National College of Ireland

Automated Detection of Semi-Conductor Wafer Map Defects Using Machine Learning Techniques

MSc Research Project
Data Analytics

Michael Ward
Student ID: x20190212

School of Computing
National College of Ireland

Supervisor: Zahid Iqbal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Michael Ward
Student ID:	x20190212
Programme:	MSc in Data Analytics **Year:** 2022
Module:	MSc Research Project
Supervisor:	Zahid Iqbal
Submission Due Date:	15/12/22
Project Title:	Automated Detection of Semi-Conductor Wafer Map Defects Using Machine Learning Techniques
Word Count:	8151 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Michael Ward

Date: 13/12/22

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automated Detection of Semi-Conductor Wafer Map Defects Using Machine Learning Techniques

Michael Ward
x20190212

Abstract

The process of manufacturing and creating silicon wafers for the development of chip sets is a particularly meticulous task. Full automation is used in many semiconductor manufacturing facilities. This results in minimal human intervention as batches of wafers move from processing tool to processing tool using robotic systems such as overhead vehicles (OHVs). This ensures the wafers have a low contamination possibility and ensures a high level of efficiency with tool-to-tool time (T2T). As is the case with many processes that are fully automated, issues can occur where certain parts of the robotic systems can become misaligned. This then has the potential to cause damage to the wafers in various ways. When a wafer undergoes an analytical process where the wafer is scanned for particles, the analytical tool will publish a wafer map that contains the location of the particles. Technicians manually review these wafer maps to check for possible issues but due to the sheer amount that is generated, many issues go unnoticed until it is too late. The aim of this research is to create a system and model using machine learning that can automatically detect and classify issues as soon as the wafer maps are generated. Due to the nature of the system, many machine learning classification models were researched and reviewed for the needed functionality and the speed. Support Vector Machine (SVM) and neural network sequential classification models are used due to “One vs One” approach being a good option and the high accuracy rate of the sequential model. A graphical user interface then alerts any stakeholders of excursions related to the equipment from the data generated on the analysed wafer maps.

Keywords: Defects, Wafer Maps, machine learning, semiconductor industry, classification, SVM, CNN, Sequential Neural Networks

1 Introduction

Wafer defects are a constant problem in the semiconductor industry. Many different types of defects can occur on a wafer. This can range from a sectional build-up of particles to wafer scratches and impacts. The cause of these defects is the result of many different factors. Early notification to engineers and technicians of the problem as soon as the issue occurs has the potential of saving hundreds of thousands of euros from a possible wafer scrap event. The quality of a wafer and eventual quality of the processors that comes from them depends entirely on the amount and types of defects that are on it. The quality of the wafer die can be affected by any type of defect. Die is what ultimately becomes the processors. Dependant on the size of the die, wafers can contain hundreds to thousands of die. A tool or robot excursion

can cause defects on the wafers making the die defective. This then affects the wafer yield. The number of usable die when the wafer reaches the end of the line is known as the yield. The wafer will not contain enough die to be profitable if too many die is defective. The result of this is wafer scrappage. As the wafers are processed in batches, all the wafers in that batch would be defective from processing on the same processing tool. The cost of the scrappage would be added to the cost of the end of line yield, increasing the price per wafer which in turn, increases the price per die then waterfalling to the final processor units.

1.1 Motivation

Semiconductor processing and manufacturing is a completely automated process. Firstly, the vendor brings the wafers into the fabrication facility. From here they enter the production environment and process flow. The process flow contains hundreds of production tools that will process the wafers from start to finish. Before the wafers can enter the production environment, they are inserted into a Front Opening Unified Pod (FOUP)[1] . The capacity of a FOUP is 25 wafers maximum with an approximate weight of 10kg. To transport the FOUPs around the fabrication facility, a system named Automated Material Handling System (AMHS)[2] is used to transport FOUPs to and from each processing tool. A Wafer Handling Robot (WHR)[3] then removes the wafers in single file from the FOUP and places them within the processing tool. The WHR contains a robotic arm called an end-effector.[4] End-effectors can have many shapes. In the handling of semiconductor wafers, contact between robotic equipment and wafers should be at a minimum. For this reason, the end effector is small but positioned in a manner that ensures minimum contact. Some end-effectors can look like small robotic pinchers. It can pick up a wafer by applying a small amount of force that is enough to prevent any pressure damage.

The wafer is then inserted into the processing tool for processing. The same process is followed in reverse order for the removal of wafers from the processing equipment back to the FOUP. As there are a high amount of components involved in the transportation and movement of the wafers, a many number of problems can arise that could damage the wafer and result in defective die. A common occurring defect is the edge location defect. The usual root cause for this defect is a WHR misalignment with the tool entry point. The WHR can push the wafer at an oblique angle which can result in the wafer partially colliding with the processing chamber doors. A consequence of this is that particles would spread around the impacted locations of the wafer which would be the edge locations. The die in these locations would be defective and unusable.

The main problem with this is that this issue like many others can affect hundreds of wafers on many lots before it becomes noticed through a manual wafer map[5] inspection. Another wafer excursion that is less common is wafer scratches. These types of issues can occur for many reasons but due to its rarity it can go unnoticed for thousands of wafers causing ripple effects within the production line. Resulting in a large number of semiconductor wafer scrappages and bad defective yield.

---

[1] https://www.entegris.com/shop/en/USD/Products/Wafer-Handling/Wafer-Processing/300-mm-Front-Opening-Unified-Pods-(FOUPs)/c/300mmfrontopeningunifiedpodsfoups

[2] https://www.daifuku.com/solution/cleanroom/

[3] https://yaskawa.co.il/en/product/wafer-handling-robots/

[4] https://www.automate.org/news/what-is-an-end-effector-and-how-do-you-use-one

[5] https://www.confovis.com/en/solutions/semiconductors/aoi-automated-optical-inspection

1.2 Research Question

"Can semi-conductor wafer map defects be automatically detected using machine learning classification techniques?"

This research aims to build a system with the ability to detect defects when the wafer map data becomes available. This will be accomplished using machine learning image classifications models. A system like this will alert stakeholders to possible process equipment excursions and reduce the needed human workload. This is also a work content reduction tool that can free much needed user time and resources to work on other objectives.

2 Related Work

This section covers previous work researched and implemented using image classification techniques. It also covers prior work done defect classification on wafer maps. Work from many different authors and researchers are examined and compared to develop an understanding to support how the research achieved its goals. Machine learning models are evaluated to understand its performance and efficiency.

2.1 The Study of Image Classification

The classification of images can be a challenging job. Numerous models and methods have been created to a varying degree of success which have many application types. An example model of this is the Bag of Features (BOF) paradigm used by (Huang & Chung, 2016). Inspiration for this came from the Bag of Words (BOW) model in which the frequency of vocabulary words in a document is represented by document classification fields. BOW is used in computer vision systems for the purpose of classifying images. This is essentially representing image features as words. BOF is an enhanced version developed by (RongAn Chen & Qu, 2014). The research conducted by the developers emphasised that the inverse document frequency[6] and the term frequency[7] fails to contain image class information. An adaption of a model known as Chi Square that does hold class information of images is integrated into the tf-idf model. the total enhancement of the model becomes known as the tf-idf-chi-exp model. Scale Invariant Feature transformation (SIFT) is utilized for feature extraction from the images. Feature experimentation shows the tf-idf model performs significantly better than BOW. A large amount of image classification is done using deep learning techniques. (Loussaief & Abdelkrim, 2018) completed research which sought to evaluate how BOW in machine learning compares to a deep learning classification model. As both models are compared, it is highlighted that image encoding and feature extraction are the biggest issues with image classification and computer vision. Convolutional Neural Networks (CNN) is the deep learning method is used. An AlexNet CNN model on the imageNet dataset to a high accuracy. The conclusion of the evaluation of mode models shows that the CN model had a significantly higher performance than that of the BOF model. The results of the performance are evident from the number of experiments undertaken. The classification accuracy of the nearest neighbour was investigated. The tests conducted for the experiments looked at the cosine, medium, coarse, cubic and weighted K Nearest Neighbour accuracy. BOF had on average 39% accuracy when compared to the CNN model with an accuracy result of 92%. The delta between both models is massive and shows how deep learning networks are far superior to BOF in image classification and feature extractions.

