WAREHOUSE-BASED AUTOMATED DETECTION AND CLASSIFICATION OF SCREEN DEFECTS IN RETURNED PHONES FOR MOBILE CARRIER NOTIFICATION

BHOGASENA REDDY KALAKATA

Final Thesis Report

FEBRUARY 2024

DEDICATION

This thesis is dedicated to my parents, whose love and support have always been the driving factor behind all my accomplishments. Your unwavering faith in my abilities has been my inspiration throughout this journey.

To my beloved spouse, whose patience, understanding, and unwavering love has been my rock during the most challenging parts of this journey. Your belief in me has been my source of strength and resilience.

To my children, my precious son and daughter, whose understanding and cooperation during my study sessions have been incredibly helpful. Your reminders to attend class and your respect for my need for quiet during those times have been invaluable.

To my mentor, who has guided me through this academic journey with wisdom, patience, and kindness. Your expertise and guidance have been invaluable to this project and to my growth as a researcher.

Thank you all for your support, encouragement, and endless love. This achievement would not have been possible without you.

ACKNOWLEDGEMENT

I am profoundly grateful to a number of individuals whose support and assistance have been invaluable throughout the course of this thesis.

Firstly, I would like to express my deepest gratitude to my thesis advisor, Dr. Deanne Larson, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate your vast knowledge and skills in many areas, and your assistance in navigating the complex paths of research.

I am heartily thankful to my family: my parents and to my spouse, who has been my constant source of inspiration throughout this journey. To my children, who have been understanding and patient throughout this process, thank you for reminding me of my schedule and maintaining quiet during my study sessions.

Special thanks to my Upgrad buddy and technical support team, for the constant support, invaluable suggestions, and encouragement when it was most needed. Your companionship during this journey made the process less daunting and more enjoyable.

I am also thankful for the resources provided by Liverpool John Moores University which played a significant role in the completion of this study.

Finally, I wish to express my love and gratitude to all those who provided me with the possibility to complete this report, whose support and encouragement were valuable to me.

Thank you all.

Bhogasena Reddy Kalakata

ABSTRACT

This research proposal addresses the imperative of optimizing reverse logistics for warehouses catering to mobile carrier customers/vendors. through the development of an advanced mobile screen damage detection and classification system. Reverse logistics, encompassing the intricate processes of managing returned mobile phones, faces challenges amplified by the exponential growth of e-commerce and evolving environmental regulations. Specifically, the prevalent issue of glass damage incurs significant costs and delays within the supply chain, necessitating innovative solutions.

My primary objective is to design and implement a scalable, robust, and diversified system capable of identifying and classifying three specific mobile screen damage types: oil, scratch, and stain. To achieve this, the research adopts a comprehensive workflow spanning dataset acquisition, preprocessing, transformation/augmentation, modeling, evaluation, and result interpretation. Transfer learning, data augmentation, and regularization/dropout methods constitute pivotal components of my methodology, enhancing the system's adaptability and performance.

In the context of reverse logistics, the proposed system emerges as a transformative solution to the challenges associated with manual inspection and processing of returned items. By focusing on specific damage types, system not only addresses immediate industry concerns but also aligns with the broader global push towards sustainable supply chains. The integration of advanced machine learning techniques, namely transfer learning and data augmentation, contributes to the efficiency of reverse logistics practices, surpassing traditional manual methods.

In summary, this research underscores the significance of adopting advanced technological solutions to enhance the efficiency of reverse logistics in the mobile industry. The proposed system's scalability, adaptability, and focus on specific damage types position it as a model for sustainable and efficient supply chain practices. Beyond the mobile industry, the potential impact of this research extends to diverse sectors engaged in reverse logistics, contributing to the evolution of artificial intelligence applications in supply chain management.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGV: Automated Guided Vehicle

AI: Artificial Intelligence

AS/RS: Automated Storage and Retrieval Systems

CNN: Convolutional Neural Networks

COVID-19: Coronavirus disease 2019

CPU: Central Processing Unit

DCGAN: Deep Convolutional Generative Adversarial Networks

DL: Deep Learning

ECG: Electrocardiogram

FCM: Fuzzy C-Means

FPN: Feature Pyramid Network

GAN: Generative Adversarial Networks

GPU: Graphics Processing Unit

R-CNN: Region-Based Convolutional Neural Networks

HRI: Human-Robot Interaction

ML: Machine Learning

MPCG: Mobile Phone Cover Glass

PIVD: Product Image-based Vulnerability Detection

SSD: Single Shot Detection

TPU: Tensor Processing Unit

VGG: Visual Geometry Group

VOC: Visual Object Classes

YOLO: You Only Look Once

ReLU: Rectified Linear Unit

CHAPTER 1: INTRODUCTION

1.1 Background of the Study

With the rapid expansion of e-commerce and an increased focus on environmental sustainability, consumer behavior has undergone a paradigm shift (Mangiaracina et al., 2015). This shift is prompting businesses to reassess their strategies in reverse logistics (Janse et al., 2010).

This shift is particularly evident in warehouses catering to mobile carrier customers/vendors, where the effective management of returned mobile phones poses intricate challenges (Bravo et al., 2022). The traditional approach to reverse logistics involves manual inspections, repairs, and recycling procedures within warehouses, leading to resource-intensive processes that occupy substantial space and require a significant workforce (Alarcón et al., 2021; Genchev et al., 2011; Schlüter et al., 2021).

Amid these challenges, the identification and mitigation of glass damage emerge as pivotal concerns for both mobile carrier customers and vendors. Glass damage not only incurs substantial costs and delays in the supply chain but also aligns with the evolving demands of an environmentally conscious market. Consequently, a critical reassessment of operational practices is imperative to enhance overall efficiency and reduce operational costs.

In response to these challenges, the adoption of automated strategies, particularly leveraging machine learning algorithms, has gained prominence. These automated approaches demonstrate superior cost and time efficiency, offering transformative solutions for the identification of glass damage in warehouse settings. As the demand for such strategies continues to grow, there is a pressing need for advanced systems capable of not only identifying but also classifying specific types of damage. This research addresses this critical gap by proposing the development of a mobile screen damage detection and classification system. The aim is to revolutionize current warehouse procedures, contributing to a more sustainable and efficient supply chain within the mobile carrier industry.

The overarching goal of this research is to develop an advanced mobile screen damage detection and classification system, optimizing reverse logistics processes for warehouses catering to mobile carrier customers/vendors.

1.2 Problem Statement

The rapid advancement of e-commerce has led to an increase in returned mobile phones, creating unique challenges in warehouse management and reverse logistics. Inspecting returned devices for potential repairs or recycling is traditionally a manual, time-consuming, and resource-intensive process.

A specific challenge lies in the accurate detection and classification of screen damages, such as oil, scratch, and stain defects. Manual processes may not consistently identify and categorize these damages, which could lead to inefficiencies and increased costs.

Recognizing these challenges, the aim of this research is to develop a scalable, robust, and diversified mobile screen damage detection and classification system. This system is intended to improve upon the traditional methods, enhancing the efficiency and sustainability of warehouse procedures and the broader supply chain.

By leveraging machine learning algorithms, the proposed system aims to address issues like overfitting and optimize performance with high-resolution datasets. The goal is to provide a more advanced solution for damage detection and classification, contributing to the overall efficiency of reverse logistics processes.

1.3 Aim and Objectives

The aim of this research is to develop a scalable, robust, and diversified mobile screen damage detection and classification system, specifically focusing on the detection and classification of oil, scratch, and stain defects.

The research objectives are formulated based on the aim of the study, which are as follows:

- To design and implement a damage detection algorithm capable of identifying and classifying three specific types of mobile screen damages: oil, scratch, and stain.
- To apply transfer learning techniques to improve the accuracy of these specific damage detections.
- To utilize data augmentation methods to increase the diversity of these specific screen damage types the system can identify and classify.
- To incorporate a combination of regularization and dropout methods to enhance the system's robustness against overfitting.
- To evaluate the effectiveness of the proposed system through rigorous testing, particularly focusing on its performance with a high-resolution dataset.
- To analyze and document the findings, with a specific focus on the types of defects present in the dataset.

1.4 Research Questions

1. How can transfer learning be applied to improve the accuracy of mobile screen defect detection and classification?
2. In what ways can data augmentation enhance the diversity of damage types that the system can detect?
3. How can a combination of regularization and dropout methods be implemented to make the system robust to overfitting?

1.5 Scope of the Study

1.5.1 In Scope

This study is designed to focus on the development and implementation of a mobile screen damage detection and classification system tailored for reverse logistics operations in the warehouses. The primary components within the scope of this research include the design and implementation of a damage detection algorithm capable of identifying and classifying three specific types of mobile screen damages: oil, scratch, and stain. The study will leverage transfer learning techniques to enhance the accuracy of these specific damage detections. Furthermore, data augmentation methods will be employed to increase the diversity of the identified screen damage types, and a combination of regularization and dropout methods will be incorporated to enhance the system's robustness against overfitting. The evaluation of the proposed system's effectiveness will be conducted through rigorous testing, with a specific focus on its performance with a high-resolution dataset. Findings will be documented and analyzed, with particular attention to the types of defects present in the dataset.

1.5.2 Out of Scope

This study acknowledges certain aspects that fall outside its defined scope. Firstly, it does not encompass a comprehensive analysis of the broader reverse logistics processes beyond the specific focus on mobile screen damage detection. Additionally, the study does not delve into the intricacies of hardware or software components of mobile devices beyond the scope necessary for damage identification. The research does not extend to the development of physical robotics for automated handling of damaged mobile devices in a warehouse setting. Furthermore, while the proposed system aims to detect and classify specific damage types, it does not address the repair or recycling processes associated with the identified damages.

1.6 Significance of the Study

The study holds significant importance in the contemporary landscape of reverse logistics for mobile carriers, driven by the surge in e-commerce and heightened environmental regulations. In response to these challenges, businesses face the need to optimize processes related to the management of returned mobile phones. Manual inspection and processing of returned items, a common practice in warehouses, not only occupy substantial space and workforce but also lead to inefficiencies in the supply chain. Addressing these issues, the proposed mobile screen damage detection and classification system seeks to automate and streamline these warehouse procedures. By specifically targeting the identification of glass damage, a prevalent concern for both warehouse customers and vendors, the research aims to introduce a more efficient, cost-effective, and time-saving alternative to traditional manual methods.

The expected outcome of this research is the development of a scalable, robust, and diversified mobile screen damage detection and classification system. The system, designed to identify and categorize specific damage types such as oil, scratch, and stain, is anticipated to surpass the accuracy and efficiency of manual methods. Leveraging transfer learning, data augmentation, and regularization/dropout methods, the system aims to enhance its adaptability to various damage scenarios and exhibit robust performance against overfitting. The successful implementation of this system holds the potential to revolutionize reverse logistics operations for mobile carriers, offering cost reductions, minimized delays, and contributing to a more sustainable and efficient supply chain.

On a broader scale, the implications of this research extend nationally and internationally. At the national level, the study has the potential to significantly influence reverse logistics practices within the mobile industry. The efficient handling of reverse logistics is not only crucial for cost optimization but also aligns with global environmental regulations, emphasizing sustainability. The proposed automated system could serve as a model for other industries facing similar reverse logistics challenges. Internationally, the research contributes to the broader field of artificial intelligence and machine learning applications in supply chain management. The development of advanced detection systems for specific damage types extends beyond the mobile industry, potentially impacting practices in various sectors engaged in reverse logistics. This study aligns with the global push towards technological innovation in supply chain sustainability, making it relevant and impactful on an international scale.

1.7 Structure of the Study

This research is organized into several chapters, each providing a unique perspective on the development and evaluation of a scalable, robust, and diversified mobile screen damage detection and classification system. The structure of the study is as follows:

The study begins with a comprehensive Literature Review in Chapter 2, which provides a comprehensive overview of existing literature on machine learning (ML) and deep learning (DL) methods applied in damage detection and classification. It outlines the current landscape and identifies gaps that this research can address, laying the foundation for the proposed system.

Chapter 3 is dedicated to the Methodology. This chapter provides detailed insights into the design and implementation of the mobile screen damage detection and classification system. It explains the techniques employed, such as transfer learning, data augmentation, regularization, and dropout methods.

Chapter 4, Results and Discussion, presents the outcomes of the proposed system and compares the performance of different models used in the study. This chapter provides a critical analysis of the results, leading to the selection of the most efficient model for mobile screen damage detection and classification.

The study concludes with Chapter 5, which provides the Conclusion and Future Work. This final chapter synthesizes the key findings and their implications, and proposes potential areas for future research to further enhance the proposed system.

Each chapter plays a pivotal role in the overall study, building upon the previous chapters and contributing to the understanding and development of a more advanced and efficient mobile screen damage detection and classification system.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This literature review aims to explore the current state of knowledge in the field of automated damage detection and classification systems, specifically focusing on screen defects in returned mobile phones in the logistics sector. The chapter will delve into the methodologies, technologies, and applications of various damage detection systems, aiming to draw insights that can be applied to the context of screen defect detection.

The first section, Automation in Logistics, explores the transformative role of automation in logistics, particularly its impact on enhancing efficiency and accuracy in warehouse operations. Following this, the review delves into the realm of Automation in Defect Detection, examining the shift from traditional methods to automated systems and the role of machine learning in this process.

Broadening the scope, the review explores the Broad Applications of Damage Detection Systems, discussing systems applied in various industries such as automotive, infrastructure, and healthcare. This allows us to draw insights from a diverse range of applications and understand their relevance to screen defect detection.

The focus then narrows down to Screen Defect Detection and Classification in Returned Phones, a key area of focus for this research. A detailed exploration of this topic will provide valuable insights for the development of own defect detection system.

