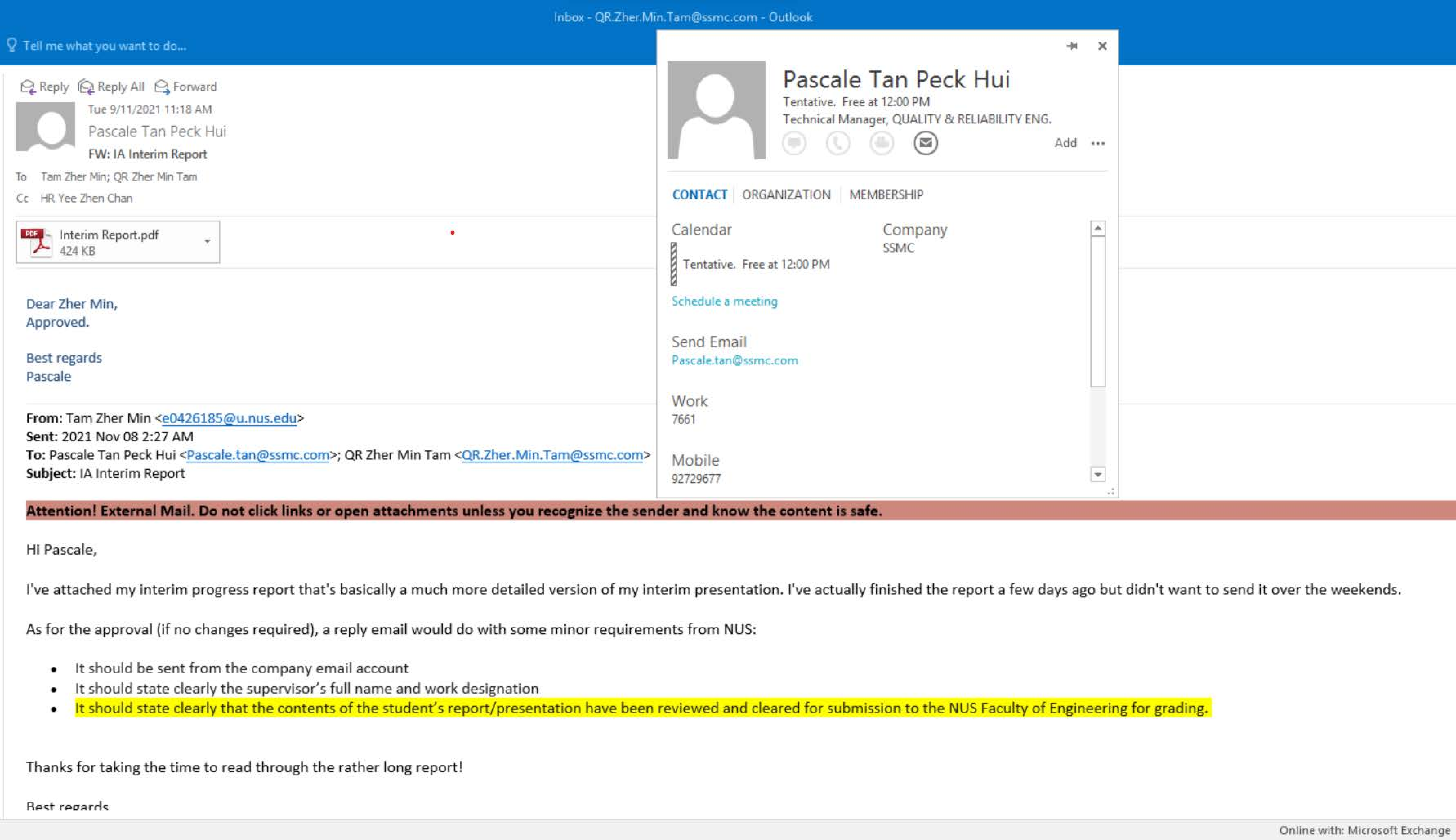
Inbox - QR.Zher.Min.Tam@ssmc.com - Outlook
Tell me what you want to do...
Reply
Reply All
Forward
Add
CONTACT
ORGANIZATION
MEMBERSHIP
Schedule a meeting
Online with: Microsoft Exchange