[6] https://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html
[7] https://www.opinosis-analytics.com/knowledge-base/term-frequency-explained/

These results are also validated by work conducted by (Panigrahi, et al., 2018). A CNN model was recreated that could continuously optimise with many different parameters. Over 8000 pictures of cats and dogs which were labelled was classified by the utilized parameters. The purpose of the model was to train and learn dog and cat features. Once this was done an Artificial Neural Network (ANN) classifier was used for the classification. The obtained results were 88%. Deep Learning Networks for image classification purposes popular amongst developers and analysts alike. A small form of CNN was used by (Shyava Tripathi, 2019) for feature and object identification within an image. The authors propose in the paper that a small CNN has the capability to classify images to a high accuracy. It is also suggested that the small CNN would be less complex for any tested dataset. The starting results for the training accuracy was very low. This was due to a large number of classes from the tested dataset. Overtime, the model trains itself on previous data and eventually reaches 99% training accuracy with a 0.12% validation loss. Research by (Winoto, et al., 2020) shows that a small, trained CNN has many different applications. The Researchers showed how Deep Convolutional Neural Networks (DCNN) can be slimmed and compressed for use with mobile devices. These so called Small DCNNs can run on mobile hardware or embedded micro controllers. Small DCNNs were achieved by reducing the computational complexity, this in turn enables the processing time to be faster. Floating point Operation per Second (FLOP) is threshold limited to a minimum and maximum. This is done to ensure the harder it runs, it is not overwhelmed with activity. A high average of 86% accuracy is achieved through this model. The researchers mention that the accuracy isn't as high as state-of-the-art models. The overall accuracy and design is still very impressive given that the model is meant to be run on low powered computer devices. CNN is widely used in many classification tasks, (Adly, et al., 2014) proposed how CNN can be used for the classification of semi-conductor defects.

2.2 Defect Detection on Wafer Maps

When wafer maps are reviewed and defects are noticed, it could be a sign that a processing tool has an underlying issue. This can range from simple mechanical part replacements to robot recalibrations that could be needed. A great amount of research has been undertaken in this area. The reason for this is that subtle changes in the alignment of some mechanical systems within the processing tool can result in damage to hundreds of wafers which could cost hundreds of thousands of euros. This is something that was studied and researched by (Muhammad Saqlain 2020). The research details how unreliable over time manual defect identification techniques are. The researchers propose a deep learning based convolutional neural network for the purpose of identifying defects. The proposed method to accomplish this was to use convolutional layers for feature extraction. The model that was developed gained an accuracy of 96.2%. this was obtained from 9 different classes with real world examples used. Similar research has also been completed by (Naigong Yu 2019) where an 8-layer CNN model was developed for defect classification. The had an average accuracy of 93.13%. This is a good classification performance and just a little under the previous model. An interesting research paper published by (Shu-Min Li 2020) aimed to predict potential wafer scratches with the use of a CNN model. It was observed by the researcher the difficulty of predicting certain types of wafer map patters as there is no true patterns and wafer maps tend to be full of noise as random particles can spread anywhere on the wafer. It was noted that this is a problem for many test engineers in that defective die can cause leakage to surrounding die. This can inhibit potential die performance or even destroy it altogether. Their solution was to locate and mark potential scratch patterns using machine learning. As a result a CNN model was created that could accomplish this with 89% accuracy. The researcher explains that this eventually leads to better quality products at end of line. The

quality of the generated wafer maps plays a huge role in the classifying of defects on wafers. The higher the quality the easier a defect can be seen. This is also true in terms of machine learning algorithms and is highlighted by (Yanh li 2021). The researchers developed a method to target small features on the wafer map. The accuracy for this is 87% with the researchers mentioning that the quality of the wafer plays a very important role. Work completed by (Xiaoke Cao 2021) used the YOLOv3 (You Only Look Once)[8] method for defect detection. This model is used for object detection in real time. Features are trained from a CNN. The average detection accuracy is 85.1%. The resembles the work completed by (Jaegyeong Cha 2020) which used Xception[9] CNN that contains 72 layers to classify wafer that contain many defect types. The classification accuracy for this model was 93.7%. (Batool, et al., 2021) gives an overview of many different applications for machine learning with semi-conductor wafers. These range from particle selection to positional density defects.

2.3 Machine Learning in the Semi-Conductor Industry

The semi-conductor industry generates a massive amount of data daily. All of this data is stored for use internally. The data is warehoused so it can be accessed by engineers and technicians for both reactive and proactive data pulling. This also opens the opportunity of allowing machine learning to perform some of its own experiments. Research completed by (Chen He 2017) discusses the possibility of enhancing yield with the implementation of automations that constantly monitor for failures. The discussion highlights how a machine learning algorithm could map the failure patterns of yield. This would be able to help support those in decision making roles. The same authors mention an enhanced version of this (Chen He 2021). The authors discuss how the semi-conductor industry is effectively a big data environment with an immense amount of machine learning possibilities. The term, “Smart manufacturing” is mentioned in this paper which happens to be the route most semi-conductors take to adapt the methodology of decision automation. Various types of machine learning methods can be used for decision making. (Moriya 2021) shows how manufacturing processes can be improved with the use of machine learning. Regression algorithms are used on PEALD (Plasma Enhanced Atomic Layer Deposition) film thickness processes. An experiment was conducted on decision making between the machine learning algorithm and a process engineer. The experiment conducted had wafers that had the same conditioning and non-uniformity. Five trials were taken to measure the performance of the engineer and the machine learning model to settle wafer variation on non-uniformity. The engineer was unsuccessful where the machine learning model was successful. The results showed that the machine approach could settle the variable quickly and with optimal measured conditions.

2.4 Non-Classification Use Cases in the Semi-Conductor Industry

Machine learning is used in this industry for use cases other than that of classification. (Daewoong An, 2009) shows how yield is predicted using a machine learning model. But instead of using a neural network an SVM method was used. The reason for this as per the authors is the issue with overfitting. SVM was mathematically easier to understand and integrate. Another interesting feature used is Out of Control (OOC) detection. A paper by (Ilham Rabhi, 2021) how SVM is used for the prediction of values going below or over set control points. This is used in conjunction with processing tool sensors that continuously measure values. The data is fed to the SVM model which retuns a quick decision. Other uses

[8] https://viso.ai/deep-learning/yolov3-overview/
[9] https://keras.io/api/applications/xception/

cases such as manufacting flow simplification. (Felix Pistorius, 2020) proposes the use of a smart evaluation matrix that can help choose the best models and eliminate unneeded tasks.

This literature reviews covered the study of many papers that apply to the classification of wafer maps. It also highlights the work done in the field of classification for many purposes. Other use cases for machine learning in the semi-conductor industry is also studied. The application of various machine learning models is covered to conclude how the machine learning models are applied. The majority of work covered is in the neural network scope, but some SVM models were also covered. The industry now includes many of these methods.

3 Research Methodology

The problem that this project is trying to solve involves a great amount of data that is continuously being generated. A method that can stem informative results should be used. In the case of this research, it was achieved using the Cross Industry Standard Process for Data Mining (CRISP-DM) method as seen in Figure 1 below. To accomplish this the following tasks were taken to answer the research question as accurately as possible.

- Analysed the research question.
- Sourced data that is relevant that helped answer the question.
- Decided on an algorithm.
- Prepared data to suit the algorithm.
- Split the data into training and test datasets to build models around the training data.
- Validated and evaluated the models on the test data.

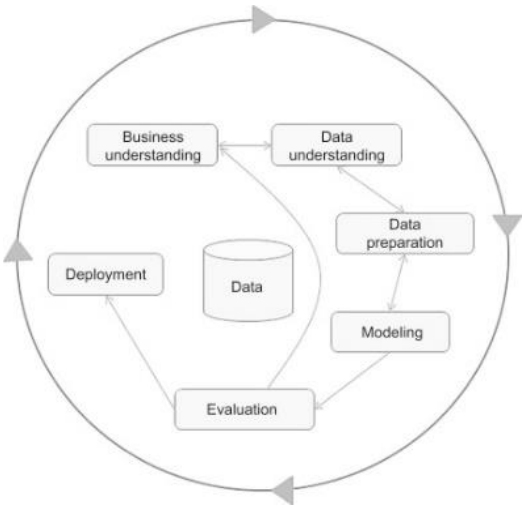

Figure 1: Crisp-DM methodology used

3.1 Dataset Selection

The dataset used for this research was obtained from MIR Corpora. The name of the dataset is MIR-WM8-WM811K.[10] The data was released and collected from various semiconductor fabrication facilities. The data was release publicly for the purpose of research. Exploratory

[10] http://mirlab.org/dataset/public/

analysis on the dataset returns 811457 wafer map records. As the dataset is so large each record is stored in an array of 26x26 comprising of 0s,1s and 2s. The 0 represents the wafer map background. The 1 represents the wafer. The 2 represents the defect on the wafer. Visualizing a random sample of the records displays the many different types of wafer map defect excursion ranges that can occur, this can be seen in Figure 2 below.