Finally, the review examines Related Research in the field of damage detection systems, providing a broader context for this research and drawing insights from a range of fields and applications. Each section of this review will identify relevant studies, methodologies, and technologies, with the aim of identifying gaps and opportunities that this research can address. The insights gained will provide a foundation for the design and implementation of an automated system for detecting and classifying screen defects in returned phones.

2.2 Automation in Logistics

In recent years, the growing interest in automation has become palpable across various industries, each embracing technological advancements to optimize processes and reduce reliance on human intervention. While sectors like engineering and medicine have made considerable strides in automation, the logistics industry finds itself at the nascent stages of this transformative journey. This subsection aims to scrutinize the current landscape of automation within logistics, focusing specifically on the integration of Artificial Intelligence (AI), machine learning (ML), and deep learning technologies. The deployment of these state-of-the-art technologies promises increased efficiency, reduced errors, and a substantial decrease in dependency on human labor, ultimately enhancing competitiveness within the logistics (Ferreira & Reis, 2023).

The foundation of automation in logistics lies in the application of AI, encompassing various facets such as ML, deep learning, and artificial neural networks. Machine learning, a subset of AI, plays a pivotal role in this paradigm, offering capabilities like anomaly detection. By identifying irregular patterns and deviations from the norm, ML contributes to adaptive logistics systems capable of swift and informed decision-making.

As the logistics industry embraces automation, the evolution of Human-Robot Interaction (HRI) emerges as a significant juncture. This collaborative effort between humans and robots transcends mere transactional engagement, envisioning the creation of social service robots that seamlessly coexist with their human counterparts. This collaborative synergy has the potential to revolutionize logistics operations, creating a dynamic and adaptable environment where robots complement human capabilities.

Simultaneously, under the umbrella of AI, deep learning stands as a powerful force, particularly in processing non-linear information. Deep learning models excel in interpreting intricate patterns within data, providing logistics systems with the capability to navigate complex relationships and adapt to evolving scenarios. In Figure 1, a visual depiction is presented, offering a structured overview of the interconnected topics and subtopics, elucidating the key components such as the relationships between AI, ML, and deep learning that are essential for comprehending the integration of automation technologies within the logistics sector (Ferreira & Reis, 2023).

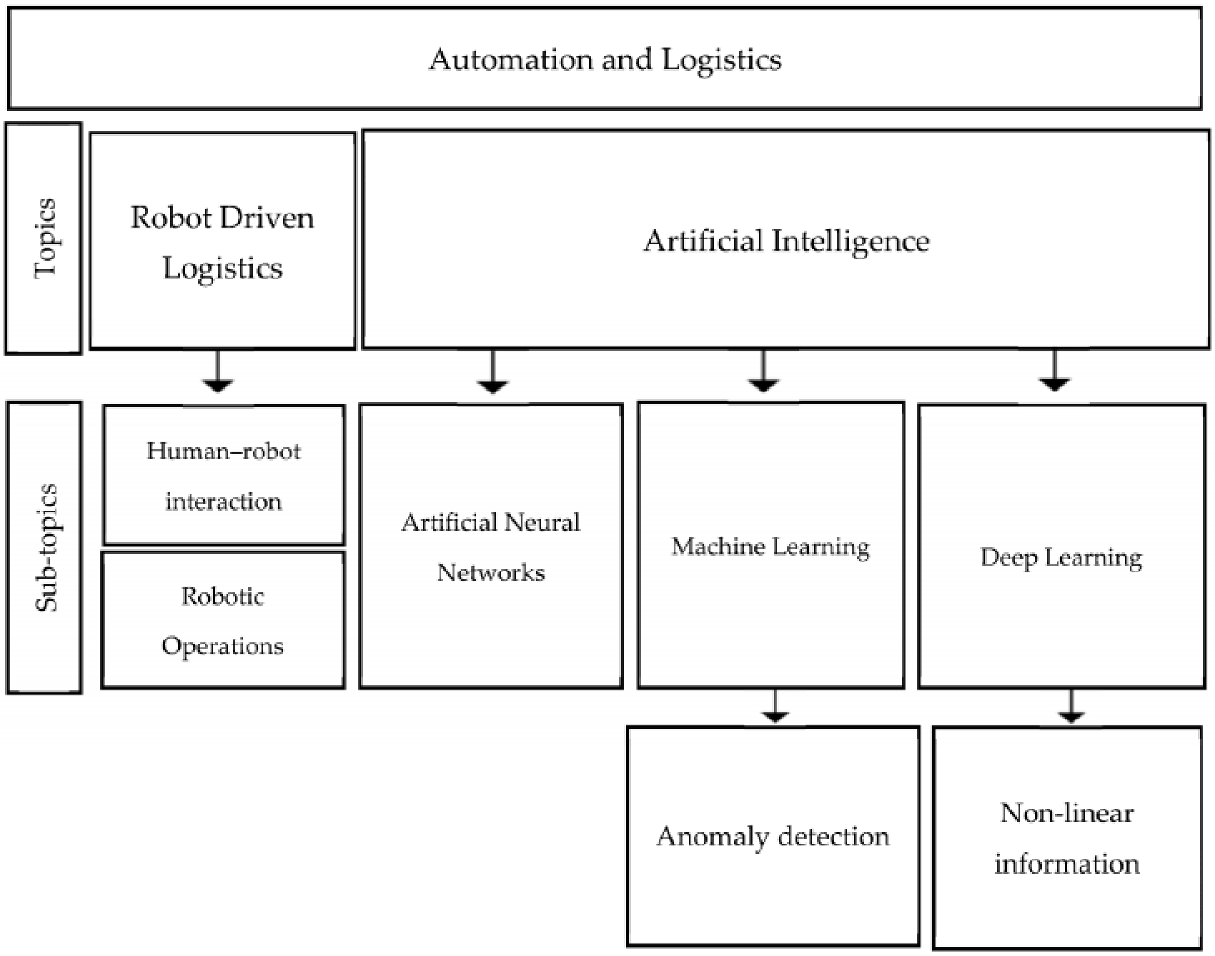


Figure 1: Emerging topics of automation in logistics.

Transitioning from the theoretical to the practical, automation in logistics manifests in various forms, each catering to different operational needs for maximum efficiency and productivity. Some of the most common applications and types of automation in logistics include (Matt et al., 2020):

Automated Loading and Unloading Systems: These systems greatly reduce human effort and increase speed in loading and unloading goods, enhancing overall operational efficiency.

Automated Guided Vehicles (AGVs): AGVs transport goods autonomously within a facility, following predefined paths, thereby reducing manual handling and increasing safety.

Automated Storage and Retrieval Systems (AS/RS): AS/RS are computer-controlled systems that can store and retrieve items from defined storage locations swiftly and accurately, optimizing storage space and inventory management.

Automatic Fork-lift Trucks: These mechanized handling vehicles are essential in warehouses for lifting and moving heavy goods, improving operational speed and reducing labor costs.

Conveyor Belts, Carousels, and Conveyor-based Sorting Systems: These systems facilitate the efficient movement and sorting of goods within a facility, reducing manual handling and enhancing workflow.

Industrial Robots/Robotics: Robots can perform a variety of tasks including picking, packing, and sorting, reducing the need for manual labor and minimizing errors.

Item-picking Devices: Automated item-picking devices can accurately pick and sort items for orders, increasing speed and reducing errors in order fulfillment.

Lift and Turntables/Aids and Linear Actuators: These devices help in lifting and moving heavy goods, increasing safety and reducing the strain on workers.

Mechanized Palletizing: Automated palletizing systems can quickly and accurately stack goods on pallets, reducing manual labor and increasing efficiency.

Moving Decks and Screening and/or Sorting Systems: Moving decks are mechanized platforms that efficiently transport goods within a facility, reducing manual labor and enhancing operational speed. On the other hand, screening and sorting systems employ automation to sort goods based on specified criteria like dimensions, weight, destination, or condition. In certain contexts, these systems can use technologies like image recognition to detect and separate damaged items, enhancing quality control processes. Together, these systems contribute to streamlined, efficient logistics operations, customizable to the specific needs of a facility.

In the broader context of automation in logistics, this research work could be conceptualized as falling under the category of "Moving Decks and Screening and/or Sorting Systems." While the system doesn't physically move or sort goods, it performs a type of screening – it filters devices based on their condition and identifies those with screen damage. This highlights the versatility of automation technologies, which can be tailored to meet a wide range of operational needs within the logistics sector.

In summary, the application of automation techniques, including machine learning and deep learning, in the logistics domain can guide practitioners and decision-makers in implementing effective automation strategies. This not only improves overall performance and adaptability in the ever-changing logistics landscape but also paves the way for a future where automation and human expertise work hand in hand to drive logistics towards unparalleled efficiency and success.

2.3 Automation in Damage Detection

Automation in damage detection is crucial across a variety of sectors, specifically in automotive, infrastructure, energy, e-commerce, and logistics. Advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), and computer vision are increasingly being adopted to detect, analyze, and make precise decisions regarding different types of damage with enhanced efficiency.

While certain types of damage may require specific detection methods, Machine Learning, especially when applied to image-based data, offers unique advantages. For instance, in the case of wind turbine blades, while certain techniques like strain measurement, acoustic emission, ultrasound, vibration, and thermography can detect specific types of damages (Movsessian et al., 2021), ML and computer vision provide a more comprehensive and efficient damage detection solution for visible surface damages.

In the realm of the automotive industry, infrastructure maintenance, energy sector, e-commerce and logistics, automated damage detection systems have demonstrated their ability to significantly improve the speed, accuracy, and efficiency of damage detection and analysis. Each of these sectors presents unique challenges and opportunities for the application of automation in damage detection.

To fully appreciate the impact of these automated systems, it's essential to understand how they compare to traditional methods of damage detection. This comparison, explored in the following section, will highlight the advantages and limitations of both approaches.

Further, the application of Machine Learning in these automated systems is a crucial factor contributing to their effectiveness. The next section will delve into how Machine Learning is applied in damage detection, enhancing the capabilities of these automated systems.

2.3.1 Traditional Methods vs. Automated Systems

Traditional methods of damage detection often rely on physical inspections and manual labor. These methods, although proven and reliable, are time-consuming and prone to human error. For example, a human inspector might miss minor but significant damage due to fatigue or oversight. Furthermore, in industries like automotive or infrastructure, where the scale of inspection can be enormous, traditional methods can be inefficient and costly.

On the other hand, automated systems for damage detection utilize advanced technologies like AI, ML, and computer vision. These systems, capable of working continuously without fatigue, offer higher efficiency, speed, and accuracy (Xie et al., 2020). They can detect complex damages that might be missed by human inspectors and can be integrated with other systems for real-time analysis and decision-making.

However, the effectiveness of automated systems, particularly those using Machine Learning, heavily depends on the availability and quality of the data used for training. In many cases, such as in civil structures, pre-damage and post-damage data sets are rarely available, making the training process for supervised ML classifiers challenging. Innovative solutions are needed to overcome these obstacles and train effective ML models even in the absence of measured damaged data (Avci et al., 2021). While automated systems require significant initial setup and maintenance costs, the long-term benefits in terms of efficiency, precision, and cost-saving often outweigh the initial investment.

The following table provides a concise comparison of these two methods, summarizing their key strengths and weaknesses.

Table 1: Comparison of Traditional and Automated Damage Detection Methods

	Traditional Methods	Automated Systems
Pros	• Proven and reliable techniques. • Can handle a variety of damage types.	• Higher efficiency and speed. • Greater accuracy. • Can detect complex damages. • Can work continuously without fatigue. • Can be integrated with other systems for real-time analysis
Cons	• Time-consuming. • Prone to human error. • Can be inefficient and costly on a large scale.	• Heavy dependence on the availability and quality of training data. • Significant initial setup and maintenance costs. • Challenges in training ML classifiers when pre-damage and post-damage data sets are not available

2.3.2 Application of Machine Learning in Damage Detection

Machine Learning, a subset of AI, plays a significant role in the effectiveness of automated damage detection systems. ML algorithms can learn from data, identify patterns, and make predictions, enhancing the capabilities of these systems.

In addition to detection, Machine Learning can also classify the severity or categorize different types of damage. This is particularly useful when the damage varies in terms of impact and required intervention. For example, a minor scratch on a car body might need a simple touch-up, while a significant dent could require more substantial repair or even part replacement. ML models can be trained to recognize and classify these differences accordingly, providing detailed insights that can guide subsequent actions.

Supervised learning (Jiang et al., 2020), a type of ML, is commonly used in damage detection where the algorithm is trained on a labeled dataset. For instance, in the case of car damage detection, the algorithm is trained using images of cars, labeled as 'damaged' or 'not damaged'. Once trained, the model can then predict whether a new image shows a damaged car or not, and also determine the severity or category of the damage.

Deep Learning (Lecun et al., 2015), a more complex form of ML, is often used for image-based damage detection. Techniques like Convolutional Neural Networks (CNNs), a type of deep learning model, are particularly effective at processing images, making them a good fit for tasks such as detecting damage on car bodies, wind turbines, or other objects.

However, it's crucial to note that the application of ML in damage detection requires a substantial volume of labeled data for training and significant computational resources. Also, the interpretability of ML models can pose challenges, particularly in industries where understanding the decision-making process is critical.

2.4 Broad Applications of Damage Detection Systems

Damage detection systems, particularly those utilizing Machine Learning and AI, find applications across a myriad of industries and sectors due to their versatility, efficiency, and precision. Here are some key areas where they are being widely used.

2.4.1 Automotive Industry and Related Sectors

Automated damage detection systems are indispensable in the automotive sector and its related industries. They can quickly and accurately detect and categorize damage to vehicles, such as dents, scratches, or rust. This can significantly streamline the process of vehicle inspections and maintenance, enhancing operational efficiency.