Tue 9/11/2021 11:18 AM

Pascale Tan Peck Hui

FW: IA Interim Report

To Tam Zher Min; QR Zher Min Tam

Cc HR Yee Zhen Chan

Interim Report.pdf
424 KB

Pascale Tan Peck Hui

Tentative. Free at 12:00 PM

Technical Manager, QUALITY & RELIABILITY ENG.

Calendar

Tentative. Free at 12:00 PM

Company

SSMC

Send Email

Pascale.tan@ssmc.com

Work

7661

Mobile

92729677

Dear Zher Min,
Approved.

Best regards
Pascale

From: Tam Zher Min <e0426185@u.nus.edu>
Sent: 2021 Nov 08 2:27 AM
To: Pascale Tan Peck Hui <Pascale.tan@ssmc.com>; QR Zher Min Tam <QR.Zher.Min.Tam@ssmc.com>
Subject: IA Interim Report

Attention! External Mail. Do not click links or open attachments unless you recognize the sender and know the content is safe.

Hi Pascale,

I've attached my interim progress report that's basically a much more detailed version of my interim presentation. I've actually finished the report a few days ago but didn't want to send it over the weekends.

As for the approval (if no changes required), a reply email would do with some minor requirements from NUS:

- It should be sent from the company email account
- It should state clearly the supervisor's full name and work designation
- It should state clearly that the contents of the student's report/presentation have been reviewed and cleared for submission to the NUS Faculty of Engineering for grading.

Thanks for taking the time to read through the rather long report!

Best regards

EG3611A Interim Report

Name: Tam Zher Min

Company: Systems on Silicon Manufacturing Company (SSMC)

Supervisor: Ms. Pascale Tan

Internship Period: Aug 23 2021 — Jan 7 2022

Acknowledgements

I would like to express my deepest appreciation to SSMC for providing me this internship opportunity alongside my supervisor, Ms. Pascale, as well as my colleagues in the QRA department, Mr. Wen De and Ms. Nedlin, for their continued support throughout my internship journey. Special gratitude to my supervisor for vetting my presentations and reports as well as appreciating my efforts and trusting my abilities to advance the project independently.

Not forgetting my mentors from NUS, Professor Vincent, who attended my interim presentation and provided valuable comments as well as my CELC tutor, Dr. Lira, for the constant feedback and detailed deliverable breakdowns for this industrial attachment module.

Table of Contents

1.0 Introduction

1.1 Company Overview

Systems on Silicon Manufacturing Company is a Singaporean semiconductor fabrication company incorporated in 1999 and is a joint venture between two other large semiconductor companies, NXP and TSMC, which are based in Netherlands and Taiwan respectively.

Specializing in producing semiconductor chips, they range from 0.25-micron to 0.14-micron and are mostly used for logic units, embedded flash memory, mixed signal, and radio-frequency applications [1]. Apart from manufacturing, SSMC also provides consultancy and support to customers' business processes.

In recent years, there has been a global chip shortage. This was largely fueled by the pandemic, which dealt huge blows to the supply chain due to ports and factories shutting down, while simultaneously, demand for digital devices that often require several of these tiny chips to operate one device, exploded, resulting in massive backlogs [2]. From electric toothbrushes to phones and computers to cars, many of these applications are waiting for this final component before shipping out. Inevitably, some of them are going to be cancelled or delayed [2]. Opinions on when this shortage will end vary, with estimates projecting into the end of 2023. Although demand may slow down as pandemic restrictions ease up, it will still take time before supply catches up with demand and equilibrium is reached [2].

Naturally, SSMC is poised as a key player along the semiconductor supply chain. Having recently shipped their 10-millionth wafer in 2020 [3], SSMC continues to surpass new milestones and positions itself at the forefront in this industry.