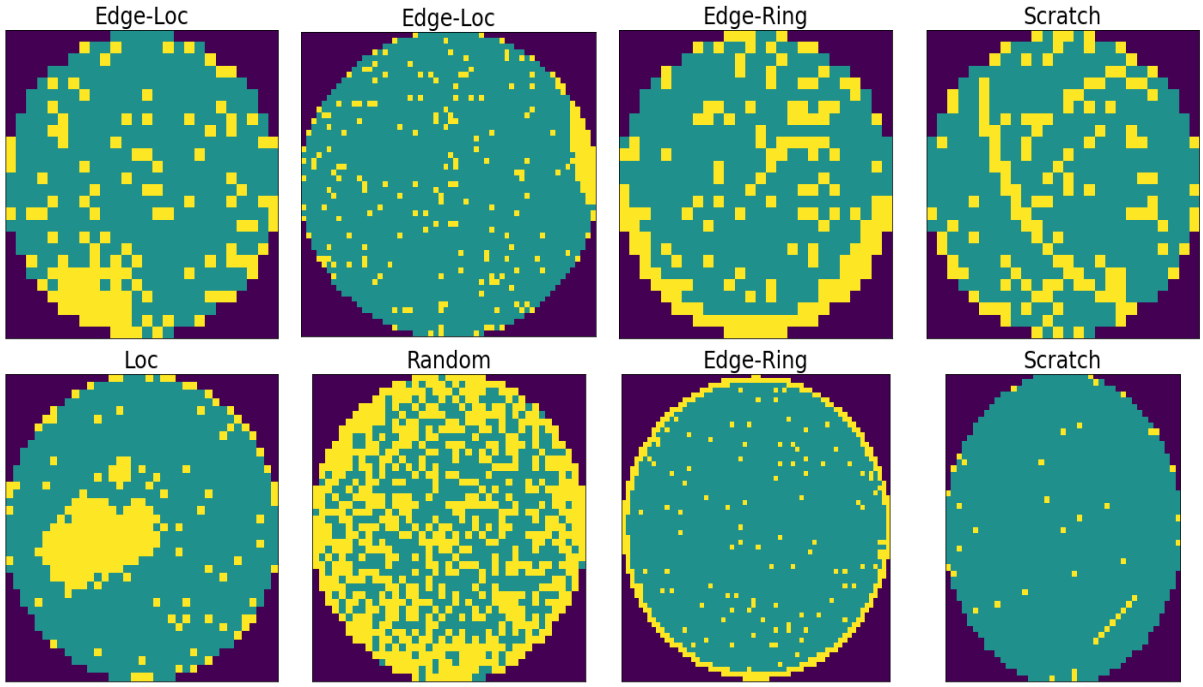

Figure 2: Wafer map excursions

Much like how the array values represents each part of the wafer map, The different colours represent those same values of the array but visualized. Background is purple, wafer is green while the defects are yellow. The random defect type is the most common occurring wafer defect. Surface scans that the wafer undergoes when removed from a processing tool shows the position of the defects on the wafer (Chiang, 2006). Defects that are random usually show as spread across the wafer. The labelled random defect in Figure 2 would be an extreme case of excursion for that type of defect. The possible causes for this would most likely be from processing injector burst. The excursion even though labelled Random, would be more inclusive to a Near-Full failure. Many of the other label defects are less common and has the potential to go unnoticed for numerous days which results in many wafers in many lots being affected by the defect excursion. These failures then contaminate further flowing processing tools in the production line with the high particle count from the defect excursions. A frequency chart is generated to see how much of each failure type is contained within the dataset. The chart shows that the dataset has a high distribution imbalance as can be seen from Figure 3. The Edge-Ring defect has the highest occurrence of defect in the dataset. These types of defects are mostly caused by defective robotics and transfer modules (Huang & Chung, 2021). It’s worth noting that the defects with the higher distribution in this dataset are not as common in the real-world semiconductor manufacturing industry. Research into these less common occurring defects would have a greater business impact and benefit. It's for this reason that the dataset has a greater distribution of these defect records than the other defects.

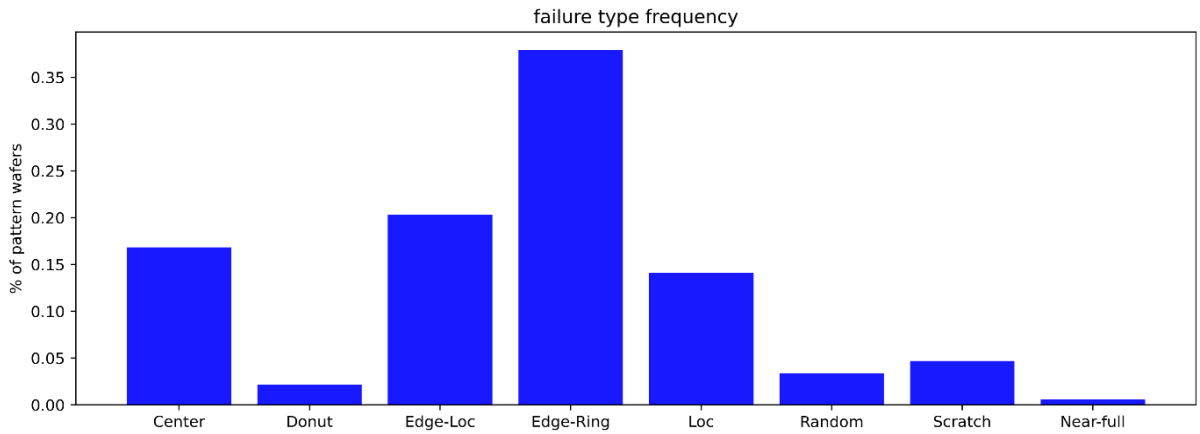

Figure 3: failure type frequency

3.1.1 Data Pre-Processing

Before fitting any models, the dataset needs to be investigated and cleaned for the purpose of removing any unnecessary attributes and features.[11] This will ensure they will fit correctly to the created model. Investigating the data, it can be seen that there are 811,457 wafer maps. It also shows that the data is taken from 47,543 lots. Each lot has a maximum capacity of 25 wafers and each lot should be maximized. If each lot had 25 wafers, then the total amount of wafer maps should be equal to 1,188,575. This shows a discrepancy in how many presumed wafer maps should be in the data set by 377,118 wafers. To investigate this further, a bar chart is created to display the frequency of the lots and how many wafers are in them. As can be seen from Figure 4, not all of the lots have 25 wafers contained in them. A paper by (Wu, et al., 2002) highlights why a semiconductor lot might have less than 25 wafers. It's because wafers are sometimes scrapped due to yield problems or defect problems affecting a wafer. A lot with less than 25 wafers is known as a small lot whereas a lot with 25 wafers is known as a full lot.

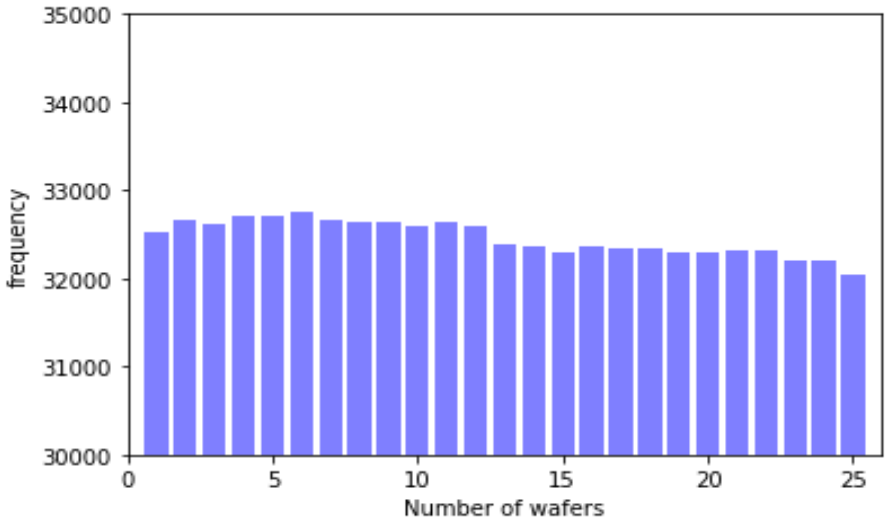

Figure 4: Number of wafers

[11] https://data.gov.ie/edpelearning/en/module11/#/id/co-01

Now that it is known that there is no issue with the dataset, the variable named "waferIndex" is dropped from the data frame as it won't be needed going forward and has no bearing on the training of the model. Most of the data that is needed for the training of the model will be taken from the "waferMap" column. Something that might prove to become an issue is the size of the wafer maps. Looking at available columns, a column named "dieSize" exists. This column shows that the size of each die is different for each wafer map. The different sizes of the wafer map images could cause issues with the training of the model and the extraction of features. To understand this better, the dimensions of the wafer maps are calculated and added to the data frame as "wmDim". A max and min calculation is done on the "wmDim" column which shows that the max dimensions of the wafer maps in the dataset to be (300,202) and a minimum dimension of (6,21). Next, a calculation is done to see how many unique values exists in the column which returns a value of 632. This highlights that there are 632 different dimension sizes for the wafer maps in the dataset. Next, the dataset is investigated for missing values. In this case, the only interesting data available is the wafer maps that contain failure patterns. Therefore, it is beneficial to remove wafer map data that does not contain any failure labels. To accomplish the removal of unnecessary data, the wafer maps are first counted and then separated based on the type of label and whether it contains any defect label. The results of the separation are shown in table 1 below.