The effectiveness of these systems extends to insurance claims systems, accident reporting systems, car garages, and car trading services (Thomas et al., 2023).For insurance companies, an automated damage detection interface can expedite claim processing, reduce fraud, and improve customer satisfaction (Khan et al., 2021; Kyu & Woraratpanya, 2020). Accident reporting systems can benefit from quicker, more accurate damage assessments. Car garages can utilize this technology for efficient inspection and repair planning. Car trading services can leverage automated damage detection for accurate valuation and condition reporting.

In all these interconnected fields, the use of Machine Learning and AI in damage detection can lead to significant improvements in operational efficiency, cost-effectiveness, and customer experience.

2.4.2 Infrastructure and Construction

In the field of infrastructure and construction, these systems can detect structural damages in buildings, bridges (Zhang et al., 2020),and roads (Arya, Maeda, Kumar Ghosh, et al., 2020). By identifying issues early, it allows for preventative maintenance, improving safety and potentially saving substantial repair costs.

2.4.3 Energy Sector

Wind turbines (Movsessian et al., 2021), solar panels, and other energy infrastructure (Shihavuddin et al., 2021)can benefit from automated damage detection. For instance, ML algorithms can identify damage to wind turbine blades from captured images, aiding in maintaining the efficiency of the energy production process.

2.4.4 Aerospace Industry

In the aerospace industry, automated damage detection systems are critical for maintaining aircraft safety and integrity. Various techniques are employed to monitor the structural health of aircraft, allowing for the identification and location of potential issues. These systems collect structural data during various flight conditions, contributing to the development of post-flight structural health monitoring tools and real-time damage detection during flight (Gur, n.d.). This utilization of automated damage detection significantly enhances operational efficiency, safety, and cost-effectiveness in the aerospace industry.

2.4.5 E-commerce and Logistics

In the fast-paced world of e-commerce and logistics, automated damage detection systems are invaluable. These systems can inspect goods for any damages during the warehousing process, ensuring the quality of products stored and dispatched. This is particularly relevant in the growing market of secondhand goods, where accurate damage detection can boost consumer confidence and drive sales (Kaur Chatrath et al., 2022).

During packaging, these systems can also verify the integrity of the products and the packaging, crucial for protecting goods during transportation. Furthermore, with the increasing shift towards online shopping, these systems can help address the susceptibility issues consumers encounter when assessing product condition based on online images (Kaur Chatrath et al., 2022).

Automated damage detection can also streamline the handling of product returns. By quickly assessing the condition of returned items, it can expedite the refund or exchange process, enhancing the customer experience.

In industries dealing with delicate and high-value goods, such as electronics, these systems can be especially beneficial. For instance, in the mobile phone industry, automated damage detection can play a pivotal role in managing returned products by efficiently identifying and classifying screen defects.

Overall, the integration of automated damage detection in e-commerce and logistics operations, as proposed in methods like the Product Image-based Vulnerability Detection (PIVD) (Kaur Chatrath et al., 2022), can significantly improve efficiency, cost-effectiveness, and customer satisfaction. The following section explores the specific application of these systems in detecting and classifying screen defects in returned phones.

2.5 Screen Damage Detection and Classification in Returned Phones

Screen damage detection and classification is crucial across various sectors, including manufacturing, e-commerce, and logistics. It's especially relevant in the mobile phone industry, where these systems significantly streamline the handling of returned or second-hand devices.

Mobile phone glass, which is often subject to breakage, requires careful evaluation during the exchange or reselling process. E-commerce companies, for instance, invest significant resources in checking the condition of mobile screens during these processes. Automated damage detection systems facilitate this task by accurately determining the level of screen damage, which helps predict the device's condition. This assists in decision-making, whether a device can be resold as 'Used/Refurbished', or should be scrapped for parts (Selvi et al., 2021).

In the manufacturing sector, these systems play a vital role in quality assurance, particularly for Mobile Phone Cover Glass (MPCG). They help identify and classify various surface defects that occur during the production process, such as scratches, pits, dirt, and edge breakages. The ability to accurately classify these defects is essential since different defects necessitate varying remediation strategies [Mobile phone screen - manufacture].

In logistics, particularly in the returns process, these systems can significantly enhance operational efficiency. By providing rapid and accurate assessments of screen damage, they expedite the handling of returned devices, leading to improved customer satisfaction and substantial cost savings.

Overall, the use of automated systems for screen damage detection and classification is increasingly becoming a standard practice across sectors, given its significant contributions to operational efficiency, cost-effectiveness, and customer satisfaction.

2.6 Related Research

Deep learning methods, a subset of machine learning techniques, are characterized by their ability to learn from large amounts of data and extract complex patterns. These methods employ artificial neural networks, which are inspired by the structure and function of the human brain, to model complex relationships between input data and desired outputs. Deep learning has revolutionized various fields, including computer vision, natural language processing, and speech recognition, and has proven to be particularly effective in tasks involving image and pattern recognition. Deep learning methods have shown substantial progress in diverse areas such as car damage detection (Sruthy et al., 2021; Thomas et al., 2023; Widjojo et al., 2022), road damage identification (Arya, Maeda, Ghosh, et al., 2020; Chen et al., 2023; Pham et al., 2020, 2022), COVID-19 detection from ECG images (Irungu et al., 2023), tank barrier surface damage detection (Dyk & Drahansky, 2023), and structural damage identification (Feng et al., 2019). These methods have also proved to be valuable in the warehouse logistics and e-commerce industry, specifically in addressing the challenge of mobile phone screen surface defect segmentation. Deep learning techniques are emerging as promising solutions to accurately identify and classify these defects, thereby enhancing quality control efficiency in these industries. Deep learning methodologies commonly adopted in these fields include Faster Region-Based Convolutional Neural Networks (Faster R-CNN), You Only Look Once (YOLO) (Pham et al., 2020), Single Shot Detection (SSD), and various pre-trained models with transfer learning such as ResNet, DenseNet, SqueezeNet, MobileNetV2, Inception V3, Xception, VGG16, and VGG19 etc (Selvi et al., 2021; Sruthy et al., 2021; Widjojo et al., 2022). These methodologies highlight the versatility and broad applicability of deep learning across various domains, providing valuable insights for this research.

The study by (Yongfa Lv et al., n.d., 2019) presents a model that leverages Deep Convolutional Generative Adversarial Networks (DCGAN) and Faster R-CNN for detecting defects in mobile phone screen cover glass. Although this model is designed specifically for small sample learning and is effective, it exhibits limitations in speed and scalability when dealing with larger sample sizes.

Li et al., (2021) further contributes to the field by proposing an improved Fuzzy C-Means (FCM) clustering method for extracting defects from mobile phone screen covers. This method, similar to the model presented by (Yongfa Lv et al., n.d., 2019), shows efficacy but is currently limited to small-batch inspections. This highlights a scalability issue that is common in these studies, and the adaptability of this method to larger-scale, production line applications is yet to be confirmed.

Transfer learning has emerged as a powerful technique for addressing complex problems in various domains, including damage detection and classification. Recent studies have demonstrated the effectiveness of transfer learning in classifying and detecting damage in various infrastructure components, such as smartphones, vehicles and hydro-junction infrastructure. (Selvi et al., 2021) proposed a CNN-based system for classifying the level of damage in mobile glass using various pre-trained models such as Densenet, Resnet, Squeezenet, as well as custom models. The system achieves an accuracy of 85%, but the study concludes that the model could be improved in the future by considering the various complex constraints and assumptions of deep learning algorithms.

Feng et al., (2019) applied transfer learning to detect damage in hydro-junction infrastructure using a deep convolutional neural network. Their method achieved a high detection accuracy of 96.8%, surpassing traditional techniques. These studies underscore the flexibility and potential of transfer learning across various domains.

Pham et al., (2020) study focuses on the detection and classification of road damages using Detectron2's implementation of Faster R-CNN. The experiments were conducted using the Global Road Damage Detection Challenge 2020 dataset. The results show the X101-FPN base model for Faster R-CNN was effective and adaptable across different countries. However, despite promising visualizations, the F1 scores were low, indicating room for improvement. Similar to this research on glass defect detection and classification, this study involves the application of machine learning models for damage detection. The use of pre-trained models for transfer learning in this study is analogous to my approach.

Khan et al., (2021) demonstrated the effectiveness of CNNs in classifying various vehicle damage types, with MobileNet and VGG19 achieving accuracies of 70% and 50% respectively. In a subsequent study, (Sruthy et al., 2021) explored the application of CNNs for the detection, analysis, and estimation of various types of car damage. Utilizing transfer learning-based models from the Keras library, such as InceptionV3, Xception, VGG16, VGG19, ResNet50, and MobileNet, they aimed to predict and classify damage. Their analysis revealed that MobileNet exhibited the highest accuracy, reaching 97.28% in predicting and classifying damage types, and also demonstrated a faster training speed compared to other models. Building on these insights, Widjojo et al., (2022) conducted experiments highlighting the potential of Mask R-CNN in detecting damaged car parts. A simple modification on the CNN classification model resulted in a 9% average increase in the F1 score. Notably, MobileNetV2 emerged as the top-performing classification model, boasting an impressive F1 score of up to 91%.

Together, these studies form a comprehensive basis for understanding the application of transfer learning in damage detection. They provide insightful pathways for future investigations specifically focused on the detection and classification of defects in mobile phone screens using various pre-trained models.

Farhadi et al., (2022) research addresses overfitting in deep learning due to complex layer structures. It explores a combination of L1, L2, Elastic Net-regularization, and Dropout methods, comparing the performance of this combined model with a CNN model without regularization. The study found that a model combining Dropout and Elastic Net regularization outperformed others. In context of my work on mobile phone screen defect detection and classification, this study's approach to mitigate overfitting using combined regularization and Dropout methods offers valuable insights.

Mahyoub et al., (2023) study examines the use of deep learning for automatic car damage detection and addresses the challenge of limited datasets with data augmentation techniques. Specifically, Generative Adversarial Networks (GANs) were used to increase the size and improve the class balance of the dataset, which resulted in better model performance. In relation to my research on mobile phone screen defect detection and classification, this study's application of data augmentation highlights a potential strategy to overcome similar dataset limitations.

The existing literature on mobile screen defect detection and classification has highlighted the potential of deep learning methods. Nonetheless, challenges persist, such as the scalability of models to larger sample sizes and the detection of a diverse array of damage types. Transfer learning emerges as a promising approach to address these challenges. For instance, (Feng et al., 2019) demonstrated that a deep convolutional neural network with transfer learning could accurately detect damage in hydro-junction infrastructure. Data augmentation is another significant technique that enhances the performance of deep learning models. (Mahyoub et al., 2023) used data augmentation and GANs to increase the size and improve the class balance of a dataset for automatic car damage detection, leading to improved model performance. Furthermore, to tackle overfitting due to complex layer structures, a combination of L1, L2, Elastic Net-regularization, and Dropout methods can be employed. (Farhadi et al., 2022) compared the performance of a model using these combined methods with a CNN model without regularization. They found that the model with combined regularization and dropout methods outperformed the others. This approach offers valuable insights for my work on glass damage detection, and I aim to implement these techniques to mitigate overfitting.

Based on these findings, my research will focus on developing a mobile screen damage detection and classification system that is scalable, can detect a diverse range of damage types, and is robust to overfitting. I plan to use transfer learning, data augmentation, and a combination of regularization and dropout methods to achieve these goals.

2.7 Summary

This chapter has provided an in-depth look into the role of automation in logistics, with a specific focus on damage detection. Discussed the advantages of automated systems over traditional methods, emphasizing their accuracy, efficiency, and the ability to handle large-scale operations. The application of machine learning in damage detection was examined, highlighting its potential in image processing and pattern recognition tasks.

Then explored the broad applications of damage detection systems across various sectors like the automotive industry, infrastructure and construction, energy, aerospace, and notably, e-commerce and logistics. The importance of screen damage detection and classification in returned phones was underscored, showing its critical role in handling returns and improving operational efficiency in the e-commerce and logistics sectors.

The review of related research revealed the significant strides made in the field, with deep learning methods showing substantial promise in tasks involving image and pattern recognition. These methodologies have been instrumental in diverse areas such as car damage detection, road damage identification, and mobile phone screen defect segmentation.

Building on these insights, the focus of this research will be to develop a robust and scalable mobile screen damage detection and classification system. The subsequent chapters will delve into the research methodology, experimental setup, and results, providing a comprehensive account of this research journey.

In conclusion, the use of automation and machine learning in damage detection presents vast potential, promising significant improvements in various sectors, particularly in handling returned phones in logistics, which is the primary focus of this research.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

The methodology of this research is designed to systematically approach the challenge of mobile screen damage detection and classification. It entails multiple stages, each critical to the development of a robust and accurate machine learning model. This chapter outlines the comprehensive methodology employed in this study, starting from data selection and preprocessing, to modelling and evaluation of results.

The methodology adheres to a systematic and iterative approach that allows for continuous refinement at each stage. It starts with the careful selection of a publicly available dataset, followed by crucial steps of data preprocessing and transformation, including data augmentation techniques. This chapter also discusses the application of advanced machine learning techniques, specifically transfer learning with pre-trained models, to achieve accurate classification of mobile screen damages. The evaluation of the model's performance and the tools utilized to implement this research are also elaborated.

By detailing each stage of the methodology, this chapter seeks to provide a clear and replicable blueprint of the research process. This approach not only ensures the integrity and reliability of the research but also contributes to the broader body of knowledge by offering a methodological framework that can be adopted or refined in future research in this field.

3.2 Methodology

3.2.1 Data Selection

The data selection process for this study was primarily focused on sourcing a diverse and representative dataset that would support the development of a robust and accurate machine learning model for mobile screen damage detection and classification.