1.2 Objectives & Motivations

Whilst searching for an internship, I had clear goals in the kind of work I would want to do. Although formally, my course of study is in Electrical Engineering, I knew I had wanted a more software-related role, ranging from software development to machine learning or data science. Prior to the internship, an introductory module on machine learning piqued my interest in this field, which spurred me into choosing companies with machine learning job scopes.

Eventually, I found out that SSMC has a project that would be well suited for a machine learning solution, even though it was not explicitly mentioned in the job requirements. Despite my rather severe lack of machine learning skills initially, I was confident that I would be able to learn on the job and tackle the project given my strong sense of self-directness and related programming capabilities. My motivation for this internship hence morphed into a strong desire to hone my coding and machine learning abilities along with any data science and software development skills that will all undoubtedly prove critical in my future endeavors.

2.0 Project Requirements

Tagged under SSMC's Quality and Reliability Assurance (QRA) Department, which handles the quality assurance of completed wafers towards the end of the entire fabrication process, I was tasked to "set up or create an Automatic Defect Classification (ADC) system". Although this project seemed simple at first glance, delving deeper into the requirements unveiled a rather large-scale project that consists of multiple smaller moving parts working together in tandem to constitute and fulfill the "system" part of the task.

The impetus for this project is that completed wafers sometimes experience defects along the wafer fabrication process, which lower yield and directly impact revenue if critical defects are found that require the wafers to be scrapped. Hence, specialized machines use light reflected off the wafer surface to detect variations and anomalies, allowing the machines to flag out these tiny defects.

However, although these machines are great at autonomously finding these microscopic anomalies, they not only are unable to differentiate between false positives from true positives, but they are also unable to classify the exact defect types without the help of human inspections. Reliance on manual inspections can be costly in time and money. Hence, this project serves to reduce this need, which will also help in scaling up the number of inspections that can be done in a given time frame.

Because defects can occur either on the frontside, backside or edges of a thin slice of wafer, my project scope is slightly narrowed to only the edges and backside for a minor reduction in difficulty. However, since the number of defect types differ for the edges and backside, I decided to allocate the first half of the internship into getting the wafer edges to be correctly classified as it was slightly more straightforward compared to the backside. After the code and foundations have been established for wafer edge classification, I will then be able to translate into solving for the backside easily.

Ultimately, my main objective would be to create and deploy trained machine learning models that are able to correctly differentiate the wafer scans from the aforementioned specialized machines. With regards to the wafer edge problem, my model must identify if an image contains signs of chipping or not, as seen below. This entails leveraging open-source machine learning libraries for automatic defect classification such that follow-up actions can be meted out for the particular wafer or wafer lot.

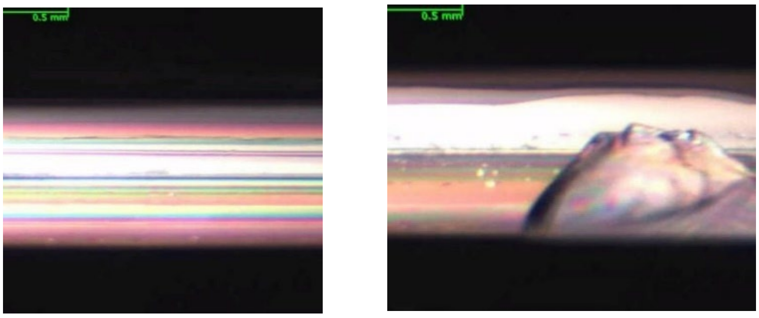

Figure 1: No-Chipping (Left), Chipping (Right)

3.0 Workflow & Progress Analysis

As outlined in the workplan, which can be referred to in Appendix A, the process which I have drafted up follows the flow of 4 major steps: data collection, data pre-processing, data analysis and machine learning, and finally, reporting and disposition. Although parts of the workplan will be repeated, further details will be included, covering the challenges and constraints encountered, solutions identified, and the contributions made for each of the 4 steps.