Type	Count
Contains no label	638507
Contains a label	172950
Contains a label but no defect	147431
Contains label and defect	25519

Table 1: Separated label counts

As can be seen from Table 1, 638507 wafer maps contain no labels. Of the 172950 wafer maps that do contain labels, 147431 of them contain no defect. This leaves the dataset with 25519 wafer maps that do contain both the label and the defect. This is the total amount of wafer maps that contains the real-world failure samples that will be used for classification purposes. The table above shows a very high imbalance distribution of the wafer maps. A possible reason for the imbalance is that the dataset is intended for more than just defect classification purposes, such as wafer analytics or trend data analysis.

3.2 Feature Engineering

Before the model can be built and data set up for performance evaluation, some feature engineering tasks must be completed on the pre-processed data. The model will be trained on the wafer map data that's already available in the dataset. In order the get the best possible accuracy, many methods of feature extraction need to be undertaken. For this model, 3 methods will be used for feature extraction.

3.2.1 Feature Engineering Radon Transform

This can create two dimensional representations of the wafer maps. Radon Transform[12] can input several projections which will then return radon-based features. Generating these features using a radon transform algorithm generates the plots as seen in figure 5 below.

[12] https://mathworld.wolfram.com/RadonTransform.html

Before values from the generated features can be used, they must be set to a fixed dimension because the wafer maps have many different sizes. This is accomplished using cubic interpolation[13].

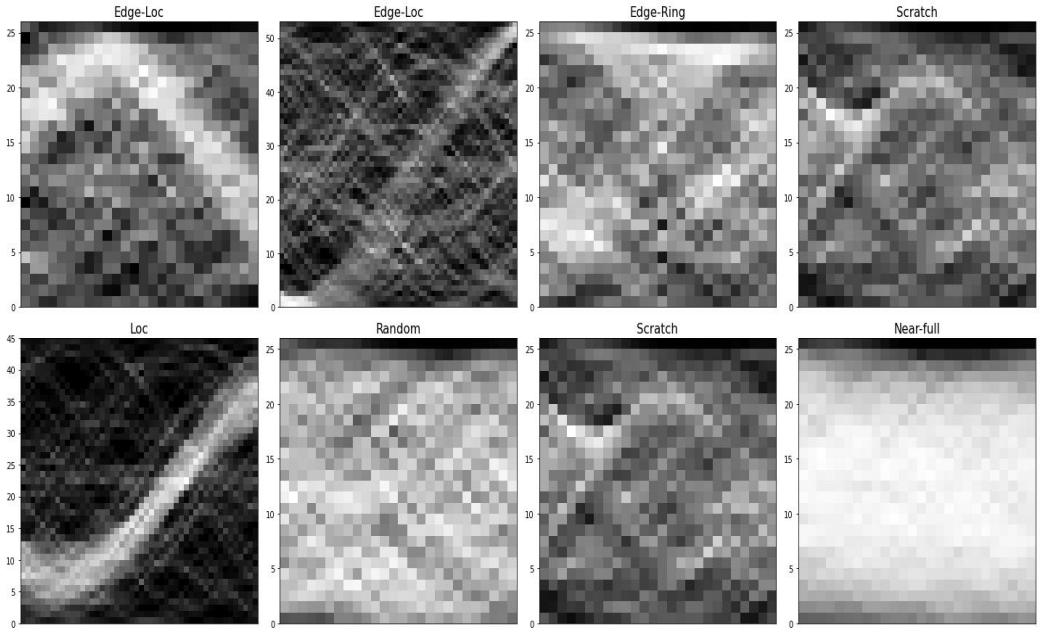

Figure 5: Radon features class extractions

Figure 5 above shows the generated results of 8 random failure types. An example of similarities can be seen from both of the scratch radon transform images in that a match could be made as the two wafer maps for these radon transforms have very different defect sizes. The total number of extracted radon based transformed features is 40.

3.2.2 Concentrated Sections

Given the circular nature of a semiconductor wafer, there are a few different zones where defects can occur. The outer locations of the wafer can be divided into four semi circles while the center of the wafer can be sliced into nine smaller sections. Depending on where the defect occurs on the wafer, the failure locations would have a higher concentration of defects. An example of this would be the “Edge-Loc” defect failure. The particles in the outer zones of the wafer map would have a higher concertation of particles than the center zones. This in turn can highlight the failure depending on the failure location and defect density. Enabling the extraction of density from these zones results in 13 features that can be used for classification purposes.

3.2.3 Salient Region Detection

In its purest form, saliency is the standout features of an image. It’s what the eye and brain make a connection to that focuses on the most important features. In terms of the wafer maps, the most important features are the defect failures. In a sense this feature is a kind of wafer

[13] https:/docs.scipy.org/doc/scipy/reference/interpolate.html

map failure noise reduction in that it will eliminate all non-failure type defects such as particle spread or sporadic patterns and show only the failure such as a wafer scratch, a center localized fail or an edge defect fail. Along with all the full failure wafer maps in the dataset, the salient regions will also be generated and saved locally for the model training purposes thus enabling a high accuracy score as the model will know more of the defect types. For the purpose of this research, a region labelling algorithm is used to select the max area region which in turn is the salient region. Geometry features such as eccentricity can then be extracted. This will return a frame showing the failure defect as opposed to a full wafer map.

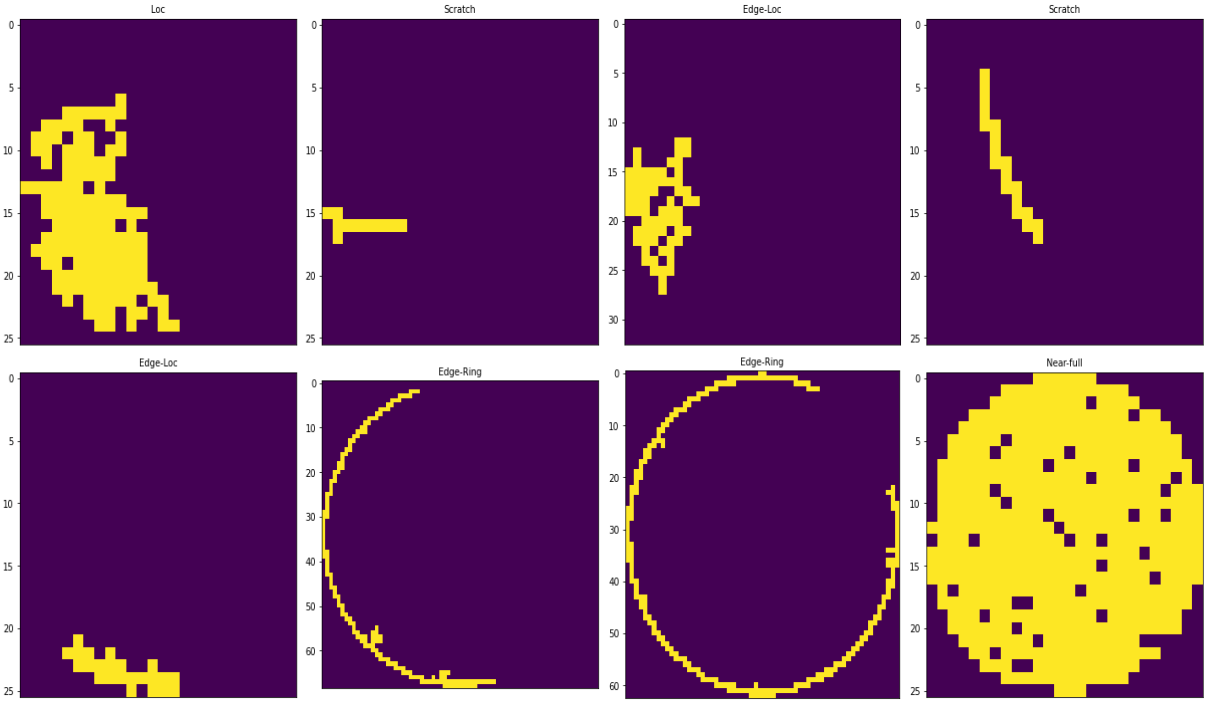

Figure 6: Salient Regions

Performing a salient region detection algorithm on a random selection of 8 wafer maps returns the salient region maps as can be seen in Figure 6 above. It can be noticeed how just the failure defect is shown in the wafer map and how other failure types are somewhat vastly different. The total number of extracted features are 6 from the salient region detection algorithm.

3.2.4 Feature Combination

A total of 59 features are extracted from the dataset. The 3 different types of features are added together for the purpose of classifying the wafer maps. This high number of features gives the model a higher chance of correctly classifying the wafer maps when they are tested.