The chosen dataset for this research is a publicly available collection of high-resolution images representing various conditions of mobile screens. This dataset was selected for its diversity in representing three types of mobile screen damages: oil, scratch, and stain, each having a significant number of associated images. Additionally, the dataset includes images of non-defective screens, providing a reference for good condition screens.

The decision to use a publicly available dataset was driven by the need to ensure that the research process is replicable and that the findings are verifiable. Publicly available datasets also offer the advantage of being pre-vetted and commonly used in research, thus offering a benchmark for comparison with other studies.

In addition to the existing images, the data selection process also planned for the utilization of data augmentation techniques to generate more diverse data. This step is crucial in machine learning as it helps improve the model's ability to generalize, thereby enhancing its performance on unseen data.

The data selection process culminated in the creation of a balanced and diverse dataset that caters to the needs of the machine learning model, setting the stage for the subsequent steps of preprocessing and transformation.

3.2.2 Dataset Description

The dataset utilized for this research comprises 1200 high-resolution images (1920×1080) showcasing three types of mobile screen damages: oil, scratch, and stain. Each type of damage is represented by 400 images, making for a balanced representation. In addition, the dataset includes 20 non-defective images that serve as a standard for comparison. These images have been captured using an industrial camera and are annotated in the PASCAL VOC format.

Figures 2 to 5 present sample images from the dataset. Figure 2 showcases an oil damage sample, Figure 3 represents a scratch damage sample, Figure 4 is an example of a stain damage sample, and Figure 5 depicts a non-defective screen sample.

Figure 2: Oil Damage Sample from the Mobile Screen Damage Dataset

Figure 3: Scratch Damage Sample from the Mobile Screen Damage Dataset

Figure 4: Stain Damage Sample from the Mobile Screen Damage Dataset

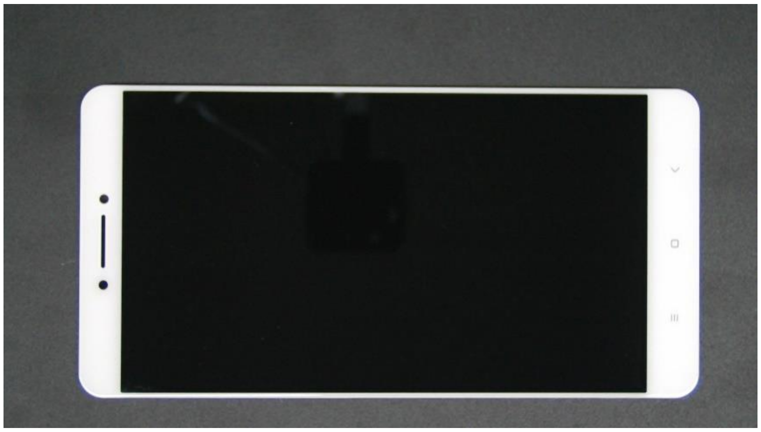


Figure 5: Non-Defective Screen Sample from the Mobile Screen Damage Dataset

To enhance the volume and diversity of the dataset, data augmentation techniques will be employed. These techniques generate additional images and help in preventing overfitting, thereby enhancing the model's ability to generalize from the training data.

The dataset and associated resources are publicly accessible on Kaggle (https://www.kaggle.com/datasets/girish17019/mobile-phone-defect-segmentation-dataset/data) and GitHub (https://github.com/jianzhang96/MSD/).

The careful selection and comprehensive description of this dataset set the foundation for the development of a machine learning model adept at accurately detecting and classifying different types of mobile screen damages.

3.2.3 Data Pre-Processing

Data preprocessing is a crucial phase in the machine learning pipeline, where raw data is transformed into a format that can be readily used by predictive models. This process can include multiple stages.

Firstly, images are resized to ensure uniform input to the model. Secondly, pixel values are normalized to a standard scale between 0 and 1, which enhances computational efficiency and promotes stability during model training.

To add more variety to the dataset and enhance the model's generalization ability, data augmentation techniques such as rotation, zooming, shifting, and flipping will be applied.

Lastly, a part of the dataset (20% in this case) is set aside for validation purposes to measure the model's performance and prevent overfitting.

These preprocessing steps are essential to prepare the data for effective training with machine learning models.

3.2.4 Data Transformation

Data augmentation is a vital phase in this research, aimed at artificially expanding the size and variability of the dataset. This process involves applying several transformations to the pre-processed images, generating different versions of the same image, each reflecting various scenarios or perspectives of mobile screen damage.

The augmentation techniques used in this research include image rotation (up to 30 degrees), flipping (both horizontally and vertically), zooming (up to 20%), and shifting (up to 10% width and height). These techniques are implemented using TensorFlow's 'ImageDataGenerator' function.

Through these augmentation techniques, I expect to generate an additional 4000 images, significantly expanding the dataset. This exposure to a wider array of damage types and perspectives enhances the model's ability to generalize and accurately detect a diverse range of damage types. In turn, this process increases the robustness of the model, making it less likely to overfit to the training data and more capable of performing well on unseen data.

3.2.5 Modelling

This research adopts a modeling phase that leverages the power of transfer learning, using pre-trained models to enhance the accuracy of the damage detection system. Utilizing models that have been pre-trained on large image datasets, like InceptionV3, Xception, VGG16, VGG19, and DenseNet201, allows us to benefit from the complex feature extraction capabilities these models have learned. These models, trained on large-scale image databases like ImageNet, have demonstrated robust performance in image classification tasks, making them potential candidates for my task of mobile screen damage detection. The final model selection will be based on performance metrics such as accuracy, precision, recall, and F1 score.

In addition to using transfer learning, the model will incorporate L1/L2 (whichever applies) regularization and dropout methods to prevent overfitting. Regularization adds a penalty to the loss function to discourage overly complex models, while dropout randomly ignores selected neurons during training, reducing the model's dependency on any single neuron.

This combination of transfer learning with regularization and dropout methods aims to achieve a balance between the model's ability to learn from the data and its robustness, which is crucial for ensuring the model's generalization to new, unseen data and enhancing its practical applicability.

3.2.6 Evaluation

The evaluation of the model's performance is a critical part of this research. Various metrics will be used to assess the model's accuracy and robustness. Specifically, the model will be evaluated based on the following key metrics:

Accuracy: This is the proportion of total predictions that the model gets right. It's a useful measure when the target classes are well balanced.

Precision: This measures the proportion of positive identifications that were actually correct. It is used as a measure of the model's relevancy.

Recall (Sensitivity): This measures the proportion of actual positives that were correctly identified. It is used as a measure of the model's completeness.

F1 Score: This is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the Accuracy Metric.

In addition to these metrics, a confusion matrix will be generated to visualize the performance of the model in classifying the images correctly.

The model's performance will be evaluated on both the training and validation datasets to ensure it is learning effectively and not just memorizing the training data (overfitting). If overfitting is detected, additional measures such as increasing dropout rates or adding more regularization may be employed to improve the model's generalization ability.

Furthermore, the chosen model's performance (e.g., InceptionV3, Xception, VGG16, VGG19, DenseNet201) will be compared to ascertain the most effective architecture for this specific task. The selection will be based on the balance between model performance (accuracy, precision, recall, and F1 score) and computational efficiency.

This comprehensive evaluation process aims to ensure that the final model is robust, accurate, and capable of generalizing well to new, unseen data.

3.3 Logical flow of the system

The research will be conducted following a comprehensive and iterative workflow that encompasses several stages, including dataset acquisition, data preprocessing, data transformation/augmentation, modelling, evaluation, and result interpretation.

Starting with dataset acquisition, relevant images of mobile screens with various damage types are collected from the entirety of a public dataset. Once the dataset is compiled, it undergoes preprocessing, which involves steps like resizing images and normalizing pixel values to standardize the data and facilitate efficient computation.

The next stage is data transformation/augmentation, where the preprocessed images are artificially manipulated to generate a diverse array of image variations. This enhances the dataset and exposes the model to a wider range of damage scenarios, boosting its ability to generalize.

After data augmentation, the modelling stage begins. Here, the power of transfer learning is leveraged by utilizing pre-trained models to enhance the damage detection system's accuracy. Regularization and dropout techniques are incorporated to prevent overfitting.

The model's performance is then evaluated on the validation dataset using metrics such as accuracy, precision, recall, and the F1 score. This evaluation phase ensures the model is performing well and can generalize to unseen data.

Finally, in the result interpretation stage, the model's outputs are analyzed and interpreted in the context of mobile screen damage detection. The findings from this stage can then inform further refinements in the previous stages.

In essence, each phase is intrinsically linked, with findings from each stage informing adjustments and improvements in the others, fostering a dynamic and continuously evolving research process.

A visual representation of the workflow is depicted in the below Figure 6.

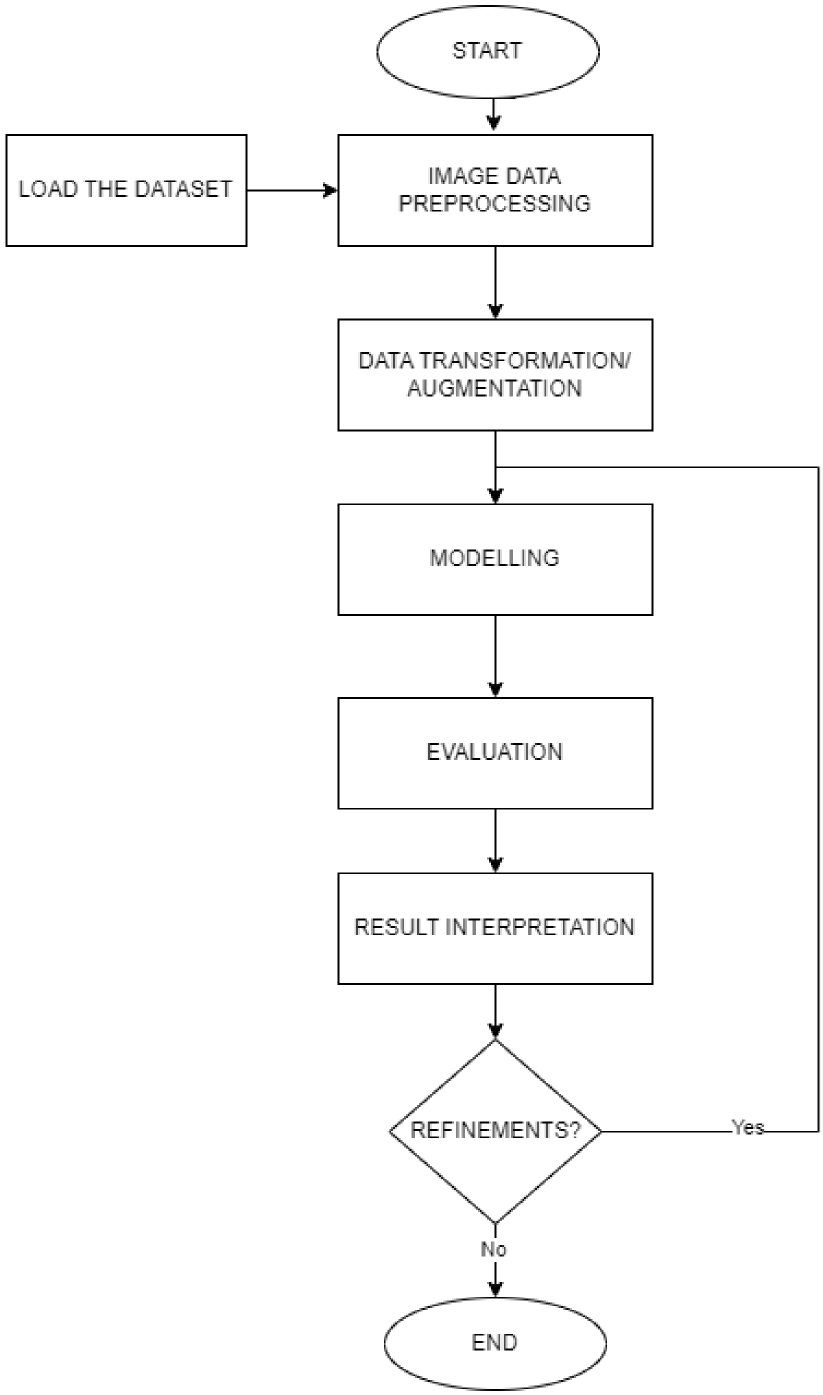


Figure 6: Logical Flow the System

3.4 Tools

3.4.1 Python

Python is a high-level, interpreted programming language widely used for its readable syntax and extensive library support. It's the language of choice for many data scientists and machine learning practitioners due to its simplicity and the availability of numerous scientific computing libraries such as NumPy and pandas.

3.4.2 TensorFlow

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and community resources that assist researchers in building and deploying machine learning models. Its flexible architecture allows easy deployment of computation across various platforms, including CPUs, GPUs, and TPUs.

3.4.3 Keras

Keras is a user-friendly neural network library written in Python. It's built on top of TensorFlow and provides a simplified interface for creating and training deep learning models. It supports a wide array of neural network architectures, making it a versatile tool for many machine learning tasks.

3.4.4 Google Colab

Google Colaboratory, or "Colab" for short, is a free cloud service hosted by Google. It provides a Jupyter notebook environment that requires no setup and runs entirely in the cloud. Colab is particularly useful for machine learning tasks due to its free GPU support, making it an excellent tool for training deep learning models.

3.4.5 Other Libraries

Other Python libraries that will be used in this research include NumPy and pandas for data handling, Matplotlib and Seaborn for data visualization, and Scikit-learn for various machine learning tasks such as splitting the dataset into training and test sets and evaluating the model's performance.

Each of these tools plays a vital role in the research, from data handling and model creation to visualization and model evaluation.

3.5 Summary

This chapter outlined the comprehensive research methodology used in this study. The process commenced with the selection of a publicly available dataset of mobile screen images, followed by a thorough description of the dataset's properties. The images then underwent preprocessing and transformation, including data augmentation techniques to enhance the model's generalization capability.

