	Decision Trees	Random Forest	K-Nearest Neighbor	Naive Bayes Classifer
Support Vector Machines	Gradient Boosting Classifier	Convolutional Neural Networks	ResNet Transfer Learning	XCeption Transfer Learning

Block Diagram



Problem Statement

Using an image dataset of labeled images of defected items, apply machine learning algorithms to train models to accurately detect which production items have surface defects

Data cleaning and pre-processing

- Load labeled images
- Principal Component Analysis for image compression
- Perform data analysis with augmentation and without augmentation to compare the results
- Data augmentation includes:
  - Normalize images by diving the number of pixels by 255
  - Flattening images
  - Resizing the images to remove any unnecessary pixels such as white color around the image.
  - Apply the SMOTE procedure to augment the data with synthetic images to bring balance to the dataset.
- Split the data into training, testing, and validation sets

Modeling Approaches

1. Logistic Regression
2. Decision Trees
3. Random Forest Classifier
4. K-Nearest Neighbors
5. Naive Bayes Classifier
6. Support Vector Machines
7. Gradient Boosting Classifier
8. Convolutional Neural Networks
9. ResNet Transfer Learning
10. XCeption Transfer Learning

Evaluation Metrics

1. Train and Test Accuracy
2. Precision
3. Recall
4. AUC
5. F1 Scores

Model Selection

Compare models and choose appropriate model based on industry standards in manufacturing.

Ethical Concerns

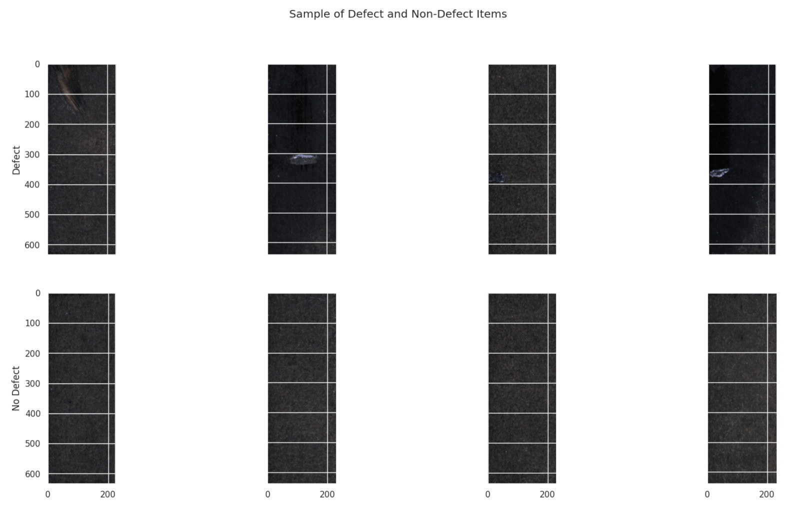
Manage expectations and be transparent with the business on how this automation can impact current workflows

Business Impact

Discuss the business impact in terms of cost savings and process improvements

Images

Samples of defected and non-defected wafers are as follows:

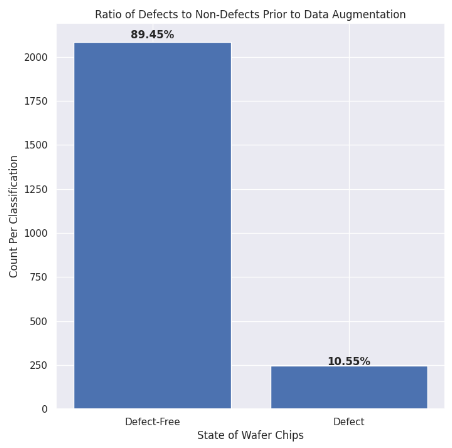


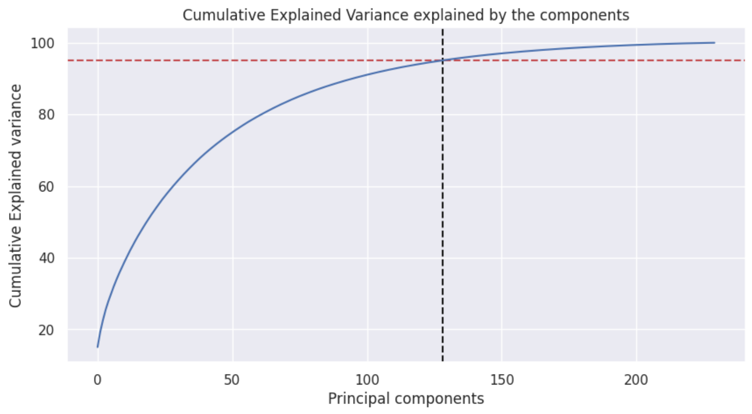
Datasets

Dataset source: https://www.vicos.si/resources/kolektorsdd2/

The dataset consists of:

- 246 images with visible defects
- 2085 images without any defect
- image sizes of approximately 230 x 630 pixels
- several different types of defects (scratches, minor spots, surface imperfections, etc.)





The graph above on the left shows the ratio of defects to non-defects in our dataset. To mitigate the bias due to the imbalance dataset, we performed SMOTE (Synthetic Minority Over-sampling Technique) on the train set to balance the ratio to 50% of defects and non-defects. We also implemented and analyzed the Principal Components in the image dataset to reduce the dimensions of the image. As shown in the graph above on the right, we can explain more than 95% of the variation in the images with at least 125 Principal Components. Therefore, we proceeded to compress the image resolution in the dataset using 125 Principal Components.