3.3 Evaluation

The evaluation metrics that are used in this project are accuracy precision, recall, F1 Score and confusion matrix. Using these evaluation metrics will ensure an accurate representation of performance. This has been accomplished by plotting the algorithms performances using a confusion matrix. The rest of the performance identifiers data is generated in the

classification report. F1 precision and accuracy scores are calculated in a manner needed to measure the model performance. The calculations and formulas for the given evaluations can be seen in the descriptions below.[14]

- **Precision –** This is the ratio of classes that have been predicted correctly over the total number of positive predictions taken by the model.
  Precision = True Positives / (True Positives + False Positives)

- **Recall** – This is ratio of observations correctly positively predicted over all positive class observations.
  Recall = True Positives / (True Positives + False Negatives)

- **F1 Score** – This is the weighted average between precision and recall
  F1 = (2 * Precision * Recall) / (Precision + Recall)

- **Accuracy** – This the ratio of correct predictions over the total number of predictions
  Accuracy = (True Negatives + True Positives) / (True Positives + False Positives + True Negatives + False Negatives)

4 Design Specification

To implement this project efficiently, a three-tier architecture was designed as can be seen from Figure 7 below. It contains a brief overview of all the steps and technology used for the development of the model and the creation of the graphical user interface for use by the users.

- **Tier 3** – This tier of the design spec shows that when the data source is selected, it is then imported and collated. The usable data is then selected from this using the Python programming language within the Spyder integrated development environment. The data is then split into training, test, and validation data this is efficiently done which then brings the process to the next tier of business logic flow.

- **Tier 2 -** As part of the previous tier, all the wafer maps get exported to a local location on the user's hard drive. Once they are saved, they are imported with directory name acting as the class name. the next step is the data will undergo augmentation. Once completed then the images will be resized to a universal size. Features are then extracted from the images. The SVM and CNN models are then trained on the data and the features to see which model has the better performance.

- **Tier 1 –** This is the presentation tier. It highlights how the data will be displayed. The analytical data is shown within the Integrated Development Environment (IDE) while the end user information such as the classification and imported wafer map is shown in the graphical user interface.

[14] https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

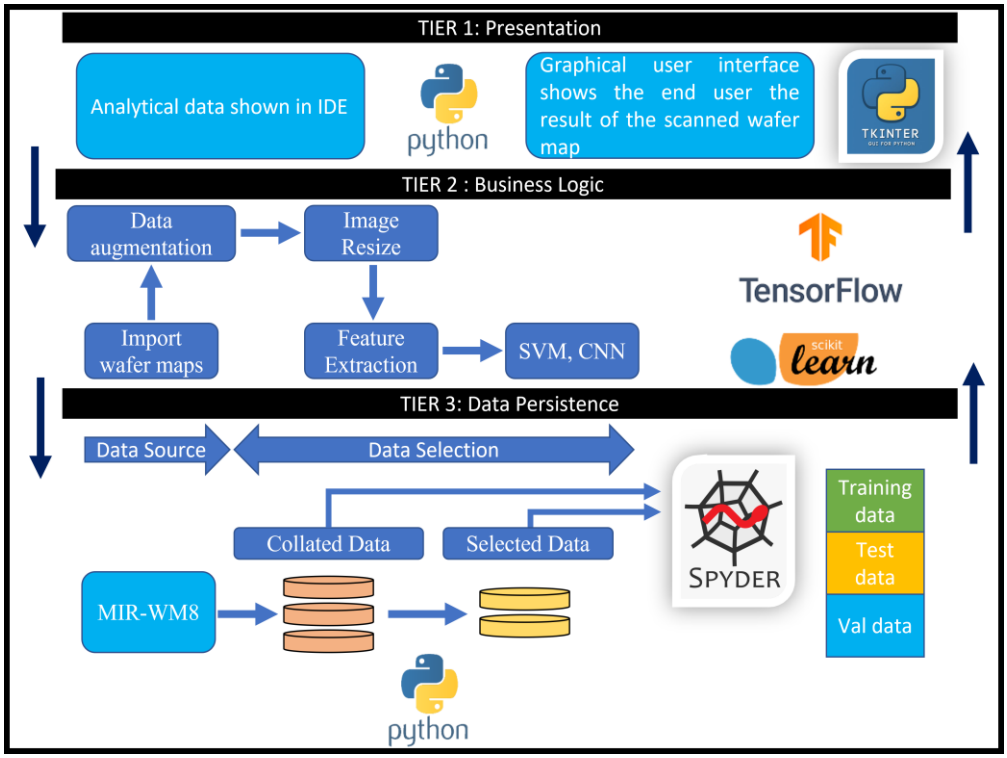

Figure 7: Wafer Map classification design specification

This design approach was taken to better meet the project requirements and ensure an efficient performance of the created models. 8 classes and 25519 defect wafer map images are extracted and are processed and used. The train and test data are created and split using those images. The implementation of this is covered in the next section of the project report.

5 Implementation

This section discusses the implementation, evaluation, and results from each model. The project focuses primarily on two machine learning models. An SVM model and a Sequential model. The dataset, data location, training methods and feature extraction types are also discussed. The evaluation which plays an important factor in the project uses classification reports and confusion matrices to show the performance of the models. The determination of the performance of the models will come from the precision, recall. accuracy and F1 scores. The two models are compared, then the best performing model is used for the final application of the project. An overview of the graphical user interface and how the application works is also examined. The use case will also be examined for the business need.

5.1 Dataset Creation and Manipulation

The dataset used is the MIR-WM8-WM811K obtained from MIR corpora as discussed in section 3.1. For the SVM and sequential model, the dataset contained 8 classes which are Scratch, Random, Near_Full, Loc, Edge Ring, Edge_Loc, Donut and Center. For the purpose of the SVM model, the images generated with plotting are adequate but for the purpose of the sequential model, a more real-world approach is needed for importing data. Therefore, each generated wafer map is exported to the local working directly and placed in defect type folder. The saliency generated wafer maps are also exported to the working directory. From here, they will be imported with directory names being used as classes for training the model.

5.2 Support Vector Machine Model Implementation and Evaluation

Support Vector Machine (SVM)[15] is a supervised learning method that is used for problems related to regressing and classifications. It is a linear model but can also solve non-linear problems. The SVM algorithm creates a line or a hyperplane which can separate data into classes. As this project has many different types of defects it needs to classify, a multi class approach is needed. To accomplish this One-Vs-One (OVO) method is used. OVO is a heuristic method that uses binary classification algorithms for the purpose of multi-class-classification. OVO solves multiclass classification problems by splitting a dataset into one dataset for each class versus every other class.

5.2.1 SVM Model Implementation

The model was split into a train and test dataset. A one vs one classifier was implemented and fit to the model with the split data using the scikit learn python package. All of the extracted features mentioned in section 3.2 of this report are used to be validated against the dataset failures with labels as "X". This is done by first concatenating all of the extracted features into one array variable. Then that array can then be fit to the model alongside the labels as "y".

5.2.2 SVM Model Evaluation

20415 wafer maps were used to train the model while 5103 were used for validation. Upon training the model, a sample of 100 predictions were taken and compared to the train dataset.

y_train_pred[:100]:	[4 0 2 2 0 3 2 0 2 2 6 0 4 0 3 2 3 2 2 4 2 2 0 4 3 0 3 3 3 3 2 3 0 3 2 3 2 4 3 2 2 2 3 3 0 6 3 5 2 3 0 3 2 2 2 0 3 3 4 0 4 2 3 3 3 3 4 3 0 3 2 2 0 2 2 3 3 0 2 0 2 4 3 3 3 3 0 0 2 0 0 3 4 3 0 3 3 2 4 2]
y_train[:100]:	[5 0 4 2 0 3 2 0 2 2 6 0 4 0 3 0 6 2 2 5 2 4 0 5 3 0 3 3 3 3 2 3 4 3 4 3 3 4 3 2 2 3 3 3 0 6 3 5 2 3 0 3 2 2 2 0 3 3 1 0 4 2 3 3 3 3 4 3 0 3 2 4 0 4 2 3 3 0 2 0 2 4 3 3 4 3 0 0 6 0 0 3 4 3 0 3 3 2 4 2]

Table 2: comparisons of train and predictions values

The second column in Table 2 above shows the numbers associated with each class. 0 = Center, 1 = Donut , 2 = Edge-Loc , 3 = Edge-Ring , 4 = Loc, 5 = Random, 6 = Scratch , 7 = Near-full. It can be manually observed that a number of predictions are wrong while quite a few are correct predictions. For more detail on performance, a classification report is needed.

Upon completion of training the SVM model. A classification report was generated to show how the model performed at classifying the wafer maps. As can be seen from Table 3, the results vary among each class. The report shows that the classes with the more training support generally had a higher f1-score compared to the classes with the lower training support. A probable exception for this would be the "Loc" class. It has a relatively high training support but an F1 score of 50%. This is also in line with its recall and precision score. A possible reason for this is that it resembles some of the other classes. "Loc" is a shortened term for localized. They are usually large spots of defects which can occur on

[15] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

mostly centralized positions of the wafer. Sometimes “Edge-Loc” defects can resemble “Loc” but just a little further outside the centralized perimeter resulting in an incorrect classification.

