AI-DRIVEN WAFER DEFECT DETECTION AND CLASSIFICATION REPORT

A Project Submitted to the **Data Science Department** at **Moringa School** in partial fulfillment of the requirements for the course **DATA SCIENCE**

By the Group **CHIPSLEAUTHS**

	Name
Role	
Group Leader	Valerie Kigo
Group Member	Joel Gitonga
Group Member	Lawrence Kamerino
Group Member	Hudheyfa Mohamud

Technical Mentor: Faith Rotich

Date: November 2025

TABLE OF CONTENTS

I. DECLARATION AND CERTIFICATION

We, the undersigned, declare that this project report, titled **AI-Driven Wafer Defect Detection and Classification Report**, is an authentic record of the group work of **Chipsleauths**, carried out under the technical mentorship of **Faith Rotich** at **Moringa School**, in partial fulfillment of the requirements for the **Data Science** course. The contents of this project have not been submitted to any other Institute or University for the award of any degree.

Role	Name	Signature	Date
Group Leader	Valerie Kigo		
Group Member	Joel Gitonga		
Group Member	Lawrence Kamerino		
Group Member	Hudheyfa Mohamud		

This research project report has been submitted for examination with my approval as Technical Mentor.

Technical Mentor: Faith Rotich

II. ABSTRACT

Semiconductor wafer manufacturing is highly sensitive to defects, which severely reduce yield and increase production costs. This project addresses the inefficiency of manual quality control by developing an automated system for classifying wafer defects from map images. The project utilized the **WM811K Wafer Map Dataset** (initially 811,457 records), sampling **300,000 rows** for modeling.

A total of four models were evaluated:

1. **Logistic Regression** achieved a baseline accuracy of **70%**.
2. **Random Forest Classifier** improved performance to **80%** accuracy.
3. **XGBoost Classifier** achieved the highest classic model test **Accuracy of 81%**.
4. The initial **Convolutional Neural Network (CNN)** architecture achieved a lower validation **Accuracy of 66%**.

Based on these results, the **XGBoost Classifier** is identified as the best-performing model for deployment in this iteration, validating the strength of hand-crafted feature engineering for this particular dataset over the initial deep learning architecture. However, since we are particularly looking into mapping defects we opted to use CNN as it analyses and learns images.

Chapter 1: Introduction and Business Understanding

1.1 Background

Semiconductor manufacturing relies on highly precise steps to fabricate functional circuits onto a wafer. Even minor defects on a **die** (the individual functional circuit block) can lead to product failure, necessitating expensive rework or scrapping, thereby reducing the manufacturing yield. Traditionally, quality assurance has depended on manual or outdated rule-based inspection systems. These methods are slow, subjective, and unable to scale with the increasing complexity and volume of modern semiconductor fabrication. Leveraging **Artificial Intelligence (AI)** and computer vision, this project aims to create an automated defect detection system that analyzes wafer map images for early and accurate intervention.

1.2 Problem Statement

Semiconductor manufacturers are challenged by the need for a scalable, automated solution to accurately and efficiently classify subtle defect patterns in wafer maps. Current manual and conventional systems are incapable of keeping pace with high-volume production and often miss complex or rare defect types.

The core problem is to develop a machine learning model that can effectively learn and classify the distinct spatial signatures of common wafer defects to improve overall yield and significantly reduce quality control time.

1.3 Business Objectives and Goals

The overarching business objective is to enhance quality assurance and production efficiency in the semiconductor manufacturing process through automation.

The project goals are:

1. To identify and classify common wafer defect types (e.g., Center, Edge-Ring, Scratch) using image pattern recognition.
2. To build a **deep learning-based image classification model** using the WM811K dataset that can categorize the 8 distinct defect classes.

3. To integrate the model's predictions into an interactive **Streamlit dashboard** for real-time visualization and classification.

1.4 Success Metrics

The project's success is defined by achieving robust technical performance and measurable business impact:

- Achieving a high **Accuracy** and **F1 Score** for the final classification model.
- Demonstrating a significant reduction in the time required for defect inspection.
- Improving the detection rate (high **Recall**) for minority and critical defect patterns.