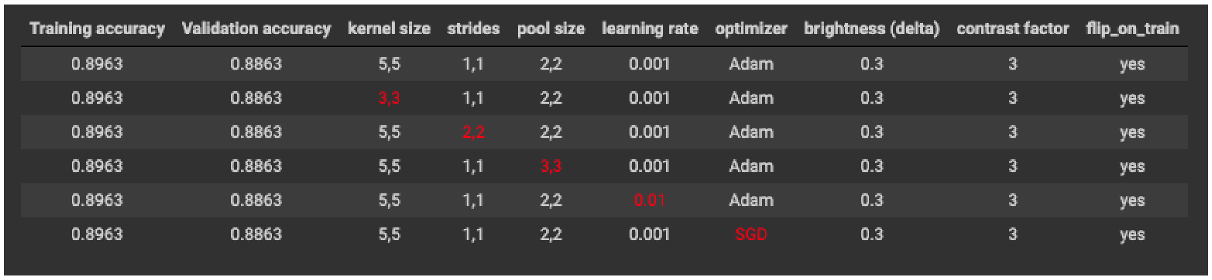
What is considered success/ failure? Evaluation Parameters (potential)

Based on previous research performed in this domain, we anticipate that an accuracy greater than 90% to be successful. Additionally, we aim to identify true positives where we correctly predict images with defects while minimizing both false positives and negatives. Therefore we will focus more on maximizing Precision and Accuracy.

Failure in this project is defined to be an accuracy that is less than 90%. The presence of false negatives further compounds failure, as it signifies instances where the model wrongly predicts an image as defect-free, despite the item's actual state being defective. This can lead to unforeseen loss of revenues and negative brand recognition.

Experiments

With the Convolutional Neural Network model, we tuned hyperparameters such as Learning Rate, Optimizer, Kernel Size, Strides, and Pool Size. The results from these experiments are presented in the table below:



Training accuracy	Validation accuracy	kernel size	strides	pool size	learning rate	optimizer	brightness (delta)	contrast factor	flip_on_train
0.8963	0.8863	5,5	1,1	2,2	0.001	Adam	0.3	3	yes
0.8963	0.8863	3,3	1,1	2,2	0.001	Adam	0.3	3	yes
0.8963	0.8863	5,5	2,2	2,2	0.001	Adam	0.3	3	yes
0.8963	0.8863	5,5	1,1	3,3	0.001	Adam	0.3	3	yes
0.8963	0.8863	5,5	1,1	2,2	0.01	Adam	0.3	3	yes
0.8963	0.8863	5,5	1,1	2,2	0.001	SGD	0.3	3	yes

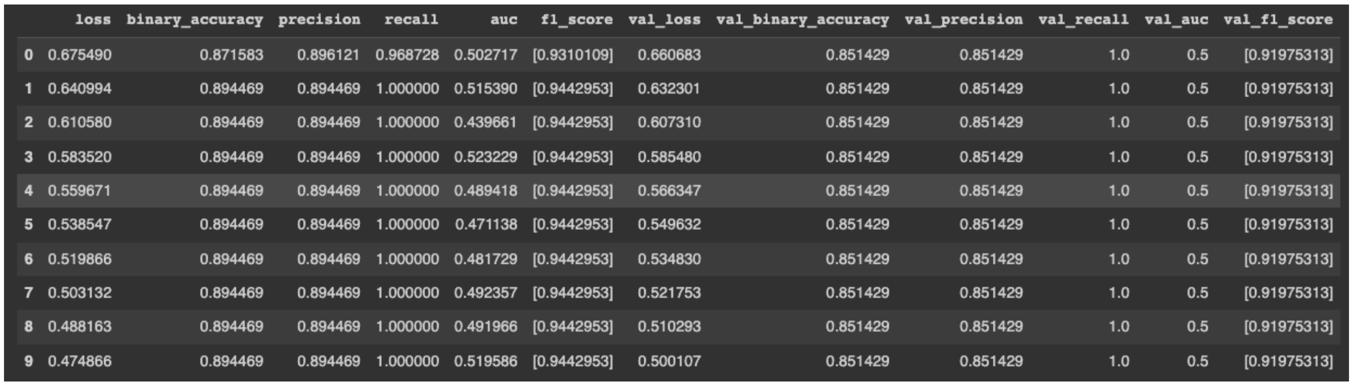
After performing hyperparameter tuning on the kernel size, strides, pool size, learning rate and Stochastic Gradient Descent, we observe no changes in the training and validation accuracy.

Results

As stated earlier, we are running different ML algorithms on the dataset to determine which of the algorithms perform better (based on our characterization of success/failure) compared to other algorithms.

1. Logistic Regression: Logistic regression is a statistical method for analyzing datasets where the outcome is binary. It's commonly used for classification tasks and this project is in that category. The outcome is binary to show if there are defects on the wafers or not. It is known that some of the advantages of logistic regression include its simplicity, interpretability, and efficiency. However, it struggles with non-linear boundaries and can be outperformed by more complex models