	Precision	recall	f1-score	support
Center	0.88	0.95	0.91	3238
Donut	0.94	0.04	0.08	404
Edge-Loc	0.59	0.74	0.66	3860
Edge-Ring	0.9	0.92	0.91	7299
Loc	0.58	0.5	0.5	2677
Random	0.82	0.64	0.72	640
Scratch	0.77	0.34	0.47	905
Near-full	0.9	0.72	0.8	116
accuracy			0.77	19139
macro avg	0.8	0.61	0.64	19139
weighted avg	0.78	0.77	0.76	19139

Table 3: Classification report of SVM accuracy results

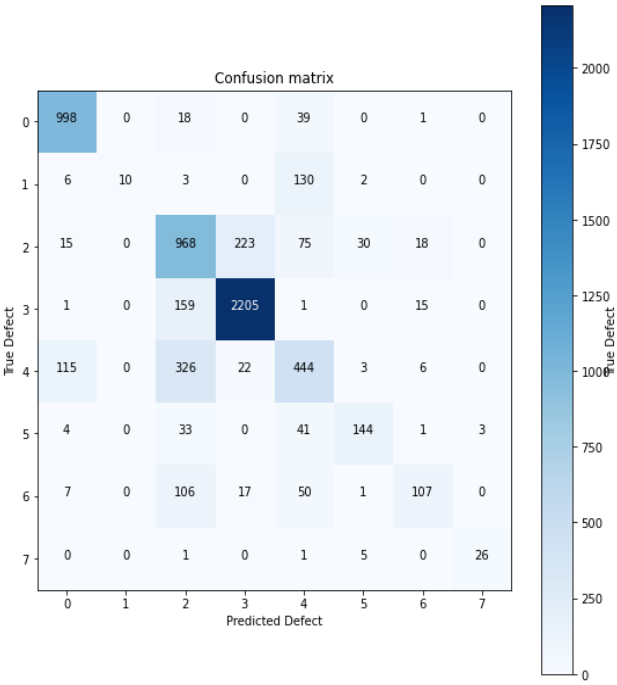

Figure 8(a): SVM Confusion Matrix

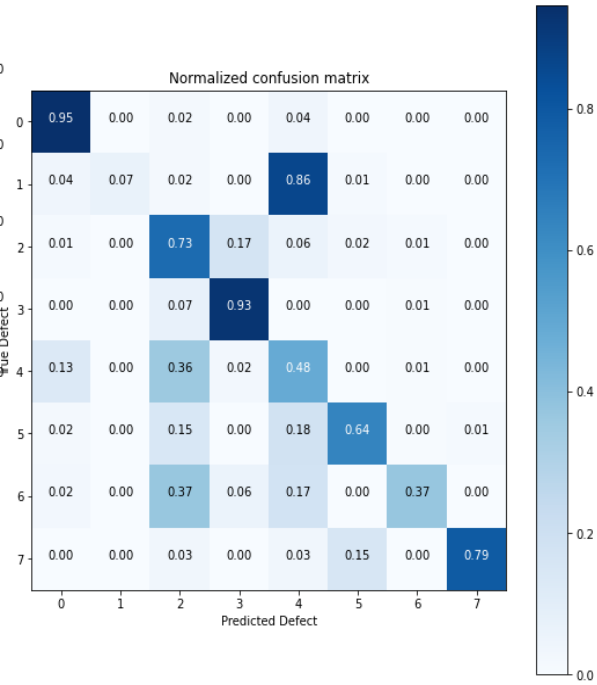

Figure 8(b): Confusion Matrix Normalized

Next, to also examine the performance of the model, a confusion matrix is generated. First, a basic confusion matrix is generated which shows the total amount of correct and incorrent predictions as seen in figure 8(a). The best performing class is the Edge Ring class as represented by the number 3. This can be attributed to the higher number of training wafer maps available for this defect. To better visualize and understand the confusion matrix, it is better to normalize the data and re generate the plot. A normalized confusion matrix is created as can be seen from figure 8(b). The normalized plot shows that the best overall performing class is the center class as represented by the number 0. The poorest performing class is the donut class represented by the number 1. The class was only predicted correctly 4% of the time, while also incorrectly being predicted to be the Loc class 86% of the time. In

total, the classification accuracy of the model is 77% which is good but possibly below the standard needed for the classification of wafer maps in a high risk semiconductor fab facility.

5.3 Sequential Model Implementation and Evaluation

Sequential models[16] are linear stacks of layers where the previous layer inputs to the next. This type of approach can be used for both classifier and declassifier models. The Keras sequential model is made up of three convolutional blocks. Each block contains a max pooling layer and a fully connected layer. The connected layer has 128 units on top of it and is activated by the Relu function. This is a convolutional neural network method for creating this type of system.

5.3.1 Sequential Model Implementation

The model was compiled using the Adam optimizer and the Sparse Categorical Cross entropy loss function. The metrics argument was passed to the compilation of the model to enable viewing of the metrics. Useful for plotting accuracy and loss for training and validation sets.

5.3.2 Sequential Model Evaluation

The model was first trained for 10 epochs on the wafer map data that was generated earlier. Next, the accuracy and loss are plotted on the training and validation sets. The plots as shown in Figure 9 highlight how the accuracy of the training and validation sets are off by a high margin. It also shows that the validation set has an accuracy of approximately 88%. The plot shows that the training accuracy increases at a linear rate over time. Since the validation set peaks at around 88%, it shows a high gap between the training and validation accuracy. This is a signal of over fitting. Although 88% is a high accuracy percentage, the overfitting signal means there is a possibility of increasing the accuracy. Data augmentation is used to generate more training data from the existing data. This will expose the model to random transformations to create more wafer map data. This in turn should enable the model to generalize the data to a higher extent, increasing the performance and accuracy of the model.

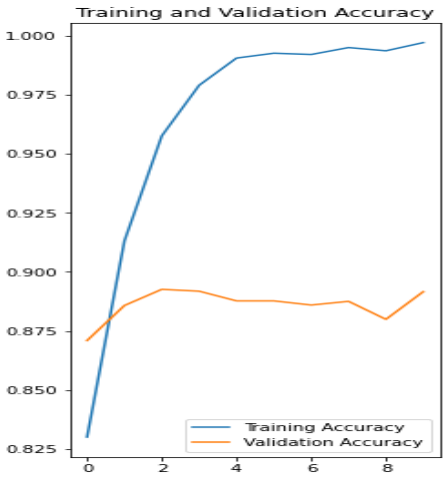

Figure 9(a) Accuracy

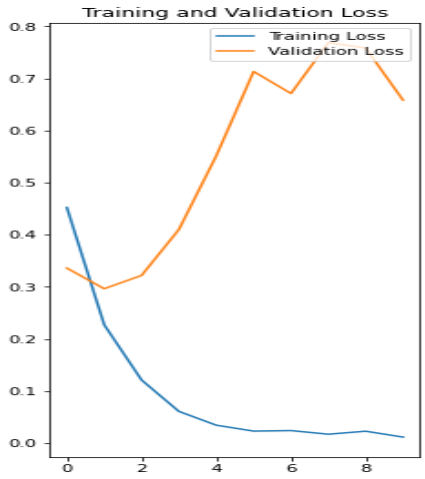

Figure 9(b) Accuracy

[16] https://machinelearningmastery.com/three-ways-to-build-machine-learning-models-in-keras/

As can be seen from Figure 10 plotting a sample of the augmented data shows the artificially generated wafer maps. These maps look much like the wafer maps obtained from the dataset. This augmented data is then added to the model by creating a new neural network before training the model again. After this, a dropout layer is applied. This randomly drops a number of output units from the layer when the model undergoes training. The dropout is approximately 20% to 40% of the output units selected randomly from the applied layer.

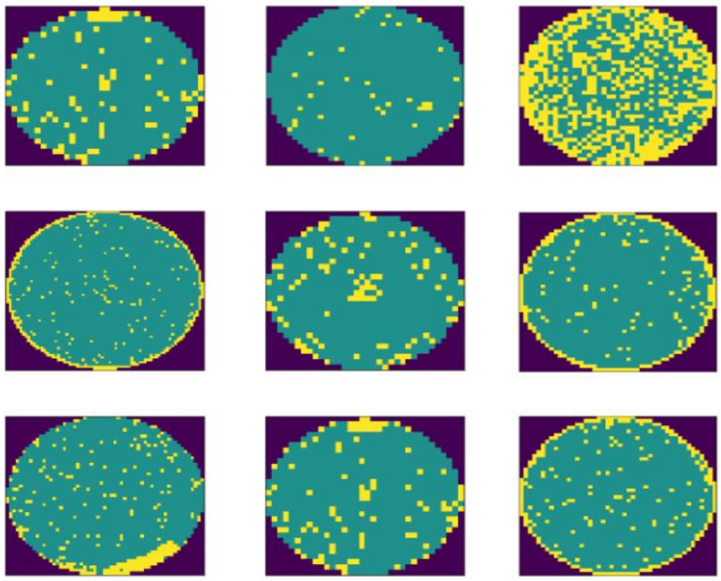

Figure 10: Augmented Wafer maps

The model is then trained with 15 epochs with the data augmentation and the dropout applied. The accuracy and validation sets are then plotted again which as per Figure 11 shows a significant decrease in overfitting. It can also be observed that validation and training sets are more closely aligned with a peak validation accuracy of approximately 92%. This is a 4% increase previous iteration of the model without data augmentation or use of a dropout layer.