Pre-trained models were employed in the modelling stage, utilizing the power of transfer learning to enhance the system's accuracy. Regularization and dropout techniques were incorporated to prevent overfitting. The model's performance was evaluated using key metrics such as accuracy, precision, recall, and the F1 score.

The logical flow of the system was discussed, highlighting the interconnected and iterative nature of the process. The tools used in the research were also detailed, including Python, TensorFlow, Keras, Google Colab, and other Python libraries.

In essence, this chapter provided a robust and systematic roadmap for detecting mobile screen damage. The methodology's iterative nature allows for continuous refinement, ensuring the system's optimal performance. This comprehensive approach forms the backbone of the research, paving the way for the subsequent stages of model implementation and evaluation.

CHAPTER 4: IMPLEMENTATION OF DEEP LEARNING MODELS

4.1 Introduction

This chapter presents an in-depth exploration of the practical application of the five deep learning models employed in this study. These models, namely InceptionV3, Xception, DenseNet201, VGG16, and VGG19 were selected for their exceptional capabilities and proven efficacy in the field of image-based machine learning tasks.

The chapter begins with an overview of each of the five models. This includes a brief discussion of their architecture, underlying algorithms, and unique features that make them suitable for the task at hand – automated detection and classification of screen defects in returned phones.

Following this, the chapter delve into the specifics of the dataset utilized for this research. This section provides a thorough description of the dataset, including its source, the number of images it contains, the variety of screen defects represented, and the format and resolution of the images.

Subsequent sections are dedicated to the crucial steps of data preprocessing, necessary for preparing the dataset for effective model training. These steps encompass image resizing and data augmentation. Image resizing ensures that all images conform to the input size requirements of the deep learning models, while data augmentation increases the variability and size of the dataset, leading to more robust and generalized models.

The chapter then transitions into the practical implementation of the models. This part explains how the dataset was divided into training and validation sets, the parameters used for model compilation and training, and the process of applying each model to the task of defect detection and classification.

Finally, the chapter elaborates on the performance evaluation metrics chosen to assess the efficiency and accuracy of each model. These metrics provide a quantifiable measure of the models' performance, which forms the basis for the comparative analysis in the subsequent chapter.

The objective of this chapter is to provide a detailed account of the processes involved in implementing the deep learning models for this study. Understanding these processes helps gain valuable insights into the performance of each model and their suitability for the task of automated screen defect detection

4.2 Overview of Selected Models

This study employs a selection of five deep learning models known for their effectiveness in image classification tasks. Each model has unique characteristics that make it suitable for the task of detecting and classifying screen defects in returned phones. This section provides an overview of each model, beginning with DenseNet201.

4.2.1 DenseNet201

DenseNet201, or Densely Connected Convolutional Network-201, is a deep learning model that is part of the DenseNet family. The architecture of DenseNet201 is unique due to its dense connectivity pattern.

In DenseNet201 (Charisma & Adhinata, 2023; Jaiswal et al., 2021) each layer is connected to every other layer in a feed-forward fashion. This means that each layer receives feature-maps from all preceding layers and passes on its own feature-maps to all subsequent layers. This drastically differentiates DenseNet201 from traditional convolutional neural networks where each layer is only connected to the next layer.

The architecture of DenseNet201 consists of 201 layers, hence the name. These layers include densely connected layers, transition layers, and a final classification layer. Densely connected layers are organized into dense blocks, where each layer within the block is connected to every other layer. Transition layers, which consist of a batch normalization layer and a pooling layer, are used between the dense blocks to reduce dimensionality and control overfitting. The final part of DenseNet201's architecture is a classification layer, which uses a softmax activation function to output the probability distribution over the classes. The architecture of DenseNet201 is illustrated in the Figure 7 (Charisma & Adhinata, 2023; Jaiswal et al., 2021):

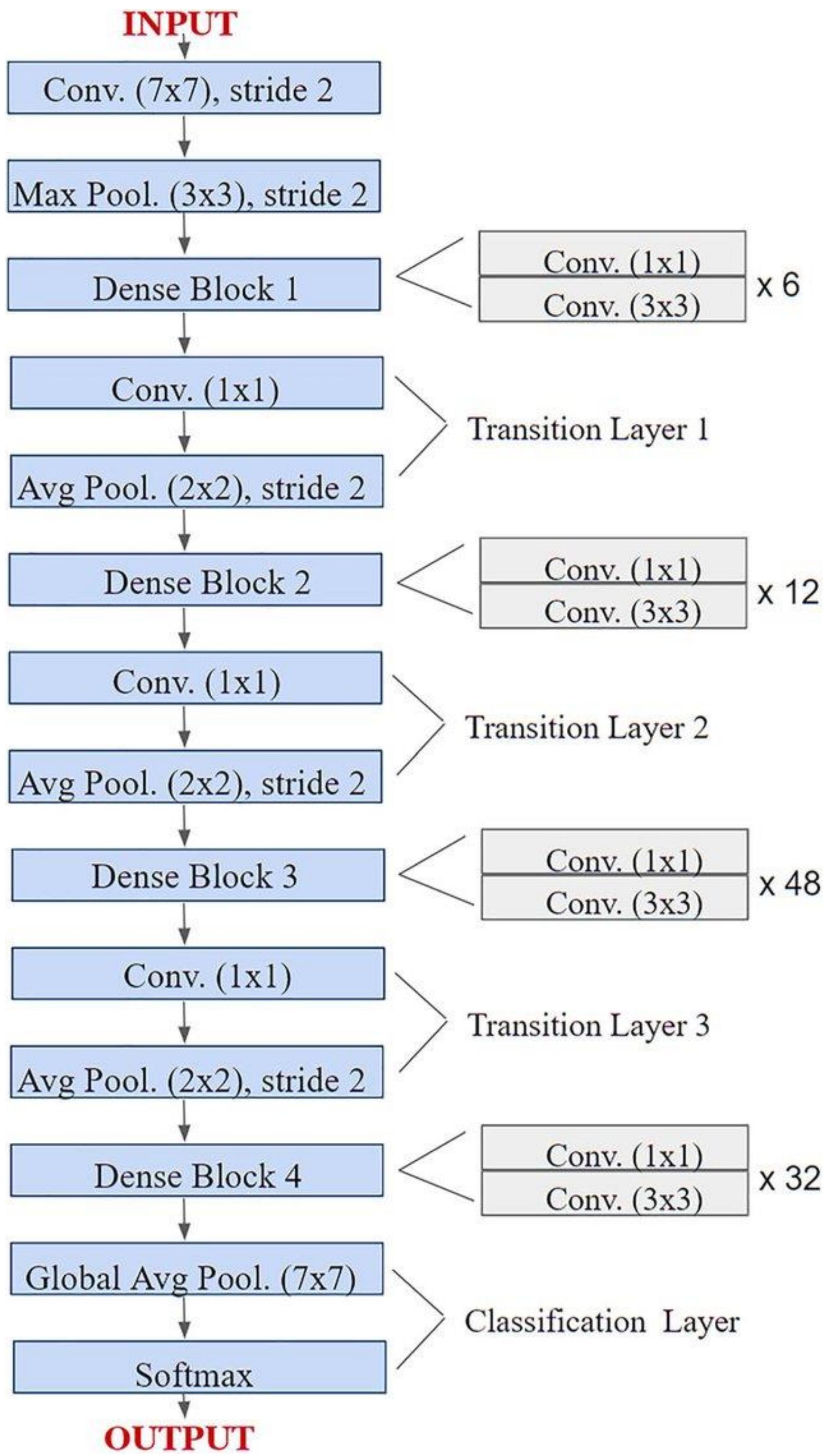


Figure 7 (Charisma & Adhinata, 2023; Jaiswal et al., 2021) : Layered Architecture of DenseNet201

One of the significant advantages of DenseNet201 is its efficiency. Due to its dense connections, fewer parameters are required, making the model more computationally efficient. Additionally, DenseNet201 is less prone to overfitting, which is a common problem in deep learning models.

In the context of this research, DenseNet201 is particularly suited for the task of detecting and classifying screen defects in returned phones for several reasons. Firstly, its ability to effectively identify and extract complex patterns from images aligns well with the need to discern subtle screen defects. Secondly, its computational efficiency is valuable when processing a large number of returned phones, a likely scenario in real-world applications. Finally, DenseNet201's resistance to overfitting ensures that the model can generalize well from the training data to unseen screen defects, a crucial requirement for reliable defect detection and classification.

In summary, DenseNet201 is known for its unique architecture leading to computational efficiency and resistance to overfitting. These attributes, coupled with its capacity to extract complex patterns from images, make it an excellent choice for the task of detecting and classifying screen defects in returned phones.

4.2.2 InceptionV3

InceptionV3, another model used in this study, is a convolutional neural network that is 48 layers deep. It's part of Google's Inception series and is recognized for its complex and efficient architecture that allows it to extract features from images at multiple scales.

The architecture of InceptionV3, as depicted in the Figure:8 (Ali et al., 2021) below, is characterized by the use of Inception modules. These modules, or blocks, are a structure of convolutional layers with different kernel sizes, which allow the network to effectively learn both local features with small convolutions and higher-level features with larger convolutions. The outputs of these convolutions are then concatenated and passed to the next layer.

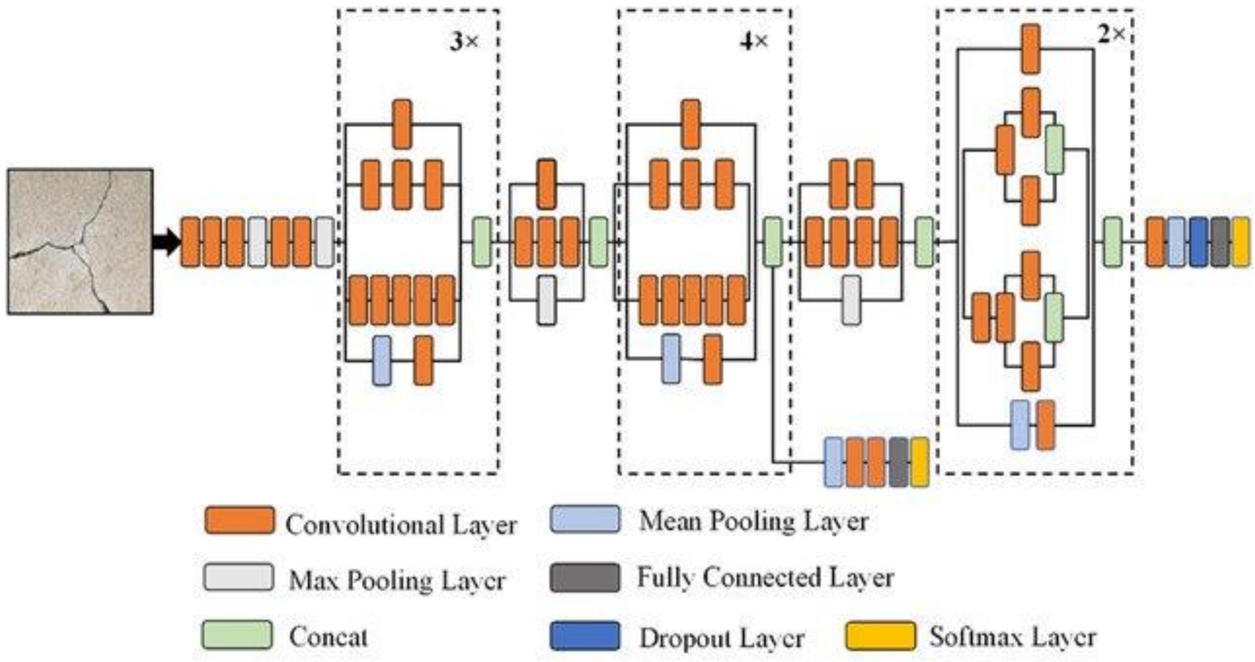


Figure 8 (Ali et al., 2021) : The architecture of Inception-V3 model

InceptionV3 also implements several techniques that improve its performance and efficiency. These include factorization, which reduces the number of parameters, and batch normalization, which speeds up training. It also uses auxiliary classifiers to combat the vanishing gradient problem, a common issue in deep networks.

In the context of this study, InceptionV3's ability to extract features at multiple scales is particularly advantageous for detecting and classifying screen defects in returned phones. Defects can manifest in various scales and forms, and a model that can capture this diversity is essential. Furthermore, InceptionV3's advanced architecture and performance-enhancing techniques align well with the requirements of efficient and accurate defect detection.

In summary, InceptionV3 is known for its efficient architecture that effectively captures multi-scale features. These attributes, combined with its performance-enhancing techniques, make it a well-suited choice for the task of detecting and classifying screen defects in returned phones.

4.2.3 VGG16

VGG16, also known as Visual Geometry Group-16, is a convolutional neural network model proposed by the Visual Geometry Group from the University of Oxford. This model is recognized for its simplicity and high performance, achieving state-of-the-art results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014.

VGG16's architecture, shown in the Figure:9 (Ali et al., 2021) below, is characterized by its uniformity. It consists of 16 layers, including 13 convolutional layers, 2 fully connected layers, and a final classification layer. The convolutional layers use small 3x3 filters, which are more efficient than larger filters and allow the network to learn more complex features.