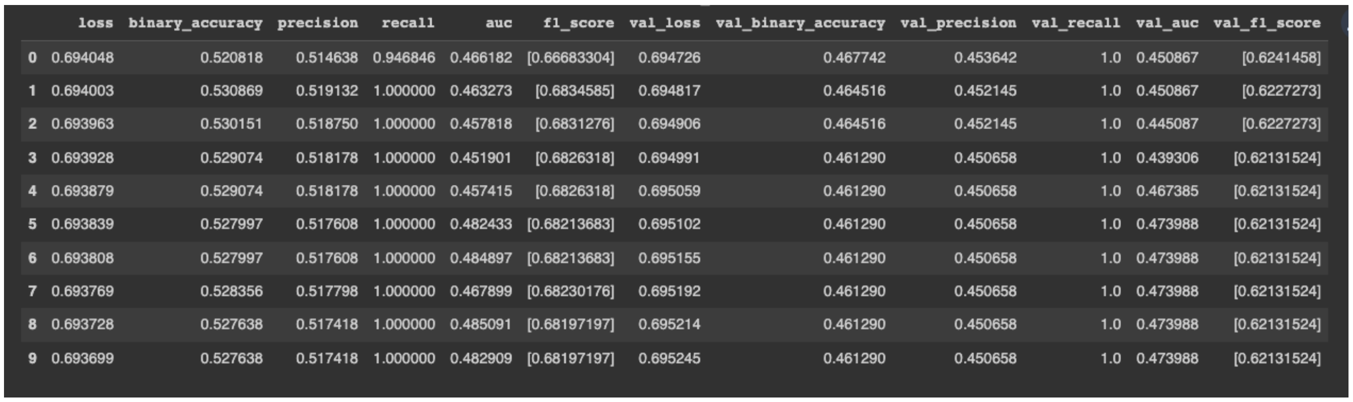
1.1 No Data Augmentation



	loss	binary_accuracy	precision	recall	auc	f1_score	val_loss	val_binary_accuracy	val_precision	val_recall	val_auc	val_f1_score
0	0.675490	0.871583	0.896121	0.968728	0.502717	[0.9310109]	0.660683	0.851429	0.851429	1.0	0.5	[0.91975313]
1	0.640994	0.894469	0.894469	1.000000	0.515390	[0.9442953]	0.632301	0.851429	0.851429	1.0	0.5	[0.91975313]
2	0.610580	0.894469	0.894469	1.000000	0.439661	[0.9442953]	0.607310	0.851429	0.851429	1.0	0.5	[0.91975313]
3	0.583520	0.894469	0.894469	1.000000	0.523229	[0.9442953]	0.585480	0.851429	0.851429	1.0	0.5	[0.91975313]
4	0.559671	0.894469	0.894469	1.000000	0.489418	[0.9442953]	0.566347	0.851429	0.851429	1.0	0.5	[0.91975313]
5	0.538547	0.894469	0.894469	1.000000	0.471138	[0.9442953]	0.549632	0.851429	0.851429	1.0	0.5	[0.91975313]
6	0.519866	0.894469	0.894469	1.000000	0.481729	[0.9442953]	0.534830	0.851429	0.851429	1.0	0.5	[0.91975313]
7	0.503132	0.894469	0.894469	1.000000	0.492357	[0.9442953]	0.521753	0.851429	0.851429	1.0	0.5	[0.91975313]
8	0.488163	0.894469	0.894469	1.000000	0.491966	[0.9442953]	0.510293	0.851429	0.851429	1.0	0.5	[0.91975313]
9	0.474866	0.894469	0.894469	1.000000	0.519586	[0.9442953]	0.500107	0.851429	0.851429	1.0	0.5	[0.91975313]

Train accuracy: 89.44%, Test data prediction accuracy: 9.3%.

1.2 Data Augmentation



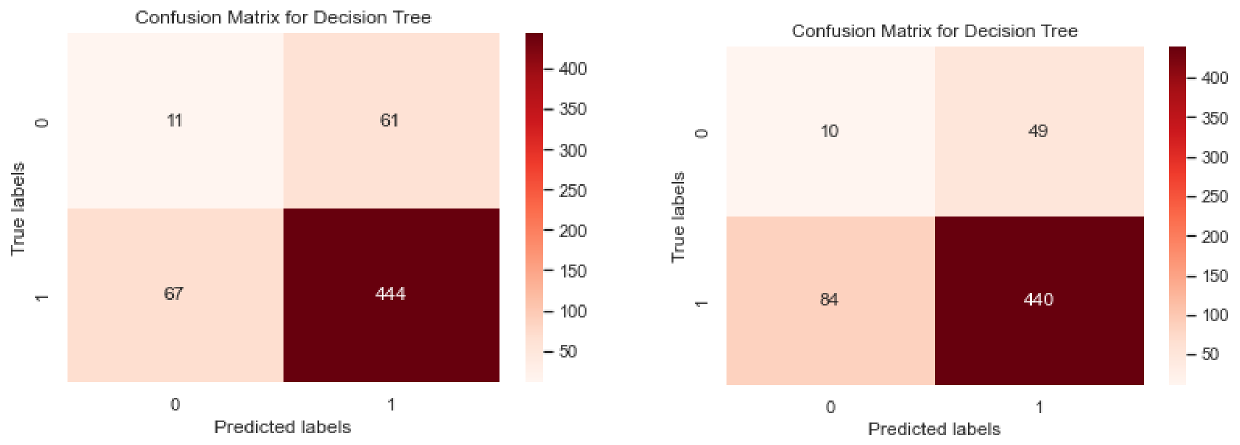
	loss	binary_accuracy	precision	recall	auc	f1_score	val_loss	val_binary_accuracy	val_precision	val_recall	val_auc	val_f1_score
0	0.694048	0.520818	0.514638	0.946846	0.466182	[0.66683304]	0.694726	0.467742	0.453642	1.0	0.450867	[0.6241458]
1	0.694003	0.530869	0.519132	1.000000	0.463273	[0.6834585]	0.694817	0.464516	0.452145	1.0	0.450867	[0.6227273]
2	0.693963	0.530151	0.518750	1.000000	0.457818	[0.6831276]	0.694906	0.464516	0.452145	1.0	0.445087	[0.6227273]
3	0.693928	0.529074	0.518178	1.000000	0.451901	[0.6826318]	0.694991	0.461290	0.450658	1.0	0.439306	[0.62131524]
4	0.693879	0.529074	0.518178	1.000000	0.457415	[0.6826318]	0.695059	0.461290	0.450658	1.0	0.467385	[0.62131524]
5	0.693839	0.527997	0.517608	1.000000	0.482433	[0.68213683]	0.695102	0.461290	0.450658	1.0	0.473988	[0.62131524]
6	0.693808	0.527997	0.517608	1.000000	0.484897	[0.68213683]	0.695155	0.461290	0.450658	1.0	0.473988	[0.62131524]
7	0.693769	0.528356	0.517798	1.000000	0.467899	[0.68230176]	0.695192	0.461290	0.450658	1.0	0.473988	[0.62131524]
8	0.693728	0.527638	0.517418	1.000000	0.485091	[0.68197197]	0.695214	0.461290	0.450658	1.0	0.473988	[0.62131524]
9	0.693699	0.527638	0.517418	1.000000	0.482909	[0.68197197]	0.695245	0.461290	0.450658	1.0	0.473988	[0.62131524]