3.1 Data Collection

3.1.1 Data Generation

For every wafer scanned by the machine, tens to hundreds of images are generated depending on how many anomalies were found. Multiplied by 25, the number of wafers in one wafer lot, before multiplying again by the number of lots scanned within a certain time frame, the number of images that need to be analyzed is immense. This is in part due to the sensitivity of the machines. Since the rulesets for the machines are unoptimized, they are tuned to be highly sensitive to avoid missing out on any defects.

This poses many problems, mainly that most of the images are false positives and hence, needed to be sieved out, alongside the large amounts of memory these images occupy in the database. Thus, in the initial part of my internship, I was finding ways to mitigate this issue. As I understood the wafer fabrication process and the machines better, I performed an in-depth analysis on the memory consumption of the wafer scans, which produces 3 types of scans – edge top, edge normal and backside scans. What I discovered was that by disabling edge top scans, the memory usage can be cut by more than half while the anomalies are still being detected by the edge normal and backside scans (highlighted green) since the wafer top scans are the biggest culprit in generating false positives (highlighted red).

Table 1: 3-Month Memory Usage of Wafer Scans (TOP scans take up twice the memory of BACK+NORMAL scans)

Memory (MB)	BACK	NORMAL	TOP	**Total**
JUL	61	7	167	235
AUG	179	19	452	650
SEPT	110	5	85	200
Total	350	31	704	1085

Although this is not the best solution, it serves as a temporary one to enable more wafers to be scanned. Optimally, one would try and optimize the rulesets to generate less noise yet ensuring defects are still found, but this would be time consuming and is not known if this method will even work. Hence, I decided to focus more on utilizing machine learning to classify the scanned images instead of tweaking the machines' algorithms.

3.1.2 Data Pipelining

When default work-from-home (WFH) arrangements struck, I needed a way to transfer the wafer scans from the company desktop to my personal laptop. This proved to be a major obstacle due to the company's strict policies which prevented interns from gaining barely any access. With no emails or communication channels out to external parties, no external storage devices, and no remote access to the desktop, I was stumped for a long time. Thankfully, virtual desktop access was provided, which prompted me to resort to screenshotting all the images from my laptop.

With thousands of images to be screenshotted, there was no way a manual solution would have sufficed. Thus, I leveraged my knowledge in Python scripting in order to write an algorithm that will programmatically capture my screen. With all the necessary data at hand, I am finally able to start training a machine learning model that will learn from these images.

3.2 Data Pre-Processing

This step pertains to the processing of all the image data, which includes sorting and labelling, image normalization, dataset balancing and data augmentations in order to prime the data for a supervised machine learning model. This step is crucial because training the model requires the data to be cleaned and processed or else the model is likely either going to fail to run completely or produce highly inaccurate results. All these tested me on my data science and data engineering skills which I have picked up outside of my curriculum through my previous internship as well as my own research and self-learning.

However, novel problems I faced lied in the balancing of the dataset. Even though I managed to extract over 2000 wafer edge scans, there were only 30 images that showed signs of chipping, with the rest being images of normal wafer edges. This poses a major class imbalance problem and if all the images were fed to the model, the model will turn out to be heavily biased towards the majority class, which means that it will simply predict no-chipping almost all the time. As a simple example, if you have 100 images, and 99 of them are of class A and 1 of class B, even if you have a model that predicts class A all the time, the model will still be 99% accurate, which indeed sounds good on paper. However, the purpose of machine learning models is often to predict the minority class instead. Hence, this class imbalance problem must be addressed before feeding into the model.

Through my own research, I managed to find 7 ways to address this issue, each with its own pros and cons in terms of accuracy and speed, which includes traditional methods such as undersampling, oversampling and using a balanced class ratio, to more exotic methods such as synthetically generating data or chaining multiple models to obtain a majority vote. Nonetheless, after much experimentation, I found out that oversampling the minority class works well enough for my needs.