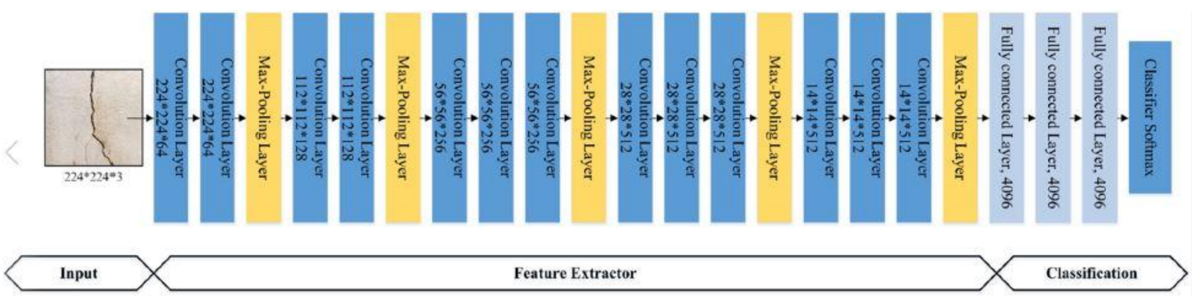


Figure 9 (Ali et al., 2021): The architecture of VGG16 model

All convolutional layers in VGG16 use a stride of 1 and are padded to maintain the same spatial dimensions. Following each set of convolutional layers, a max pooling layer is used to reduce the dimensionality and prevent overfitting. The two fully connected layers each have 4096 nodes, and the final classification layer uses a softmax activation function to output probabilities for each class.

VGG16's ability to extract excellent features from images, despite its simplicity, makes it a suitable choice for image classification tasks such as detecting and classifying screen defects in returned phones. In fact, its performance in this study has been highly competitive, indicating its effectiveness in handling this task.

In summary, VGG16, with its strong feature extraction ability, simplicity, and high performance, is an excellent model for tasks involving complex image classification. These qualities make it well-suited for detecting and classifying screen defects in returned phones, as evidenced by its high accuracy in this study.

4.2.4 VGG19

VGG19, another model developed by the Visual Geometry Group at the University of Oxford, is a convolutional neural network known for its deep architecture and excellent performance on image classification tasks.

The architecture of VGG19, as shown in the Figure: 10 (Ali et al., 2021) below, consists of 19 layers, which include 16 convolutional layers, 2 fully connected layers, and a final classification layer. Similar to VGG16, VGG19 employs small 3x3 convolutional filters throughout its architecture, allowing the network to learn more complex hierarchical features.

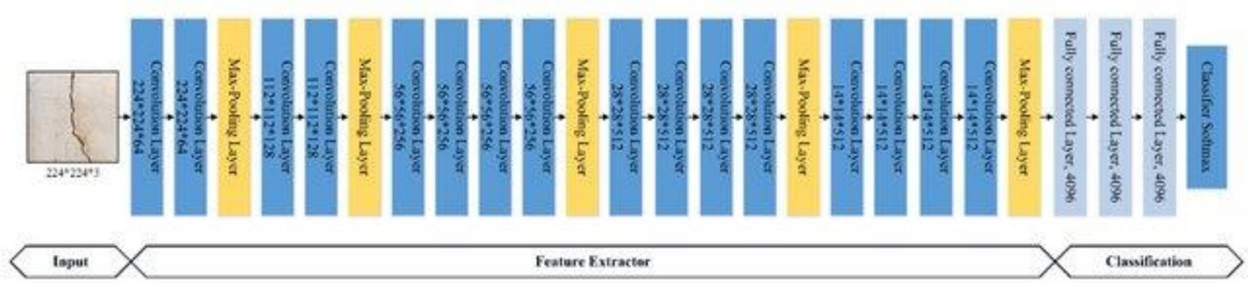


Figure 10 (Ali et al., 2021): The architecture of VGG19 model

All convolutional layers in VGG19 use a stride of 1 and padding to preserve spatial dimensions. Following each set of convolutional layers, max pooling layers are used to reduce spatial dimensions and control overfitting. The two fully connected layers each contain 4096 nodes, and the final softmax layer outputs the probability distribution over the classes.

VGG19's depth allows it to learn more complex features, potentially leading to improved performance on certain tasks. However, this also increases the model's complexity and computational requirements, which is a trade-off to consider.

In the context of this study, VGG19's ability to learn complex features from images aligns well with the task of detecting and classifying screen defects in returned phones. Given its high accuracy in this study, VGG19's performance for this specific task has proven to be very effective.

In summary, VGG19, with its deep architecture and excellent feature extraction capability, is a strong candidate for tasks involving complex image classification. These qualities, coupled with its high accuracy in this study, make it well-suited for detecting and classifying screen defects in returned phones.

4.2.5 Xception

Xception, short for "Extreme Inception," is a convolutional neural network model developed by Google. The Xception architecture is an extension of the Inception architecture, with modifications that allow it to perform even better on image classification tasks.

The architecture of Xception, as depicted in the Figure: 11 (Srinivasan et al., 2021) below, is characterized by the use of depthwise separable convolutions instead of standard convolutions. Depthwise separable convolution is a two-step process involving depthwise convolution, which applies a single filter per input channel, followed by a pointwise convolution, a 1x1 convolution, that creates a linear combination of the output of the depthwise layer.

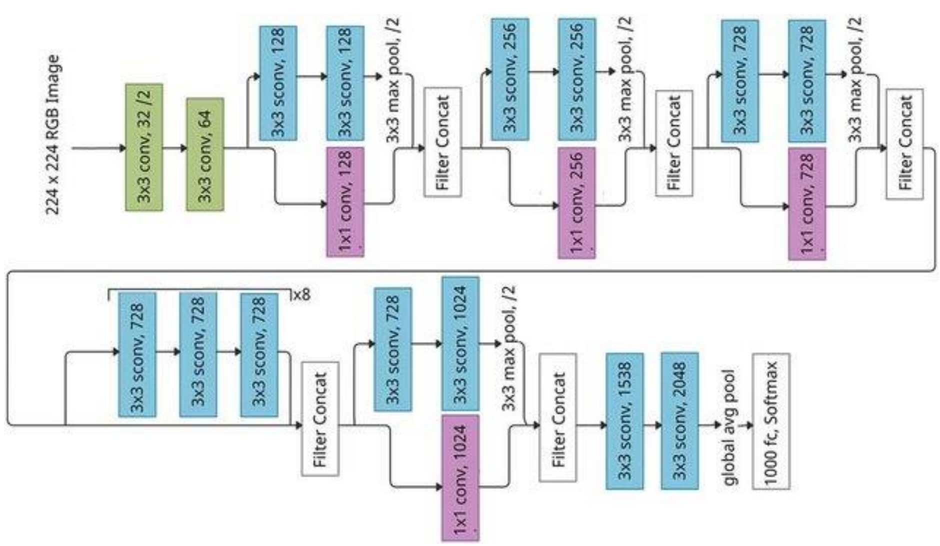


Figure 11 (Srinivasan et al., 2021) : The architecture of Xception model

This approach significantly reduces the model's parameters, making it more computationally efficient, while allowing it to separately map cross-channel correlations and spatial correlations. This can lead to better performance on complex tasks.

In terms of structure, Xception comprises 36 convolutional layers, which form a 'deep middle flow' sandwiched between two single layers at the top and bottom. The deep middle flow consists of eight identical modules, each with three depthwise separable convolution layers.

Given the task of detecting and classifying screen defects in returned phones, Xception's powerful and efficient architecture is an excellent choice. Its strong performance on this task, as indicated by the high accuracy achieved, underscores its capability for complex image classification tasks.

In summary, Xception, with its unique depthwise separable convolutions, efficient architecture, and strong performance, is well-suited for tasks involving complex image classification. Its high accuracy in this study makes it a promising model for detecting and classifying screen defects in returned phones.

4.3 Data Preprocessing

Before the images could be used to train the machine learning models, they underwent a series of preprocessing steps to ensure they were in an appropriate format. These steps are critical in any machine learning task, as they can significantly impact the performance of the models.

4.3.1 Image Resizing

While all images in the dataset were of the same size, they needed to be resized to match the input shape required by the pre-trained models. This resizing step is crucial as these models have specific input size requirements based on their architecture.

In this step, all images were resized to the required dimensions without distorting their aspect ratio. This standardization ensured that the images could be effectively processed by the pre-trained models.

4.3.2 Data Augmentation

Given the imbalance in the dataset, with fewer non-defective screen images, data augmentation techniques were applied to balance the dataset. These techniques, including image transformations like rotation, flipping, zooming, and shifting, were applied to the non-defective images to increase their number.

This process not only balanced the dataset but also added a layer of robustness to the models. By creating a more diverse set of non-defective images, the models were trained to recognize 'good' screens under various conditions.

Following these preprocessing steps, the dataset was ready for use in training and validating the models. The balanced and standardized dataset was instrumental in developing models capable of accurately detecting and classifying screen defects and identifying non-defective screens in returned phones.

4.4 Model Implementation

The study employed a transfer learning approach, utilizing five pre-trained models: DenseNet201, InceptionV3, VGG16, VGG19, and Xception. These models, each having a distinct architecture, were fine-tuned for the task of classifying screen defects and identifying non-defective screens.

4.4.1 Model Architecture

While the architectures of DenseNet201, InceptionV3, VGG16, VGG19, and Xception served as the base, modifications were made to adapt these models to the specific task. The top layers, which are typically fully connected layers, were removed from each of these pre-trained models. The remaining part was flattened to convert the 3D output to 1D, and new layers were added.

These new layers included dense layers with ReLU (Rectified Linear Unit) activation functions. The choice of ReLU was motivated by several considerations. ReLU is computationally efficient, provides non-linearity, enables sparse activation, and helps mitigate the vanishing gradient problem, a common issue in deep learning models. However, I acknowledge that other activation functions like Leaky ReLU, ELU, or SELU could also be suitable depending on the specific requirements and constraints of the task.

Dropout layers were also incorporated into the architecture. Dropout is a regularization method that randomly sets a fraction of input units to 0 at each update during training. This helps prevent over-reliance on specific weights and encourages the model to generalize better.

In addition to dropout, kernel L2 regularizer was used as another form of regularization. L2 regularization discourages large weights by adding a penalty equivalent to the square of the magnitude of the weights to the loss function, thus reducing the complexity of the model.

The final layer in each model was a softmax layer, outputting the probabilities of each class (type of screen damage or non-defective). The combination of these techniques aimed to create a model that could accurately classify screen damage types while also being robust to overfitting.

4.4.2 Training and Validation Split

The dataset was divided into a training set and a validation set, with 80% of the data used for training and 20% used for validation. This ensured the models were evaluated on unseen data, providing a measure of their ability to generalize.

4.4.3 Model Compilation and Training

Once each model was constructed with the new architecture, they were compiled for training. This important step involves defining the optimizer, loss function, and performance metrics. The optimizer directs how the model is updated based on the data it sees and its loss function. The loss function, in this case, categorical cross-entropy, measures how well the model is performing on the data, with lower values being better. The performance metric used was accuracy, which calculates the proportion of correct predictions over total predictions.

The models were then trained on the training set, with the weights updated iteratively to minimize the loss. During this process, the models learned to map the features of the images to their corresponding labels. The performance of the models was evaluated at the end of each training epoch using the validation set. The training process continued until there was no further improvement in the models' validation performance.

In this phase, the models' capabilities to generalize their learning to unseen data and accurately predict the class labels (type of screen damage or non-defective) were honed. The subsequent evaluation of these models in chapter 5 will provide a detailed account of their performance.

4.5 Performance Evaluation Metrics

The methodologies outlined in Chapter 3 laid the foundation for the evaluation metrics used in this study. As described in section 3.2.6, the models' performance was evaluated using several key metrics: Accuracy, Precision, Recall (Sensitivity), and F1 Score. These metrics provide a comprehensive view of the models' performance, considering both the correctness of their predictions and their errors.

Additionally, a confusion matrix was generated to visualize the performance of the models in classifying the images correctly. This matrix provides a detailed view of the models' true positive, true negative, false positive, and false negative predictions.

4.6 Summary

This chapter navigated through the implementation of deep learning models, specifically focusing on the application of five pre-trained architectures: DenseNet201, InceptionV3, VGG16, VGG19, and Xception. Each model, known for its proven effectiveness in image classification tasks, was fine-tuned to suit the specific task of this study: classifying screen defects and identifying non-defective screens.

The chapter started by presenting an overview of each selected model, discussing their unique architectures and how they have been employed in previous research. Following this, the chapter delved into data preprocessing, detailing the necessary steps of image resizing and data augmentation. These steps ensured the image data was in an appropriate and standardized format for the deep learning models to process.

The model implementation section then described the alterations made to the base architectures of the pre-trained models. This involved removing the top layers, flattening the remaining part, and adding new layers, including dense layers with ReLU activation functions and dropout layers. The final layer in each model was a softmax layer, designed to output the probabilities of each class.

The chapter also discussed the division of the dataset into training and validation sets, with 80% of the data used for training and the remaining 20% for validation. This split is crucial to evaluate the models' ability to generalize their learning to unseen data.

The performance evaluation metrics section outlined the key metrics that will be used to assess the models' performance. These include accuracy, precision, recall, and F1 score. Together, these metrics provide a comprehensive and balanced evaluation of the models' performance.

The chapter concluded by emphasizing that the performance of these models, as evaluated by these metrics, will be presented and analyzed in Chapter 5. This will offer a complete picture of the effectiveness of the models and the transfer learning approach used in the study. The chapter thus set the stage for a rigorous and thorough understanding of the research outcomes.

CHAPTER 5: RESULTS AND EVALUATION

5.1 Introduction

This chapter is dedicated to presenting the results and evaluating the performance of the implemented deep learning models, namely DenseNet201, InceptionV3, VGG16, VGG19, and Xception. The models have been employed to tackle a complex task - identifying screen defects and classifying damaged screens. The success of these models in this regard is scrutinized using a set of pre-defined metrics, as outlined in Chapter 4: accuracy, precision, recall, and F1 score.

The ultimate goal of this chapter is to provide an in-depth comprehension of how well these models perform, their comparative effectiveness, and what these findings mean for the broader domain of screen defect identification.