Train accuracy: 51.74%, Test data prediction accuracy: 7.9%

2. Decision Trees: Decision trees are flowchart-like models that split data based on feature values. They're used for both regression and classification tasks, like credit risk or disease diagnosis. Advantages include interpretability, simplicity, and handling of mixed data types. However, they may not be the best choice for complex image classification tasks due to their tendency to overfit and their challenge in capturing intricate spatial patterns. They can also be outperformed by ensemble methods.

2.1 No Data Augmentation

With the data not augmented, the training accuracy is 98% and the test accuracy is 80% which shows that the algorithm is overfitting and that is also obvious in the confusion matrix below



Confusion matrix for the Decision Tree algorithm: (a) With non-augmented data; (b) with augmented data

2.2 Data Augmentation

With the augmented data, the training accuracy is 98% and the test accuracy is 78%. There is not much difference in the performance of the algorithm despite the data augmentation which further confirms the limitation of this algorithm

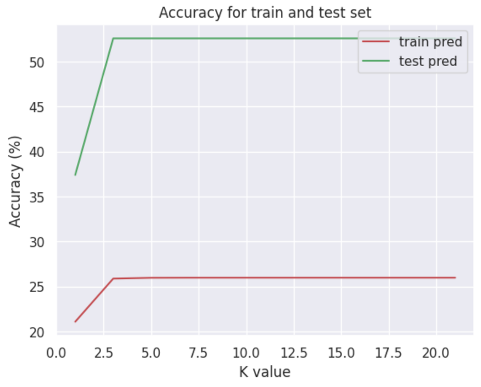
3. Random Forest Classifier: Random forests aggregate predictions from multiple decision trees to improve accuracy and prevent overfitting. They're used for tasks like feature importance and classification. Advantages include high accuracy, handling unbalanced datasets, and feature importance ranking. However, they can be slow and less interpretable than single decision trees.

3.1 Data Augmentation / Non Data Augmentation

With the raw data without augmentation, the training accuracy is 99% and the test accuracy is 90% and with the augmented dataset, the training accuracy is 99% and the test accuracy is 89%. This is expected from this algorithm because Random Forest is an ensemble approach, which combines predictions from multiple trees, are typically robust and less prone to overfitting, and tends to reduce variance without increasing bias too much. This characteristic generally leads to stable performance metrics across different datasets.

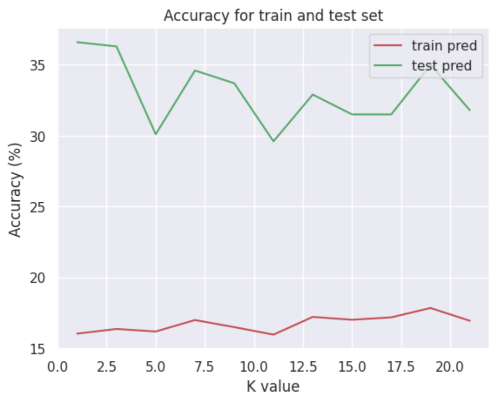
4. K-Nearest Neighbors: K-Nearest Neighbors (KNN) is a non-parametric method that classifies based on the majority vote of its k closest data points. It's widely used for classification and regression tasks like product recommendation. Advantages include simplicity and effectiveness with a well-chosen k. Disadvantages are its computational cost with large datasets and sensitivity to irrelevant features.

4.1 No Data Augmentation



The accuracy for the train and test on non-augmented data set converges to 25% and 51%, respectively, at a K value of approximately 3.

4.2 Data Augmentation



The accuracy for the train and test on augmented data set converges to 15% and 30%, respectively, at a K value of approximately 3.

5. Naive Bayes Classifier: a supervised learning algorithm, it applies Bayes' theorem with the "naive" assumption of conditional independence between a pair of features given the class variable. It is a fast algorithm to train and it works on binary and multi-class classifications. However, it assumes that all the features are independent which may not be the case in most datasets.

5.1 No Data Augmentation
Train accuracy: 80.66%,   Testing accuracy: 77.00%

5.2 Data Augmentation
Train accuracy: 89.50%,   Testing accuracy: 90.70%

6. Support Vector Machines: SVM is a powerful supervised learning algorithm primarily used for classification. The core idea behind SVM is to find the optimal hyperplane that best separates the data into classes in a multi-dimensional space. The optimal hyperplane is the one that has the maximum margin, i.e., the maximum distance between data points of both classes. Margin means the distance between the nearest data point (of any class) from the hyperplane. The "support vectors" are the critical data points that dictate the position and orientation of this hyperplane.

6.1 No Data Augmentation
Train accuracy: 92%,   Testing accuracy: 89%

6.2 Data Augmentation
Train accuracy: 94%,   Testing accuracy: 90%

7. Gradient Boosting Classifier: this algorithm builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage the class regression trees are fit on the negative gradient of the loss function, e.g. binary or multiclass log loss. Binary classification is a special case where only a single regression tree is induced.