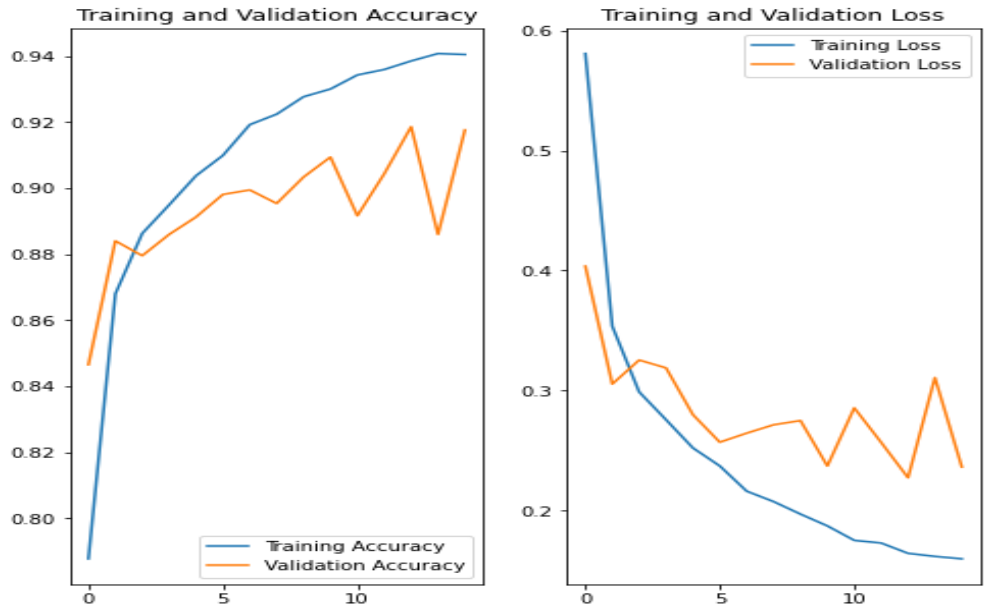

Figure 11: Accuracy and loss post augmentation

5.4 Discussion and Final Model Selection

Both the SVM and sequential model performed to a somewhat high accuracy but due to the use case of such a project, a high accuracy percentage is needed. When comparing the SVM model to the sequential neural network model its easy to see the drawbacks of SVM for the purpose of classification. SVM scored a respectable 77% percent accuracy. This would be acceptable for a non-business critical classification use case. But in the semiconductor industry, each second counts. An implantation of the SVM model would undoubtedly result in quite a few excursion misses which could result in production tool down time and potential wafer scraps. On the other hand, as seen from Table 4 the sequential model performed 15% better than the SVM model at 92%. This is a massive increase and is viable to be used in a semiconductor industry. With the constant feed of wafer maps within a facility, the accuracy of the model would increase over time. A business implementation of such a system could result in more wafers reaching end of line with a higher number of viable die for processors.

Model	Accuracy
SVM	77%
Sequential	92%
Delta	15%

Table 4: Model Comparisons

5.4.1 Final Model Selection

Due to the better performance of the sequential model, it was chosen as the final model for the project. Next step is to see how the model performs when it is faced with new data. As the wafer maps are 2 dimensional, it is easy to create new wafer maps in an image editing application for testing purposes. GIMP was used to create a new wafer map with a defect.

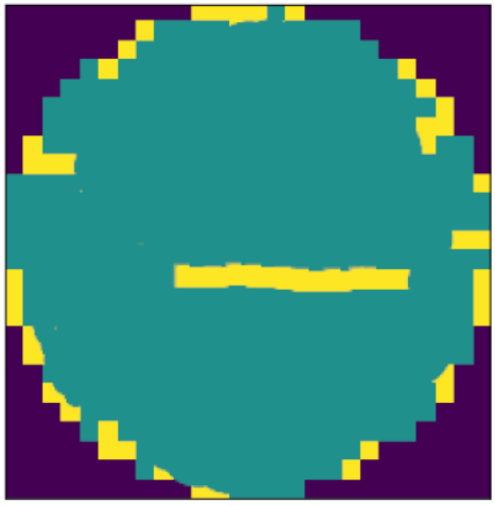

Figure 12 Manually created wafer map

As can be seen from Figure 12 above, a wafer map was created and edited to contain a scratch defect. The defect is located across the lower center of the map reaching to the right side of the wafer. As seen below, upon loading the image into the model for prediction. The model takes only 51 milliseconds to correctly classify the wafer map as a scratch with 96.58 percent accuracy as seen in Figure 13. This is a good accuracy and a good prediction speed.

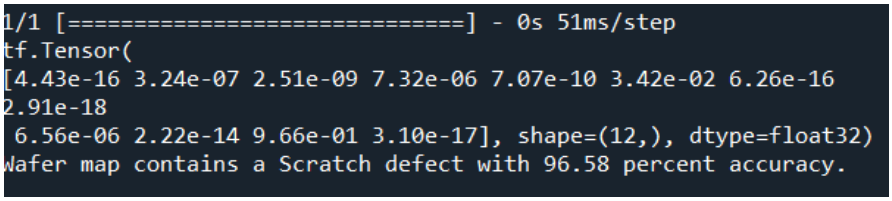

Figure 13: Scratch classification

5.5 Comparison of Existing Models

Author	Classificaiton Type	Method	Accuracy
(Muhammad Saqlain 2020)	Multiclass	CNN	96.20%
(Naigong Yu 2019)	Multiclass	CNN	93.13%.
(Loussaief & Abdelkrim, 2018)	Binary	BoF	39%
(Panigrahi, et al., 2018)	Multiclass	CNN	88%
(Shu-Min Li 2020)	Binary	CNN	89%
y (Yanh li 2021)	Multiclass	Mask R-CNN	87%
(Xiaoke Cao 2021)	Multiclass	YOLOv3 D-CNN	85.10%
(Jaegyeong Cha 2020)	Multiclass	Xception CNN	93.70%

Table 5: Existing model comparison

AS can be seen from Table 5, the machine learning methods are dominated by CNNs. The best performing model is by (Muhammad Saqlain 2020) at 96.20 with a CNN method. Followed by (Jaegyeong Cha 2020) at 93.70%. The worst performing model is the BOF method used for binary classification purposes by (Loussaief & Abdelkrim, 2018).

6 Proof of Concept UI Development

To properly test the use case of the model, it was decided to create a desktop app. As the model was already created and trained, it was then saved as a h5 file[17]. This would allow an external python file to access the model without having to go through the training process again. To make it a usable desktop application, a graphical user interface had to be created.

6.1 UI Design

Before creating the graphical user interface, a wireframe is created. The wireframe is a predesign step where all the needed elements are laid out and roughly put together. This gives an overview of how the UI will be used and what it will eventually contain as can be seen from Figure 14 below. The main screen contains 6 elements. 2 elements at the top of the window are labels. The first label prints the class prediction and the accuracy of the loaded wafer map. The second is label that comes visible when auto watch is enabled. At the center

[17] https://github.com/christianversloot/machine-learning-articles/blob/main/how-to-use-h5py-and-keras-to-train-with-data-from-hdf5-files.md

of the window are 2 elements. These are frames that will show the loaded wafer map and the salient version of that wafer map. At the bottom of the window are two buttons. The first button on the left allows the user to select a wafer map manually. The app will then do a prediction on this and classify. The button on the bottom right enables automated monitoring.

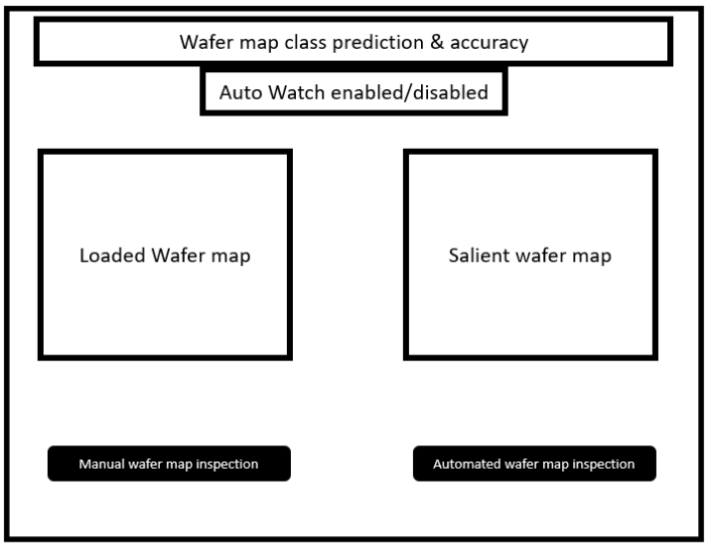

Figure 14: Wireframe

6.2 Wafer Map Detection App

Figure 15 below shows the graphical user interface of the wafer map detection app. The image on the left shows a scratched defect was detection with 99.13% accuracy through the manual inspection mode. The image on the right shows a scratch that was detected with 92.56% accuracy. When a user clicks the Automated wafer map inspection button, it will trigger the app to scan the new wafer map directly. The app will scan every few seconds looking for a new wafer map to classify. Once a defect is classified, the app will alert the user and show the defect on the screen. Which is a quick method of visualization for the end user.