Overall, with sufficient understanding and testing, I was able to chain together the necessary data processing steps into a pipeline that will then feed into the model for training and evaluation.

3.3 Data Analysis & Machine Learning

This next step in my workflow is the most time-consuming. From the further research and self-learning required to find good model architectures and coding them for my problem statement, to the time needed to train the model and experiment with different parameters, these components entail a lot of patience and time.

Because I am working on image data, I chose convolutional neural networks (CNNs) as my solution as they are well suited for computer vision tasks especially when using pretrained models. For simplicity's sake, I will only briefly explain one of the model backbones used, which is called VGG16. Using a pretrained model like VGG16 enables me to train a model that does not require as many image data as a custom model because these models have already previously been trained on millions of unrelated images such as animals and vehicles. By removing a small part of the model, shown by the green blocks in the figure below, I can customize this model created by other researchers to fit it to my goal of predicting either a chipping or no-chipping image.

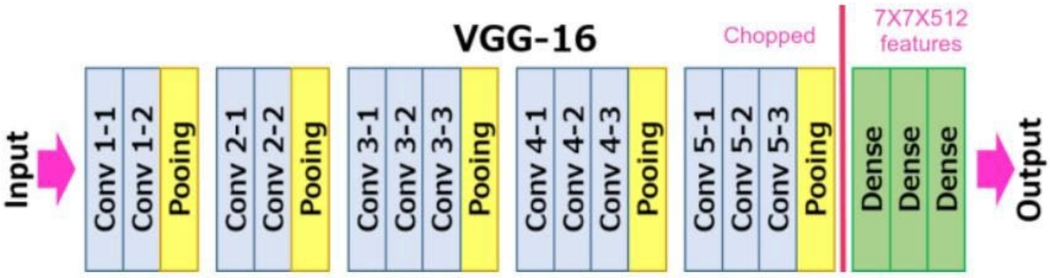

Figure 2: VGG16 Pretrained Model Architecture [4]

Since there was a need to customize this VGG16 model, I could not simply use cookie-cutter code found online. Rather, this is where my theoretical understanding of machine learning and CNNs come into play, which I have gained both in my previous introductory machine learning module as well as external competitions that I have joined. With that, coupled with extensive Googling and YouTubing, I was able to concoct the necessary code fit for my problem statement. Although the initial stage of squashing the bugs in my code was tedious and in rather uncharted waters, my perseverance paid off. Naturally, it also became easier as time passes. As it stands, there were many more model architectures and hyperparameters that I have tested but a few of the models stood out, to which the accuracies can be observed from the table below.

Table 2: Performant Trained VGG16 Models (correct / total = accuracy)

	test_correct	test_total	test_accuracy
vgg16_30-sep	4700	4793	98.1%
vgg16_12-oct	4530	4577	99.0%
vgg16_13-oct	4571	4577	99.9%

3.4 Reporting & Disposition

On top of the raw accuracies recorded in the above table, the main evaluation metric and visualization I used was the confusion matrix. A confusion matrix gives a quick overview on the specific class accuracies of a model's predictions by including both the actual and predicted values as illustrated by the figure below. Understandably, one would try to optimize for TNs and TPs, which are the true negatives and true positives respectively, since they represent correct predictions. But between FNs and FPs, the false negatives and false positives, FNs hold a slightly higher importance in my problem statement. This is because an FN indicates that the model incorrectly predicted a negative class, which translates to no signs of chipping, when in reality, it should have been flagged as a chipping image. This means that certain defective wafers could go undetected if the percentage of FNs is high. Hence, even in such a simple 2-by-2 table, one can easily discern whether a model is going to be detrimental in real-world applications or not.