Chapter 2: Data Understanding and Preparation

2.1 Data Source and Description

The project utilized the **WM811K Wafer Map Dataset**, a comprehensive collection of real-world yield data.

Data-Name	**WM811K (Wafer Map Defect Dataset)**
Records	**811,457** wafer samples
Key Feature	waferMap (2D array image representing the defect pattern)
Task Type	Multi-class Image Classification (8 classes)

The dataset exhibited severe class imbalance, with the two non-defect classes ([0 0] and None) accounting for over 96% of the data. The defect types like Edge-Ring (1.17%), Center (0.51%), and Donut (0.07%) were heavily underrepresented.

2.2 Data Cleaning and Sampling

Initial processing confirmed **no explicit missing values** and **no duplicate rows** based on the metadata columns. Array-like categorical values in columns like failureType were converted to string representations for consistent labeling.

A stratified sample of **300,000 rows** was drawn from the original dataset. This step was necessary to enable efficient computational processing while maintaining the original class distributions for a representative analysis.

2.3 Pre-processing and Feature Engineering

Data Standardization

To ensure consistency for both feature engineering and CNN input, all waferMap images were resized and padded to a uniform dimension of **(32, 32)** pixels. The failureType labels were consolidated and mapped to **8 numerical classes** using a Label Encoder.

Feature Engineering and Balancing for Classic ML

For the Classic Machine Learning models, the 2D images were augmented with hand-crafted features:

- **Edge and Texture:** Features derived from **Sobel edge detection** and **Shannon Entropy** were computed to capture structural and complexity information.
- **Dimensionality Reduction:** The processed wafer maps were **flattened** into a 1D vector and scaled using StandardScaler.
- **Oversampling:** To address class imbalance in the training data, the **SMOTE (Synthetic Minority Oversampling Technique)** algorithm was applied to generate synthetic samples for the minority defect classes.

Chapter 3: Methodology and Modeling

3.1 Model Strategy and Splitting

The project adopted a multi-stage modeling strategy: (1) establish a linear baseline, (2) test non-linear ensemble models using feature-engineered data, and (3) evaluate a dedicated Deep Learning model for comparison. The sampled data was split according to the existing trainTestLabel, resulting in **20,073 training samples** and **43,803 test samples** (prior to oversampling).

3.2 Classic Machine Learning Models (Feature-Engineered)

Classic models were trained on the SMOTE-balanced feature set but evaluated on the original, non-oversampled test set to prevent data leakage and provide a realistic performance metric.

3.2.1 Logistic Regression

This model was used as the **initial baseline** to gauge the separability of the defect classes using a simple linear approach on the engineered features. It achieved a baseline **Test Accuracy of 62%**, indicating that a simple linear model was insufficient for the complex spatial patterns.

3.2.2 Random Forest Classifier

As a non-linear ensemble model, Random Forest was introduced to capture more complex interactions between the features. It achieved a **Test Accuracy of 81%**, showing a significant gain over the baseline by successfully utilizing the non-linear feature space.

3.2.3 XGBoost Classifier

This gradient-boosting model was employed to further optimize the performance of the tree-based classifiers. The **XGBoost Classifier** achieved a **Test Accuracy of 81%**, making it the best-performing classic machine learning model.

3.3 Deep Learning (CNN) Model

A Convolutional Neural Network was employed to automatically learn spatial features directly from the standardized **(N, 32, 32, 1)** wafer map images, eliminating reliance on hand-crafted features.

CNN Architecture

The sequential architecture was composed of three convolutional blocks, each utilizing **Conv2D**, **BatchNormalization**, and **MaxPooling2D**, followed by a **Flatten** layer and a **Dense** layer with **Dropout (0.5)** before the final **Softmax** output layer (8 units). The model was trained for **100 epochs** with **Image Data Augmentation** for improved robustness.

Chapter 4: Results, Conclusion, and Recommendation

4.1 Model Evaluation and Performance

The performance metrics clearly show the effectiveness of both ensemble methods and the limitations of the initial CNN architecture compared to the feature-engineered classic models.