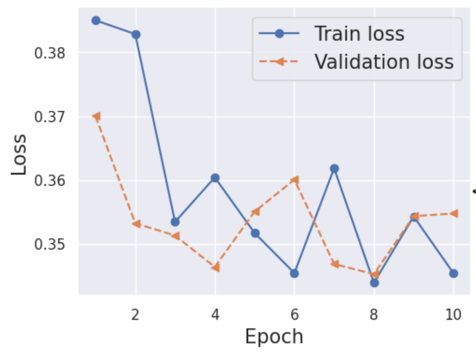
7.1 No Data Augmentation
Test accuracy 88.33%

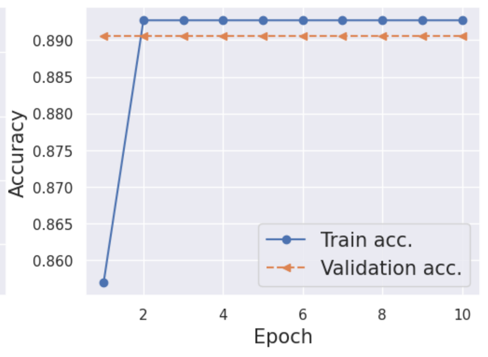
7.2 Data Augmentation

Test accuracy 90.39%

8. Convolutional Neural Networks: Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects in the image, and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms done in this project.

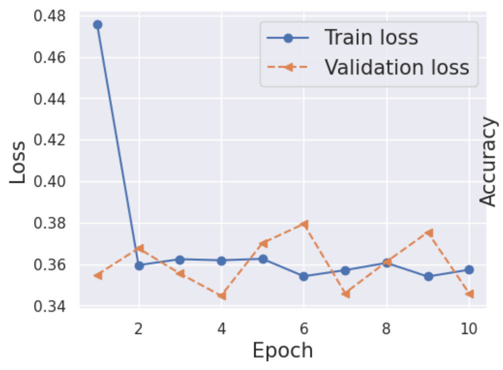
8.1 No Data Augmentation

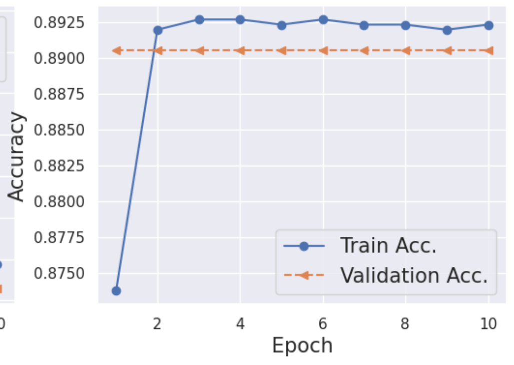




Train accuracy 89.27%, Test accuracy 90.36%

8.2 Data Augmentation

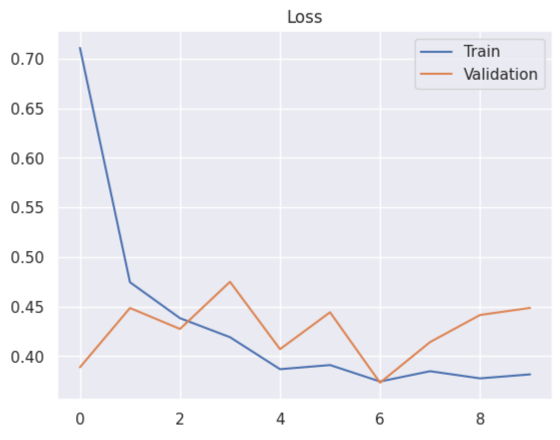


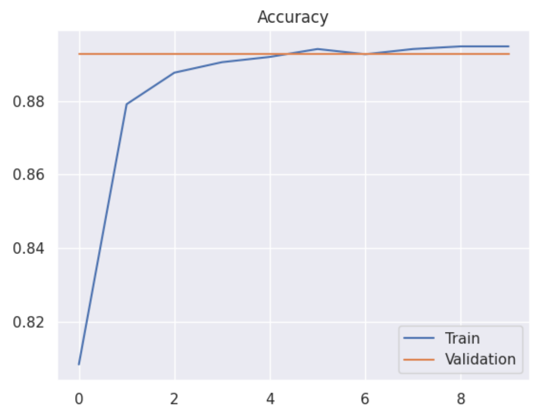


Train accuracy 89.23%, Test accuracy 90.36%

9. ResNet Transfer Learning: A transfer learning algorithm, ResNet50 is a convolutional neural network that is 50 layers deep, and is a part of the ResNet family, which are designed to handle image recognition tasks. The ‘50’ in ResNet50 denotes the depth of the network. The key innovation of ResNet is the introduction of “skip connections” or “shortcut connections”, which allow the gradient to be directly backpropagated to earlier layers.