5.2 Model Performance Evaluation

The crux of this chapter lies in the performance evaluation of the models. Each model was put to the test using the validation set, a crucial step to ensure the model's ability to generalize what it has learned to unseen data. The performance of each model was gauged using the metrics defined in Chapter 4: accuracy, precision, recall, and F1 score.

This section delves into a detailed dissection of each model's performance. It provides a meticulous comparison of the models based on the evaluation metrics, thereby offering significant insights into their ability to accurately predict class labels. This evaluation not only helps identify the most effective model for screen defect identification and classification but also sheds light on areas that may need improvement or further investigation.

In the subsequent sections, a comprehensive comparison of the models is provided, followed by a summary of the key findings and their implications.

5.2.1 DenseNet201 Evaluation

The performance of the DenseNet201 model was systematically evaluated using several metrics and visual representations.

The model's training and validation accuracy and loss were tracked throughout each epoch. The final results of this monitoring process are captured in the Figure: 12 below.

```
Epoch 1/20
41/41 [==============================] - ETA: 0s - loss: 1.4280 - accuracy: 0.3140/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:310
  saving_api.save_model(
41/41 [==============================] - 109s 2s/step - loss: 1.4280 - accuracy: 0.3140 - val_loss: 1.3813 - val_accuracy: 0.3549 - lr: 1.0000e-05
Epoch 2/20
41/41 [==============================] - 93s 2s/step - loss: 1.3051 - accuracy: 0.3665 - val_loss: 1.3556 - val_accuracy: 0.2994 - lr: 1.0000e-05
Epoch 3/20
41/41 [==============================] - 95s 2s/step - loss: 1.2702 - accuracy: 0.4159 - val_loss: 1.3051 - val_accuracy: 0.4259 - lr: 1.0000e-05
Epoch 4/20
41/41 [==============================] - 94s 2s/step - loss: 1.2348 - accuracy: 0.4136 - val_loss: 1.3186 - val_accuracy: 0.3981 - lr: 1.0000e-05
Epoch 5/20
41/41 [==============================] - 93s 2s/step - loss: 1.1852 - accuracy: 0.4861 - val_loss: 1.3490 - val_accuracy: 0.4198 - lr: 1.0000e-05
Epoch 6/20
41/41 [==============================] - 95s 2s/step - loss: 1.1513 - accuracy: 0.4977 - val_loss: 1.3058 - val_accuracy: 0.4722 - lr: 1.0000e-05
Epoch 7/20
41/41 [==============================] - 94s 2s/step - loss: 1.1503 - accuracy: 0.4853 - val_loss: 1.2888 - val_accuracy: 0.3611 - lr: 1.0000e-05
Epoch 8/20
41/41 [==============================] - 93s 2s/step - loss: 1.1275 - accuracy: 0.5147 - val_loss: 1.3584 - val_accuracy: 0.3580 - lr: 1.0000e-05
Epoch 9/20
41/41 [==============================] - 95s 2s/step - loss: 1.1012 - accuracy: 0.5347 - val_loss: 1.3113 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 10/20
41/41 [==============================] - 93s 2s/step - loss: 1.0857 - accuracy: 0.5486 - val_loss: 1.2673 - val_accuracy: 0.4815 - lr: 1.0000e-05
Epoch 11/20
41/41 [==============================] - 93s 2s/step - loss: 1.0473 - accuracy: 0.5702 - val_loss: 1.3194 - val_accuracy: 0.4630 - lr: 1.0000e-05
Epoch 12/20
41/41 [==============================] - 93s 2s/step - loss: 1.0395 - accuracy: 0.5718 - val_loss: 1.3422 - val_accuracy: 0.4167 - lr: 1.0000e-05
Epoch 13/20
41/41 [==============================] - 93s 2s/step - loss: 1.0055 - accuracy: 0.5972 - val_loss: 1.3605 - val_accuracy: 0.4846 - lr: 5.0000e-06
Epoch 14/20
41/41 [==============================] - 94s 2s/step - loss: 1.0120 - accuracy: 0.6034 - val_loss: 1.3444 - val_accuracy: 0.3981 - lr: 5.0000e-06
```

Figure 12: DenseNet201 Evaluation Metrics

These results reflect the model's learning progress during training, illustrating how the accuracy improved and loss decreased over time. A graphical representation of the training and validation accuracy and loss over each epoch is provided below in Figure: 13.

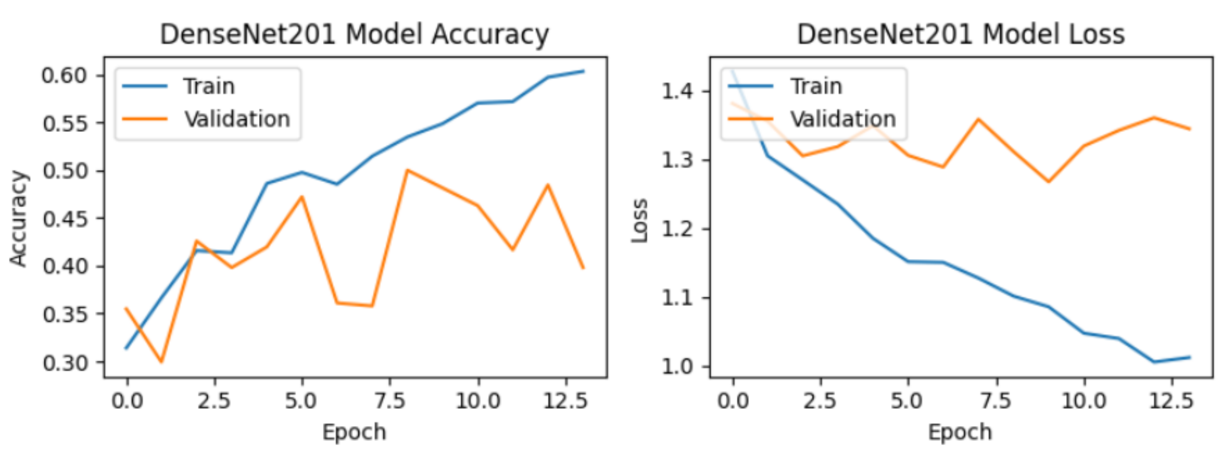


Figure 13: DenseNet201 Training vs Validation Accuracy and Loss Plots

The classification report for DenseNet201 is presented in the below Table 2.

Table 2: DenseNet201 Model Classification Report

Class	Precision	Recall	F1-score	Support
good	0.56	0.68	0.62	84
oil	0.3	0.28	0.29	80
scratch	0.49	0.47	0.48	80
stain	0.51	0.45	0.48	80

The DenseNet201 model, with an overall accuracy of 50.61%, demonstrated varied precision and recall rates across different classes. The confusion matrix, illustrated in Figure 14 below, offers a clear depiction of the model's performance in terms of correctly classifying the images.

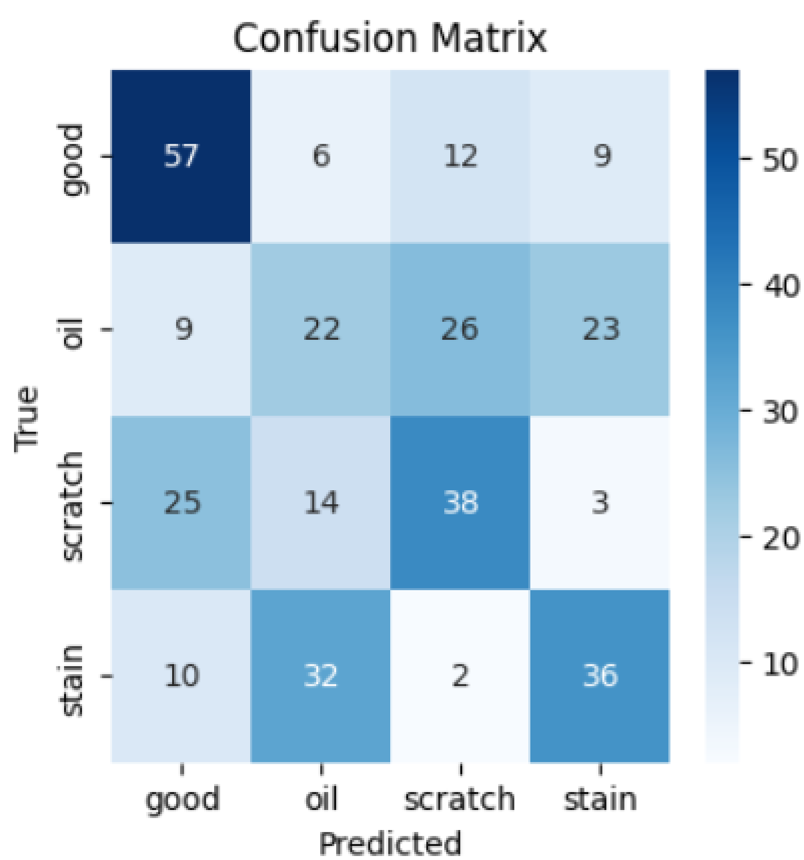


Figure 14: DenseNet201 Model Confusion Matrix

5.2.2 InceptionV3 Evaluation

A thorough evaluation of the InceptionV3 model was carried out, employing a mix of metrics and visual aids. The training and validation accuracy and loss of the model were tracked over the course of each epoch. The final outcomes of this process are depicted in Figure 15 below.

```
Epoch 1/20
41/41 [==============================] - ETA: 0s - loss: 4.1311 - accuracy: 0.3333/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3
  saving_api.save_model(
41/41 [==============================] - 127s 3s/step - loss: 4.1311 - accuracy: 0.3333 - val_loss: 1.3560 - val_accuracy: 0.5000 - lr: 0.0010
Epoch 2/20
41/41 [==============================] - 116s 3s/step - loss: 1.3045 - accuracy: 0.4637 - val_loss: 1.4435 - val_accuracy: 0.4136 - lr: 0.0010
Epoch 3/20
41/41 [==============================] - 117s 3s/step - loss: 1.1883 - accuracy: 0.5185 - val_loss: 1.1643 - val_accuracy: 0.5648 - lr: 0.0010
Epoch 4/20
41/41 [==============================] - 114s 3s/step - loss: 1.0917 - accuracy: 0.5710 - val_loss: 1.4459 - val_accuracy: 0.3920 - lr: 0.0010
Epoch 5/20
41/41 [==============================] - 114s 3s/step - loss: 1.0248 - accuracy: 0.5972 - val_loss: 1.5974 - val_accuracy: 0.4506 - lr: 0.0010
Epoch 6/20
41/41 [==============================] - 115s 3s/step - loss: 1.0664 - accuracy: 0.5864 - val_loss: 1.3675 - val_accuracy: 0.5154 - lr: 0.0010
Epoch 7/20
41/41 [==============================] - 115s 3s/step - loss: 0.8247 - accuracy: 0.6790 - val_loss: 1.0522 - val_accuracy: 0.5463 - lr: 5.0000e-04
Epoch 8/20
41/41 [==============================] - 119s 3s/step - loss: 0.7971 - accuracy: 0.6790 - val_loss: 0.9638 - val_accuracy: 0.5926 - lr: 5.0000e-04
Epoch 9/20
41/41 [==============================] - 116s 3s/step - loss: 0.7296 - accuracy: 0.7199 - val_loss: 0.8728 - val_accuracy: 0.6698 - lr: 5.0000e-04
Epoch 10/20
41/41 [==============================] - 118s 3s/step - loss: 0.6938 - accuracy: 0.7515 - val_loss: 0.9400 - val_accuracy: 0.6636 - lr: 5.0000e-04
Epoch 11/20
41/41 [==============================] - 115s 3s/step - loss: 0.6846 - accuracy: 0.7384 - val_loss: 0.9796 - val_accuracy: 0.6080 - lr: 5.0000e-04
Epoch 12/20
41/41 [==============================] - 115s 3s/step - loss: 0.6994 - accuracy: 0.7392 - val_loss: 0.9200 - val_accuracy: 0.6420 - lr: 5.0000e-04
Epoch 13/20
41/41 [==============================] - 115s 3s/step - loss: 0.6672 - accuracy: 0.7431 - val_loss: 0.9304 - val_accuracy: 0.6235 - lr: 2.5000e-04
Epoch 14/20
41/41 [==============================] - 116s 3s/step - loss: 0.6300 - accuracy: 0.7693 - val_loss: 0.9100 - val_accuracy: 0.6821 - lr: 2.5000e-04
Epoch 15/20
41/41 [==============================] - 114s 3s/step - loss: 0.6200 - accuracy: 0.7724 - val_loss: 0.9260 - val_accuracy: 0.6451 - lr: 2.5000e-04
Epoch 16/20
41/41 [==============================] - 114s 3s/step - loss: 0.6203 - accuracy: 0.7608 - val_loss: 0.9026 - val_accuracy: 0.6759 - lr: 2.5000e-04
Epoch 17/20
41/41 [==============================] - 118s 3s/step - loss: 0.6256 - accuracy: 0.7593 - val_loss: 0.8134 - val_accuracy: 0.6852 - lr: 2.5000e-04
Epoch 18/20
41/41 [==============================] - 114s 3s/step - loss: 0.6190 - accuracy: 0.7778 - val_loss: 0.8573 - val_accuracy: 0.6543 - lr: 2.5000e-04
Epoch 19/20
41/41 [==============================] - 116s 3s/step - loss: 0.6105 - accuracy: 0.7716 - val_loss: 0.9252 - val_accuracy: 0.6698 - lr: 2.5000e-04
Epoch 20/20
41/41 [==============================] - 117s 3s/step - loss: 0.6276 - accuracy: 0.7677 - val_loss: 0.9290 - val_accuracy: 0.6296 - lr: 2.5000e-04
```

Figure 15: InceptionV3 Evaluation Metrics

These figures highlight the progression of the model during the training phase, showcasing the enhancement in accuracy and the reduction in loss over time. A visual representation of the training and validation accuracy and loss over the epochs is given in Figure 16 below.