Model Category	**Specific Model**	**Key Metric**	**Value (from notebook)**
Classic ML (Baseline)	**Logistic Regression**	Test Accuracy	**62%**
Classic ML	**Random Forest Classifier**	Test Accuracy	**81%**
Classic ML (Best)	**XGBoost Classifier**	Test Accuracy	**81%**
Deep Learning	**CNN**	Validation Accuracy (Peak during training)	**72%**

Although XGBoost achieved a higher validation accuracy (81%) compared to the CNN model (72%), the CNN was selected due to the inherent nature of the data and long-term modeling objectives. CNNs are specifically designed for image analysis, enabling automatic extraction of spatial and hierarchical features directly from raw pixel data—capabilities that XGBoost lacks, as it depends on manually engineered features. Moreover, CNNs support transfer learning and visual interpretability through activation maps, which are crucial for identifying defect regions in wafer images. Despite the initial lower performance, the CNN is more appropriate for this image-based task and is expected to improve with further data augmentation and fine-tuning.

4.2 Visualizations and Analysis

Visual analysis was critical in understanding the models' performance and the nature of the data.

- **Wafer Map Samples:** Initial visualizations involved plotting the eight primary defect types (e.g., Center, Edge-Ring, Scratch, Random) to highlight the distinct spatial patterns that the models needed to learn.
- **Confusion Matrices (Classic Models):** Confusion matrices were generated for the Logistic Regression, Random Forest, and XGBoost models to illustrate classification errors. The matrices clearly showed the progressive reduction in misclassification from the linear **Logistic Regression** to the non-linear **XGBoost**. Even with the **81%** accuracy, the **XGBoost** matrix highlighted persistent confusion between visually similar defect types like Edge-Loc and Edge-Ring.
- **CNN Learning Curve:** The training history plot for the CNN was particularly insightful, revealing that the validation accuracy leveled off at **72**early in the training process. This indicates potential issues such as architecture inadequacy or underfitting, which prevented the model from capturing the complex spatial representations effectively.
- **Feature Importance (XGBoost):** Feature importance plots for the XGBoost model revealed that specific engineered features (e.g., die size, moments of inertia, and entropy measurements) were highly critical for the model's high predictive power, offering model interpretability.

4.3 Conclusion

The project successfully developed a classification system for wafer defects, meeting the business objective of automating quality control. Based on the final performance metrics, the **XGBoost Classifier**, with an **81% accuracy**, is validated as the most accurate and reliable model for immediate deployment in this project iteration. The superior performance of the XGBoost model over the initial CNN architecture (66% accuracy) suggests that the quality and relevance of the engineered features were key to achieving high classification performance.

4.4 Recommendation and Deployment

The following steps are recommended to transition the project into a production environment:

1. **Deployment of Best Model:** The trained **XGBoost Classifier (xgboost_improved.pkl)**, with its **81%** accuracy, should be deployed as the core classification service within the proposed Streamlit dashboard.
2. **CNN Optimization:** Future work should prioritize optimizing the CNN model's architecture (e.g., deeper layers, different kernel sizes) and training regimen (e.g., using transfer learning or more advanced image processing) to overcome the current performance limitations and potentially exceed the 81% accuracy achieved by the XGBoost model.
3. **Model Maintenance:** All necessary model assets, including the **XGBoost model**, the **Random Forest model**, the **CNN model**, the data scaler, and the label encoder, have been saved for deployment.

V. References

The following sources and methodologies were integral to the execution and findings of the AI-Driven Wafer Defect Detection and Classification project:

Wafer Map Dataset: WM811K Wafer Map Dataset, a comprehensive collection of real-world yield data utilized for model training and evaluation.

Modeling Framework: XGBoost Classifier, the gradient-boosting model employed, which achieved the highest test accuracy (81%) among classic machine learning models.

Deep Learning Architecture: Convolutional Neural Network (CNN) architecture, a sequential model designed to automatically learn spatial features from the standardized wafer map images.

Data Balancing Technique: SMOTE (Synthetic Minority Oversampling Technique), the algorithm applied to generate synthetic samples for minority defect classes to address severe class imbalance in the training data.

Feature Engineering Methods: Techniques such as Sobel edge detection and Shannon Entropy were used to augment the 2D images with hand-crafted features for the Classic Machine Learning models.