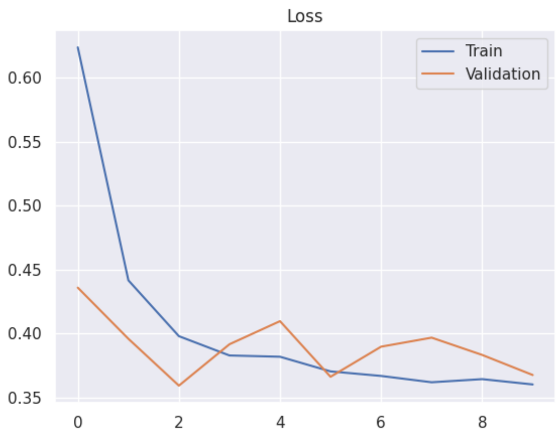
9.1 No Data Augmentation

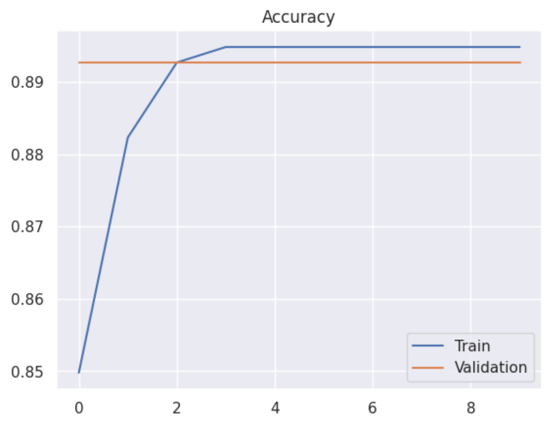




Train accuracy 89.27%, Test accuracy is 90.36%

9.2 Data Augmentation

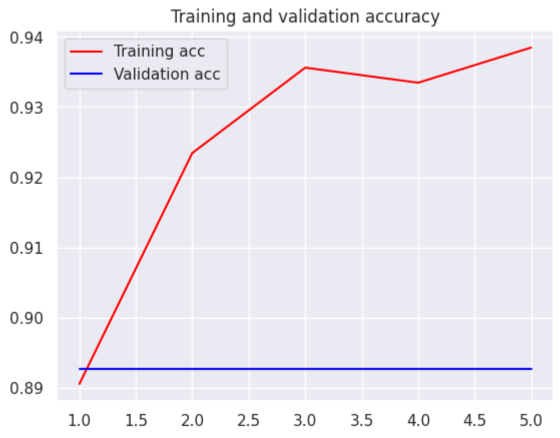


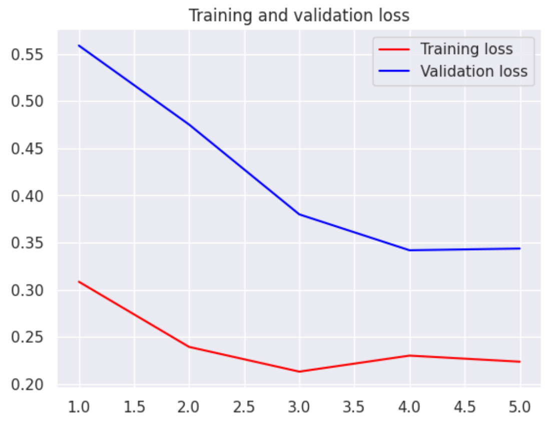


Train accuracy 89.27%, Test accuracy is 90.36%

10. XCeption Transfer Learning: another type of ensemble learning approach, Xception is an extension of the Inception architecture which replaces the standard Inception modules with depthwise separable convolutions.

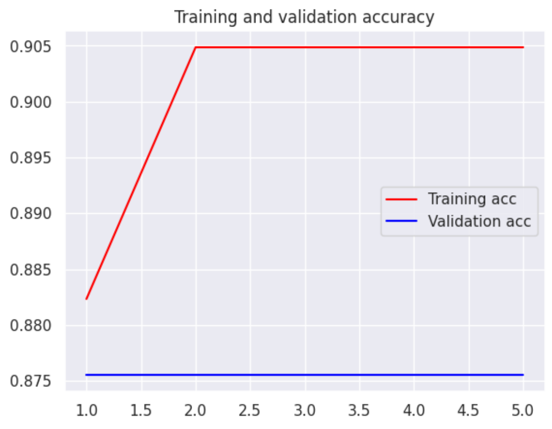
10.1 No Data Augmentation

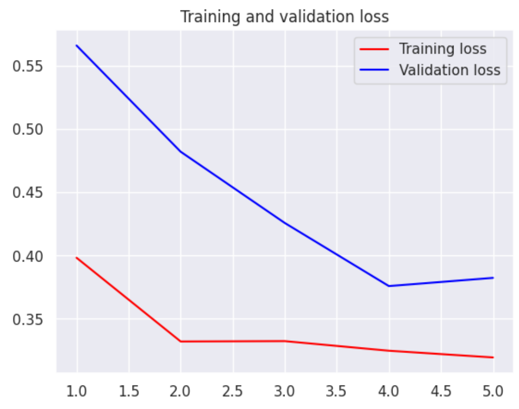




Train accuracy 92.63%, Test accuracy is 90.36%

10.2 Data Augmentation





Train accuracy 89.27%, Test accuracy is 90.36%

Tests/ Graphs/ Discussions

Logistic Regression: The model is built using Keras and it includes an input layers that takes the images and flattens them into vectors. The next layer is a dense layer that constructs the linear set of parameters for each input feature and bias, and applies sigmoid to the results. We use the sigmoid activation function since this is a binary classification task. Our optimizer includes Stochastic Gradient Descent with a learning rate of 0.001, and the loss function is Binary Crossentropy.

Decision Trees: For this algorithm, decisions are made by sequentially evaluating specific conditions on features, effectively partitioning the data in a hierarchical manner until it reaches a conclusion. For a classification tree such as this project, the decision (or the class label) of a leaf node is determined by the majority class of the samples falling into that node. Using this algorithm, we achieved test accuracy of 80% and 78% for both no-augmented and augmented datasets respectively but in both cases, the train accuracy was 98% which shows that the model did overfit.