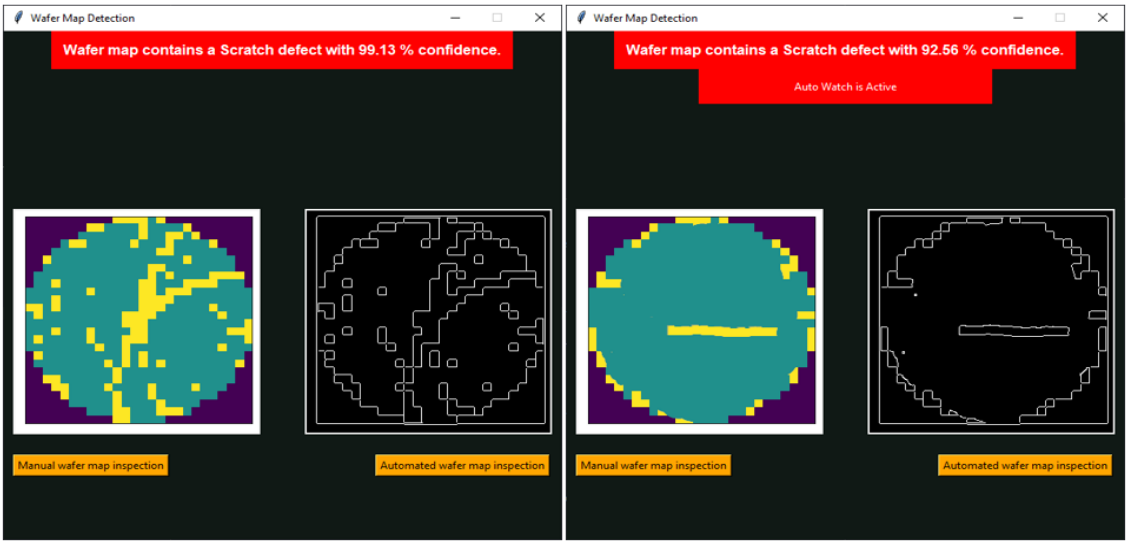

Figure 15: GUI for wafer detection app

7 Conclusion and Future Work

The automated detection of wafer map defects is something that has a real business need. As with what was seen in the previous section, the application could easily run on a command station or server. It has many benefits. Firstly, if implemented into a semiconductor facility, it would aid in the detection of down time causing defects. Resulting in a cost saving of potentially millions in reduced wafer scrappages and tool down events over time. Next, the app would be a huge work content reduction win for both engineers and technicians alike as it would free up time from the mundane and time centred reviewing of wafer map tasks. With more time, the app could be made into a suite of analytical tools directed towards the semiconductor environment. An SMTP function would prove useful where users could set up a condition to allow them to receive emails of when a defect occurs. At its base form, this project shows how analytics of a real-world problem for computer chip manufactures would solve it with the use of machine learning technology and integration. A follow up project could research the viability of other machine learning models. It could research the plausibility of deep learning neural networks for instant classification.

Acknowledgments

Thank you to my amazing fiancée for her support over the past two years. Who without, this work would not be possible. Thank you to the Intel Corporation for funding my studies and research. Thank you to Zahid Iqbal for his excellent guidance on this research paper. Dedicated to my three fantastic kids who will no doubt grow up to achieve great things.

8 References

Adly, F., Yoo, P. D., Muhaidat, S. & Al-Hammadi, Y., 2014. Machine-Learning-Based Identification of Defect Patterns in Semiconductor Wafer Maps: An Overview and Proposal. 28th International Parallel & Distributed Processing Symposium Workshops, *IEEE*, May, pp. 132-144.

Batool, U., Shapiai, M. I. & Tahir, M., 2021. A Systematic Review of Deep Learning for Silicon Wafer Defect Recognition. Imbalanced strategy for wafer defect classification using fully convolutional neural Network, *IEEE*, August, pp. 100 - 120.

Cao, X. & Zhang, F., 2020. Wafer Surface Defect Detection Based On Improved YOLOv3 Network. 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), *IEEE*, Novmeber, pp. 133-137.

Chen He, H. H. & Li, P., 2021. Applications for Machine Learning in Semiconductor Manufacturing and Test. Electron Devices Technology and Manufacturing Conference (EDTM) *IEEE*, April, pp. 10-20.

Chiang, K., 2006. Three DFM Challenges: Random Defects, Thickness. Asia Pacific Conference on Circuits and Systems, *IEEE*, April, pp. 1 - 3.

Daewoong An, H.-H. K. T. G. J. K. J.-G. B. ,. S.-S. K., 2009. A Semiconductor Yields Prediction Using Stepwise Support Vector Machine, A Semiconductor Yields Prediction. Using Stepwise Support Vector Machine, IEEE, pp. 17 - 20.

Felix Pistorius, D. G. F. E. E. S., 2020. Evaluation Matrix for Smart Machine-Learning Algorithm Choice. 1st International Conference on Big Data Analytics and Practices (IBDAP), *IEEE*, pp. 1-7.

Huang, P. W. & Chung, K.-J., 2016. Machine learning framework for image classification. 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), *IEEE*, April, pp. 1 - 4.

Huang, P. W. & Chung, K.-J., 2021. Task failure prediction for wafer-handling robotic arms by using various machine learning algorithms. *Measurement and Control, Volume 54, Issue 5-6*, May, pp. 701 - 710.

Ilham Rabhi, A. R. F. P. C. A., 2021. Out-Of-Control Detection In Semiconductor Manufacturing using One-Class Support Vector Machines. 17th International Conference on Automation Science and Engineering (CASE), *IEEE*, pp. 23-27.

Jaegyeong Cha, S. O., 2020. A Defect Detection Model for Imbalanced Wafer Image Data Using CAE and Xception. International Conference on Intelligent Data Science Technologies and Applications (IDSTA), *IEEE*, November, pp. 23-33.

Li, S.-M., Liao, P. Y.-Y. & Wang, S.-J., 2020. Potential Wafermap Scratch Defect Pattern Recognition with Machine Learning Techniques, 25th European Test Symposium (ETS), *IEEE*, April, pp. 20-23.

li, Y. & Wang, J., 2021. A Defect Detection Method Based on Improved Mask R-CNN for Wafer Maps, International Conference on Computer Network, Electronic and Automation (ICCNEA), *IEEE*, Novmember, pp. 133-137.

Loussaief, {. & Abdelkrim, A., 2018. Deep learning vs. bag of features in machine learning for image classification, International Conference on Advanced Systems and Electric Technologies (IC_ASET), *IEEE*, pp. 1 - 5.

Moriya, T., 2021. Machine Learning Approaches Optimizing Semiconductor Manufacturing Processes, 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), *IEEE*, April, pp. 20-32.

Panigrahi, S., Nanda, A. & Swarnkar, T., 2018. Deep Learning Approach for Image Classification, 2nd International Conference on Data Science and Business Analytics, *IEEE*, April, pp. 1 - 6.

RongAn Chen, Z. & Qu, J. Q., 2014. Enhanced Bag-of-Features model for image classification, Workshop on Advanced Research and Technology in Industry Applications (WARTIA), *IEEE*, Devemeber, pp. 1 - 4.

Saiz, F. A., Serrano, I., Barandiarán, I. & Sánche, J. R., 2021. A Robust and Fast Deep Learning-Based Method for Defect Classification in Steel Surfaces, International Conference on Intelligent Systems (IS), *IEEE*, May, pp. 1 - 6.

Saqlain, M., Abbas, Q. & Lee, J. Y., 2020. A Deep Convolutional Neural Network for Wafer Defect Identification on an Imbalanced Dataset in Semiconductor Manufacturing Processes, Transactions on Semiconductor Manufacturing, *IEEE*, May, pp. 436-444.

Shyava Tripathi, R. K., 2019. Image Classification using small Convolutional Neural Network, 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), *IEEE*, July, pp. 1-5.

Winoto, A. S., Kristianus, M. & Premachandra, C., 2020. Small and Slim Deep Convolutional Neural Network for Mobile Device, IEEE Access (Volume: 8) , *IEEE*, May, pp. 1-13.

Wu, M.-C., Chiou, C.-W. & Hsu, H.-M., 2002. Scrap rules for small lots in wafer fabrication, Semiconductor Manufacturing Technology Workshop, *IEEE*, Decemeber, pp. 187 -190.

Yu, N., Q. X. & Wang, H., 2019. Wafer Defect Pattern Recognition and Analysis Based on Convolutional Neural Network, Transactions on Semiconductor Manufacturing (Volume: 32, Issue: 4, November 2019),*IEEE*, May, pp. 566 - 573.