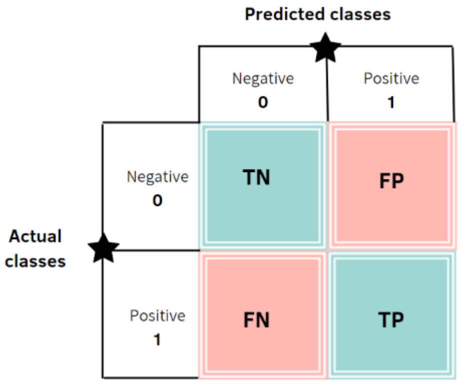

Figure 3: A Sample Confusion Matrix [5]

Apart from visualizations such as confusion matrices, reporting in terms of documentation plays a part as well. Because the long lines of code that I have written are not the friendliest to those new to machine learning concepts, sufficient explanations coupled with relevant figures and formatting should be considered when presenting my findings to my supervisor and colleagues. This is where formatting languages such as Markdown and LaTeX come in handy, which I have picked up and practiced extensively as I learnt to generate readable and understandable reports and documentation.

With that said, documentation is an ongoing process within my workflow and ties into the next section of this report, which pertains to the disposition and deployment of my models and coincide with some of the ongoing work and upcoming plans for the second half of this internship.

4.0 Current & Future Plans

Producing a performant model is not enough. If my model and code run in a self-contained bubble and environment that only I can understand and operate, then simply put, they serve no purpose to the company and provides zero business value apart from existing as a proof-of-concept at best.

Hence, I have been researching more on MLOps, or machine learning operations, which governs the architecting of a closed loop system of experimentation, deployment, and maintenance; at the same time, developing a minimum viable product (MVP) on how a deployed model would function. My MVP currently is a webapp with a user interface written in a Python library called Streamlit and hosted on Streamlit's server for free as a proof-of-concept. My webapp allows users to upload multiple images for on-the-spot predictions through some of my trained models.

However, two main concerns are present for this MVP. Firstly, it is not optimized for performance. Because the webapp is hosted on Streamlit's free servers, there are naturally memory limitations set by Streamlit, unlike their enterprise options. With regards to speed, there are some optimizations that can be done from my end such as changing the model's backbone to a lighter and more efficient model like Mobile Net V2, an improvement over the heavier VGG16 mentioned earlier. Unfortunately, because Streamlit's servers are not designed for deep learning applications such as mine, they do not possess accelerated hardware solutions such as graphical processing units (GPUs) that are often critical in further decimating the time required for model training and inference.

Secondly, there are currently no integration with SSMC's internal infrastructure. This means that there are no connections with the company's existing software or systems. This is required for a seamless flow of data, starting with SSMC's machine wafer scans, which are stored in the company's database, before I extract these images into my prediction model and bin the predicted images in the correct categories for wafer disposition and further analysis.

These are concerns that I am currently addressing and for any solution that I cannot currently execute, I will be drafting up comprehensive comparisons and proposals to ensure that implementations in the future be implemented from a well-understood and well-considered standpoint.

5.0 Conclusion

Overall, my time so far with SSMC has been fruitful. Although being stuck at a particular problem is never enjoyable, I have grown to appreciate the challenge of digging for the answer because the feeling of making it work makes the struggle worth it. Moreover, being able to create novel solutions and propose ideas that have yet to be implemented spurs me on to try and provide real business value for SSMC even though I might just be an intern.

Needless to say, in terms of application to future projects and career paths, these are all extremely useful skills and knowledge that are highly transferrable especially in the realm of software engineering. Thankfully, I have a rather clear goal in mind for the kind of jobs that I would go for down the road, which helps me to stay motivated as I am confident that these are experiences and work ethics that potential employers would be looking out for.

References

[1] “SSMC is Singapore Quality Award winner in 2005 citation,” 2005. [Online]. Available: https://www.ssmc.com/ssmcnews/2005/ssmc-is-singapore-quality-award-winner-in-2005-citation.pdf. [Accessed: 04-Nov-2021].

[2] F. Shira, “Understanding the global chip shortage, a big crisis involving tiny components,” *Popular Science*, 12-Oct-2021. [Online]. Available: https://www.popsci.com/technology/global-chip-shortage/. [Accessed: 04-Nov-2021].