Random Forest Classifier: This is an ensemble learning method that combines multiple decision trees to produce a more generalized and robust prediction. Each tree in the forest gives a "vote" for a class label, and the class label receiving the most votes is the Random Forest's final prediction for the input sample. With the training accuracy of 99% and test accuracy of 90% (99% and 89% respectively for the augmented data), the most common interpretation is that the model might be overfitting the training data but that might not be the best metric for evaluating performance, especially in imbalanced datasets such as this dataset. So, with precision of 0.92, recall of 0.99, F1 score of 0.95, the model performed pretty well

K-Nearest Neighbors: In this algorithm we rely on the distance between feature vectors and the labels associated with each image so that we can predict whether a wafer is defective or not. After experimenting with various values for K, we found that for augmented and non-augmented data the training and test accuracies converge at a value of 3. Given the low accuracy value using K-Nearest Neighbor, we found that this algorithm is not suitable for our image classification task since defects in the wafer are quite varied and it is difficult to find the most common class among k-closest examples.

Naive Bayes Classifier: with this approach we tested applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the class variable. Using the Sci-kit Learn implementation of Naive Bayes algorithm, we achieved a test accuracy of 77.53% on non-augmented data and 91.56% on augmented data.

Support Vector Machines: Using this algorithm, we achieved a test accuracy of 89% and 90% for both non-augmented and augmented datasets respectively. This is understandable for SVM because SVMs are known for their good generalization properties. Also, since this problem is not overly complex, it is easier for SVM to achieve similar accuracies on both training and test datasets

Gradient Boosting Classifier: we implement an ensemble method where the algorithm builds an additive model in a forward stage-wise fashion. This allows for the optimization of arbitrary differentiable loss functions. Using the Sci-kit Learn implementation of Gradient Boosting, we achieved a test accuracy of 87.70% on non-augmented data and 87.30% on augmented data.

Convolutional Neural Networks: we create a neural network model with 8 layers. The first layer is the convolutional 2D layer in the model with 2,432 parameters. Next, the model has a max pooling layer with pool size 2,2 and strides of 2 to reduce the spatial dimensions by hald. We add a second convolutional 2D layer with 51,264 parameters and another max pooling layer. A flatten layer is added along with a dense fully connected layer for greater generalization. The dropout layer with a rate of 0.5 is included in the model to help prevent overfitting. Lastly, the model has a fully connected layer that sets the activation function to "None" in order to output the logits. In total this model has 205,576,641 parameters.

ResNet Transfer Learning: we use the ResNet model to take advantage of the innovative neural network architecture in this model and transfer it to our image classification problem. This is a model that makes use of the residual module involving shortcut connections. With a total of 185 layers and 24,809,605 parameters, we also include the ImageNet weights when importing the model. This model performs relatively well since the test accuracy on both, non-augmented and augmented data, is 89.50%.

XCeption Transfer Learning: the Xception model is another type of convolutional neural network developed by François Chollet that implements 134 layers, 20,865,578 parameters, and takes images with an input size of 299 by 299. The main architecture improvement of this model is that it replaces the standard Inception Transfer Learning modules with depthwise separable convolutions. The test accuracy on non-augmented data is 89.51% and 88.22% on augmented data.

Constraints

We had an imbalanced dataset with 87% of the data composed of non-defect items. With more time we could have either searched for a more balanced dataset of wafers or contacted the distributors of the dataset for additional labeled defect images. Time was also another constraint as we would have liked to experiment with ensemble methods of different transfer learning models.

Standards

Enviornment:
Google Colab Pro+ (Paid version for GPU access)
Python 3.10.12

Packages and frameworks:

Numpy 1.23.5	Pandas 1.5.3	Matplotlib 3.7.1
Seaborn 0.12.2	Scikit-Learn 1.2.2	Tensorflow 2.13.0

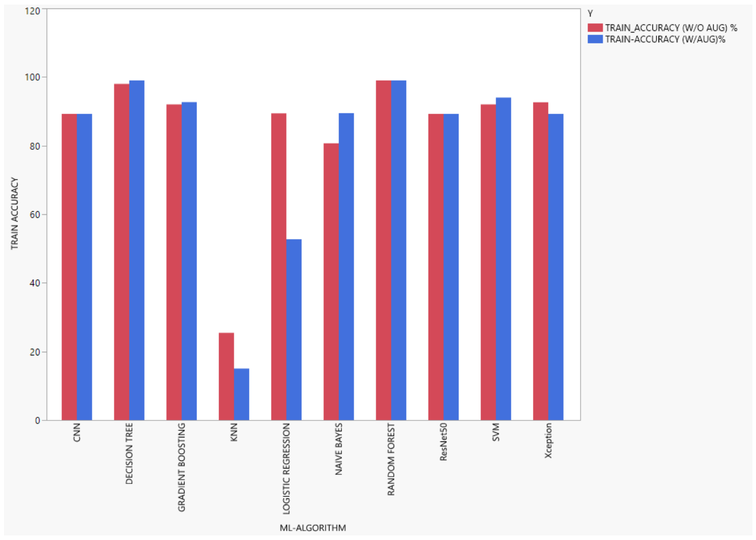
Comparison

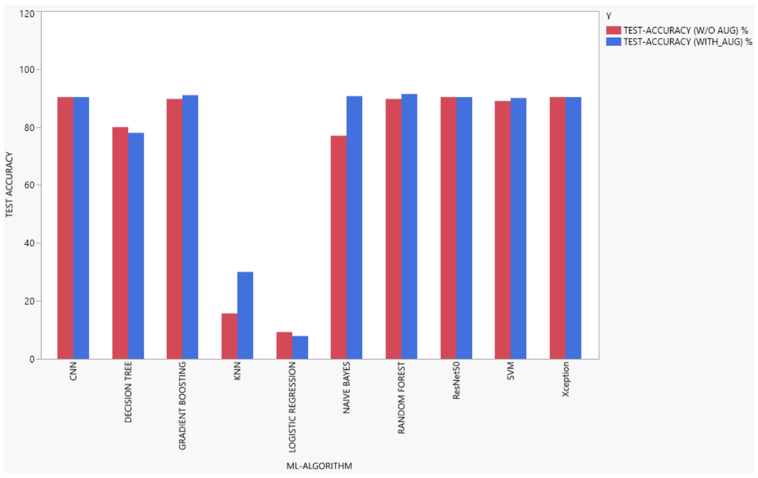
We implemented a baseline Logisitc regression model that performed well on augmented data with a test accuracy of 89.2%. After trying classical supervised machine learning algorithms, neural networks, and transfer learning approaches, we found that Convolutional Neural Networks and their variations found in the transfer learning architectures are the best approaches for our automatic wafer defect detection problem. This is because Convolutional Neural Networks have dense layers in their architecture that allows the model to generalize across images of defects and non-defects to learn the precise weights needed to make accurate predictions surpassing 90%.

ML Algorithm		Train Acc(%)	Test Acc(%)	Precision (%)	Recall	F1- Score
Logistic Regression	W/O Aug	89.44	9.3	88.33	1.0	0.93
	W/ Aug	52.72	7.9	89.19	1.0	0.94
K-Nearest Neighbors	W/O Aug	25.44	51.0	0.88	0.06	0.11
	W/ Aug	15	30	0.88	1.0	0.93

SVM	W/O Aug	92	89	90	1.0	0.95
	W/ Aug	94	90	90	1.0	0.94
Decision Tree	W/O Aug	98	80	90	0.89	0.89
	W/ Aug	99	78	90	0.84	0.86
Random Forest	W/O Aug	99	90	92	0.99	0.95
	W/ Aug	99	89	92	0.99	0.94
Naive Bayes Classification	W/O Aug	80.66	77	90.5	0.82	0.86
	W/ Aug	89.5	90.7	92.25	1.0	0.96
Gradient Boosting Ensemble	W/O Aug	92	**88.33**	89.86	0.98	0.94
	W/ Aug	92.7	**90.39**	90.34	1.0	0.95
CNN	W/O Aug	89.27	**90.36**	89.27	1.0	0.94
	W/ Aug	89.23	**90.36**	89.26	0.99	0.99
ResNet50	W/O Aug	89.27	**90.36**			
	W/ Aug	89.27	**90.36**			
Xception	W/O Aug	92.63	**90.36**			
	W/ Aug	89.27	**90.36**			

Table showing the performance comparisons between all the ML Algorithms used in this project





Bar Charts showing the performance of the algorithms both on augmented and non-augmented datasets

Limitations of the Study

There's no way to tell what type of defects we have. If we had access to a wafer dataset detailing the types of defects there are, we could work on a multi-class image classification that could provide better accuracy. Additionally, the imbalance dataset was another limitation of the study as we had to employ multiple techniques to balance the study which may cause unforeseen bias in the results. With a balanced dataset, we expect our accuracy metrics to improve.

Ethical Considerations

Incorporating machine learning applications into the sphere of semiconductor manufacturing, particularly for silicon wafer defect detection, offers a promising avenue for technological advancement. However, parallel to the technological discourse, it is essential to consider the ethical aspects surrounding these applications, particularly concerning data privacy, job displacement, algorithmic transparency, and the potential for bias and fairness.

Data privacy and security form a critical ethical consideration, as machine learning algorithms depend on extensive data for their training and development. In the semiconductor manufacturing realm, such data might include sensitive details about manufacturing processes, equipment specifics, and quality control parameters. It is thus crucial for businesses to implement robust security measures to safeguard this data and prevent potential breaches, ensuring a fair and secure operational environment.

The potential displacement of human roles due to automation forms another ethical aspect. As machine learning becomes increasingly adept at defect detection, it may threaten the jobs of human inspectors and technicians. Striving for fairness in this context means seeking a balance where machine learning supports and augments human work rather than eliminating it. It may also necessitate the creation of initiatives for retraining and upskilling workers, ensuring an equitable distribution of the benefits that machine learning brings.

Simultaneously, we must also consider issues of algorithmic bias and transparency. Bias can occur if the machine learning models are trained predominantly on data from specific types of manufacturing processes or equipment, potentially resulting in the misidentification of defects on wafers produced through different means. Ensuring fairness necessitates a diversity in the training data to prevent such biases, which could otherwise lead to considerable financial and operational inefficiencies.

Transparency and explainability of machine learning models also come into play, particularly in an industry where the stakes are high. Algorithmic decisions should be understandable and accountable, with stakeholders given the ability to discern how a certain decision was reached. Lack of transparency might lead to flawed outcomes, and thus, the pursuit of fairness dictates that manufacturers aim for models that offer clear insights into their decision-making process.

So, as the semiconductor manufacturing industry leverages machine learning for defect detection in silicon wafers, ethical considerations such as data security, potential job displacement, and the avoidance of algorithmic bias and ensuring transparency must be at the forefront. Only by acknowledging and addressing these ethical challenges can we ensure a responsible, fair, and effective deployment of machine learning in this critical sector.

Future Work

- Wafers with multiple classes of defects
- Higher resolution images to better identify defects
- Test trained model on other types of industrial settings such as steel manufacturing
- Ensemble methods of transfer learning models.

References

[1]Božič, J., Tabernik, D., & Skočaj, D. (2021). Mixed supervision for surface-defect detection: From weakly to fully supervised learning. Computers in Industry, 129, 103459.

[2]Tao, Xian, et al. "Deep learning for unsupervised anomaly localization in industrial images: A survey." IEEE Transactions on Instrumentation and Measurement (2022).