[3] “About SSMC,” *SSMC*. [Online]. Available: https://www.ssmc.com/AboutUs/AboutSSMC. [Accessed: 04-Nov-2021].

[4] Meghabansal, “Face recognition using transfer learning and VGG16,” *LaptrinhX*, 26-Aug-2020. [Online]. Available: https://laptrinhx.com/face-recognition-using-transfer-learning-and-vgg16-3833175636/. [Accessed: 04-Nov-2021].

[5] A. Eugenia, “How to evaluate you model using the confusion matrix,” *Towards AI - The World's Leading AI and Technology Publication*, 26-Feb-2021. [Online]. Available: https://towardsai.net/p/data-science/how-to-evaluate-you-model-using-the-confusion-matrix. [Accessed: 04-Nov-2021].

Appendix A: Internship Workplan

1.0 Objectives

As an intern under the QRA (Quality and Reliability Assurance) department at Systems on Silicon Manufacturing Company (SSMC), my main objective is to set up an Automated Defect Classification (ADC) system in order to reduce the number of false positives when detecting for anomalies on wafer edges and backsides as well as to leverage on machine learning and computer vision algorithms to correctly identify critical defects for wafer disposition.

2.0 Primary Work Duties & Deliverables

Primarily, my work duties require me to independently manage the data that I need, in the form of scanned wafer images, from the beginning until the end. This can be summarised into four major steps: data collection, data pre-processing, data analysis and machine learning, and finally, reporting and disposition. My final deliverables will thus include a set of automation scripts and a trained machine learning model fit for accurate defect predictions in order to reduce reliance on manual visual inspection, which is the current practice at SSMC for wafer edges and backside defect detection.

2.1 Data Collection

Wafers prepared to be shipped to customers will go through scans using specific machines in order to detect any critical defects that will result in lot failures. Currently, scans of wafer edges and backside produce significant noise and false positives due to high sensitivity in the unoptimized set of rules that the machines use. Hence, I will be required to tune the ruleset to not only alleviate this phenomenon, but also to ensure the memory footprint is lowered when the images are exported.

2.2 Data Pre-Processing

After the necessary images with minimal noise are exported and extracted locally, I will need to manually filter out actual defect images into a separate area so as to prime for a supervised machine learning model. Furthermore, I will be using Python to read these images and perform image processing techniques such as greyscale, rescaling and resizing to normalize the images for easier training.

2.3 Data Analysis & Machine Learning

The normalized images will then be analysed to account for factors that could affect model performance such as class imbalances. Next, research and systematic techniques will be applied to setup and train a machine learning model such as convolutional neural networks that are well suited for computer vision tasks. These different models will then be further assessed and cross-validated in order to extract the best performing model for deployment.

2.4 Reporting & Disposition

Relevant information on the accuracy and performance of the model such as confusion matrices, pareto charts and other data visualisation tools such as Matplotlib and Tableau will then be utilized to give clear and concise summaries for the company to understand the results.

3.0 Intended Learning Outcomes

Apart from understanding the wafer production process, it is highly crucial for me to learn the software and tools in use such that I will be able to effectively extract the information I need. This includes understanding and entering the clean room, where the wafer scanning machines reside. Moreover, because the software that the machines use are proprietary tools, information about its utilization will not be able to be found online and hence, require heavy reliance on their provided brief documentation and hands-on testing of the equipment.

Outside of the processes in data extraction, I will have to effectively apply data management and machine learning knowledge gathered during my curriculum and outside of it. This is because there is no guidance provided at all. Whether this project succeeds or not hinges on my ability to be independent in my learning and research skills. Hence, I believe a critical learning outcome would be resourcefulness, or the ability to extrapolate my limited knowledge to complete the tasks handed to me.

Lastly, apart from creating an accurate model, the ability to report my findings clearly in a non-technical way will be required. This will likely demand me to learn about business intelligence tools such as Tableau in order to output visually appealing charts that are understandable even for a general audience.