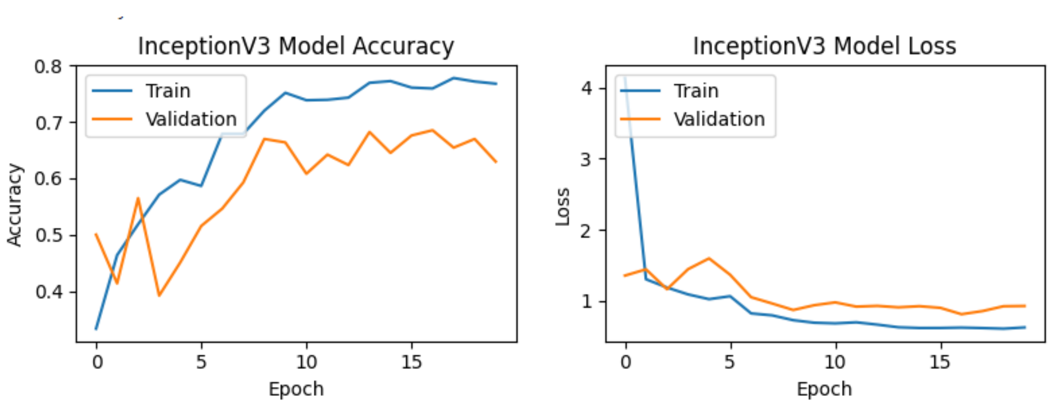


Figure 16: InceptionV3 Training vs Validation Accuracy and Loss Plots

The classification report for InceptionV3 is contained in Table 3 below.

Table 3: InceptionV3 Model Classification Report

Class	Precision	Recall	F1-score	Support
good	0.75	0.71	0.73	84
oil	0.60	0.36	0.45	80
scratch	0.60	0.66	0.63	80
stain	0.62	0.84	0.71	80

The InceptionV3 model, with an overall accuracy of 68.52% exhibited a spectrum of precision and recall rates across various classes. The confusion matrix, displayed in Figure 17 below, provides a clear portrayal of the model's success in accurately classifying the images.

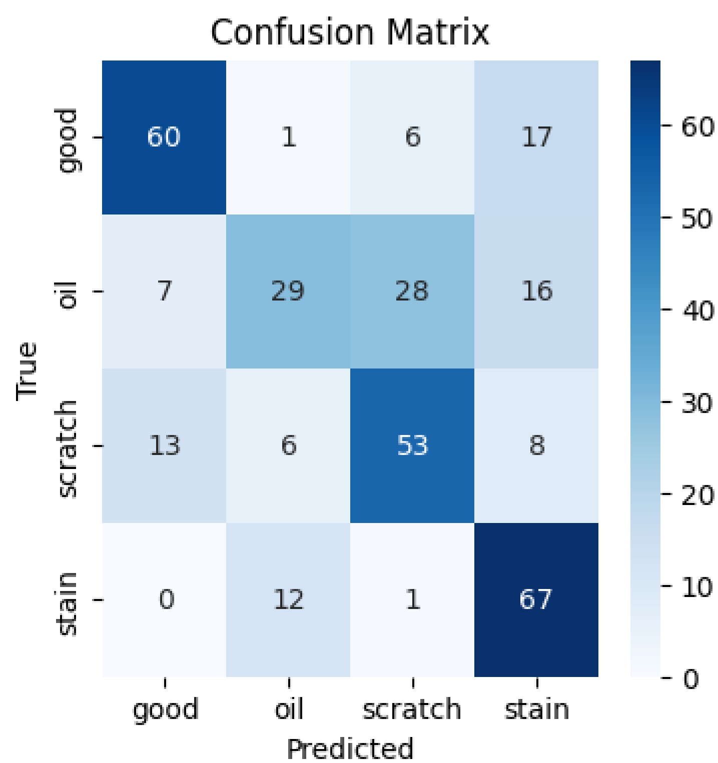


Figure 17: InceptionV3 Model Confusion Matrix

5.2.3 VGG16 Evaluation

An exhaustive assessment of the VGG16 model was conducted, leveraging a range of visual tools and metrics. The model's training and validation accuracy and loss were observed consistently across each epoch. The concluding results from this observation are presented in Figure 18 below.

```
Epoch 1/20
21/21 [==============================] - ETA: 0s - loss: 1.4071 - accuracy: 0.3356/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:
  saving_api.save_model(
21/21 [==============================] - 129s 4s/step - loss: 1.4071 - accuracy: 0.3356 - val_loss: 1.3291 - val_accuracy: 0.4043 - lr: 1.0000e-05
Epoch 2/20
21/21 [==============================] - 95s 5s/step - loss: 1.1378 - accuracy: 0.5934 - val_loss: 1.0287 - val_accuracy: 0.6667 - lr: 1.0000e-05
Epoch 3/20
21/21 [==============================] - 94s 4s/step - loss: 0.6132 - accuracy: 0.8233 - val_loss: 0.4469 - val_accuracy: 0.8395 - lr: 1.0000e-05
Epoch 4/20
21/21 [==============================] - 96s 5s/step - loss: 0.2263 - accuracy: 0.9275 - val_loss: 0.2848 - val_accuracy: 0.9136 - lr: 1.0000e-05
Epoch 5/20
21/21 [==============================] - 94s 4s/step - loss: 0.1741 - accuracy: 0.9475 - val_loss: 0.2908 - val_accuracy: 0.9105 - lr: 1.0000e-05
Epoch 6/20
21/21 [==============================] - 94s 4s/step - loss: 0.1214 - accuracy: 0.9676 - val_loss: 0.2433 - val_accuracy: 0.9198 - lr: 5.0000e-06
Epoch 7/20
21/21 [==============================] - 95s 4s/step - loss: 0.0880 - accuracy: 0.9784 - val_loss: 0.1699 - val_accuracy: 0.9568 - lr: 5.0000e-06
Epoch 8/20
21/21 [==============================] - 95s 5s/step - loss: 0.1122 - accuracy: 0.9653 - val_loss: 0.1951 - val_accuracy: 0.9506 - lr: 5.0000e-06
Epoch 9/20
21/21 [==============================] - 95s 5s/step - loss: 0.0793 - accuracy: 0.9792 - val_loss: 0.1393 - val_accuracy: 0.9599 - lr: 2.5000e-06
Epoch 10/20
21/21 [==============================] - 94s 4s/step - loss: 0.0663 - accuracy: 0.9823 - val_loss: 0.2162 - val_accuracy: 0.9167 - lr: 2.5000e-06
Epoch 11/20
21/21 [==============================] - 94s 4s/step - loss: 0.0556 - accuracy: 0.9892 - val_loss: 0.1514 - val_accuracy: 0.9506 - lr: 1.2500e-06
Epoch 12/20
21/21 [==============================] - 93s 4s/step - loss: 0.0567 - accuracy: 0.9846 - val_loss: 0.2219 - val_accuracy: 0.9321 - lr: 6.2500e-07
Epoch 13/20
21/21 [==============================] - 94s 5s/step - loss: 0.0606 - accuracy: 0.9830 - val_loss: 0.1794 - val_accuracy: 0.9444 - lr: 3.1250e-07
Epoch 14/20
21/21 [==============================] - 94s 4s/step - loss: 0.0629 - accuracy: 0.9807 - val_loss: 0.1959 - val_accuracy: 0.9414 - lr: 1.5625e-07
```

Figure 18: VGG16 Evaluation Metrics

This data underscores the model's advancement during the training phase, highlighting the rise in accuracy and the decrease in loss with time. Figure 19 below offers a visual breakdown of the training and validation accuracy and loss throughout the epochs.

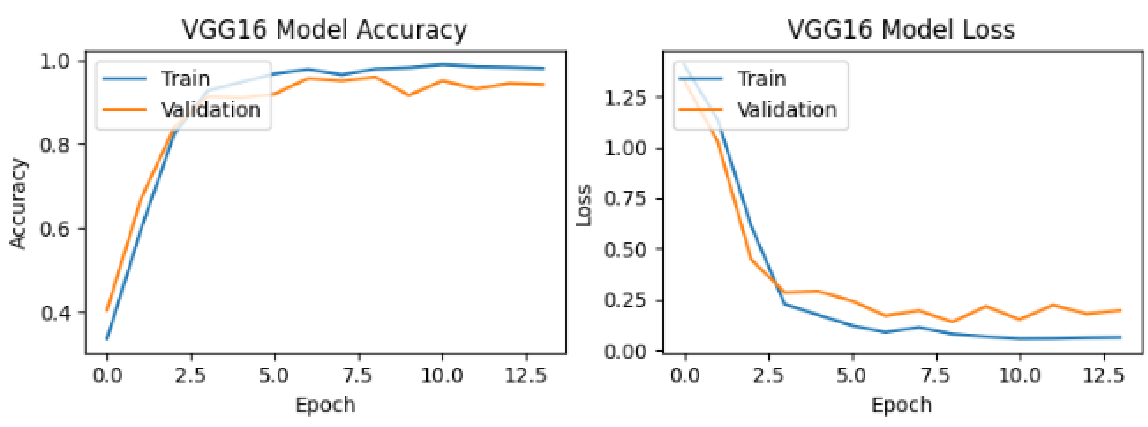


Figure 19: VGG16 Training vs Validation Accuracy and Loss Plots

The classification report for VGG16 is detailed in Table 4 below.

Table 4: VGG16 Model Classification Report

Class	Precision	Recall	F1-score	Support
good	0.93	0.96	0.95	84
oil	0.99	0.96	0.97	80
scratch	0.95	0.91	0.93	80
stain	0.95	0.97	0.96	80

With an overall accuracy of 95.06%, the VGG16 model showed a variety of precision and recall rates across different classes. The model's effectiveness in accurately classifying the images is well represented in the confusion matrix, seen in Figure 20 below.

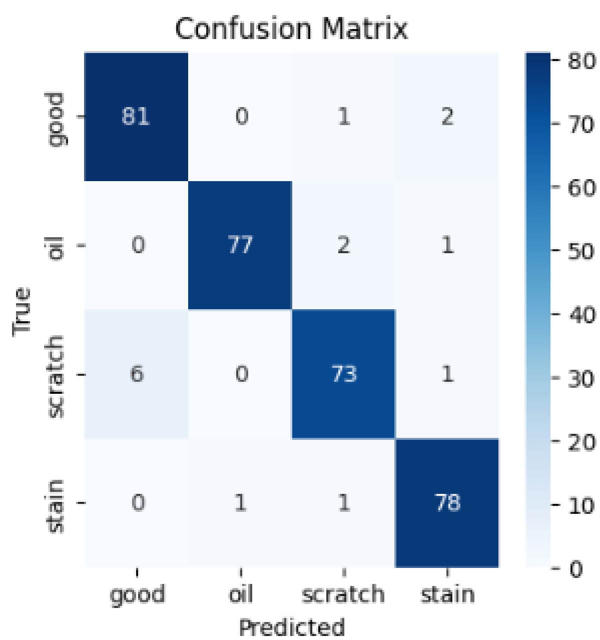


Figure 20: VGG16 Model Confusion Matrix

5.2.4 VGG19 Evaluation

A comprehensive analysis of the VGG19 model was executed, using several metrics and visual representations. The model's training and validation accuracy and loss were systematically tracked throughout each epoch. The ultimate results from this tracking are illustrated in Figure 21 below.

```
Epoch 1/20
21/21 [==============================] - ETA: 0s - loss: 1.3132 - accuracy: 0.4043/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py
  saving_api.save_model(
21/21 [==============================] - 100s 5s/step - loss: 1.3132 - accuracy: 0.4043 - val_loss: 1.2497 - val_accuracy: 0.4815 - lr: 1.0000e-05
Epoch 2/20
21/21 [==============================] - 95s 5s/step - loss: 0.9310 - accuracy: 0.6991 - val_loss: 0.7941 - val_accuracy: 0.6420 - lr: 1.0000e-05
Epoch 3/20
21/21 [==============================] - 96s 5s/step - loss: 0.4237 - accuracy: 0.8681 - val_loss: 0.3425 - val_accuracy: 0.8765 - lr: 1.0000e-05
Epoch 4/20
21/21 [==============================] - 95s 5s/step - loss: 0.2378 - accuracy: 0.9267 - val_loss: 0.2299 - val_accuracy: 0.9167 - lr: 1.0000e-05
Epoch 5/20
21/21 [==============================] - 95s 4s/step - loss: 0.1636 - accuracy: 0.9545 - val_loss: 0.1644 - val_accuracy: 0.9444 - lr: 1.0000e-05
Epoch 6/20
21/21 [==============================] - 95s 5s/step - loss: 0.1173 - accuracy: 0.9653 - val_loss: 0.1890 - val_accuracy: 0.9383 - lr: 1.0000e-05
Epoch 7/20
21/21 [==============================] - 95s 4s/step - loss: 0.0929 - accuracy: 0.9707 - val_loss: 0.1449 - val_accuracy: 0.9599 - lr: 5.0000e-06
Epoch 8/20
21/21 [==============================] - 95s 4s/step - loss: 0.0678 - accuracy: 0.9838 - val_loss: 0.1316 - val_accuracy: 0.9599 - lr: 5.0000e-06
Epoch 9/20
21/21 [==============================] - 94s 4s/step - loss: 0.0716 - accuracy: 0.9815 - val_loss: 0.1693 - val_accuracy: 0.9444 - lr: 2.5000e-06
Epoch 10/20
21/21 [==============================] - 94s 4s/step - loss: 0.0508 - accuracy: 0.9838 - val_loss: 0.1667 - val_accuracy: 0.9444 - lr: 1.2500e-06
Epoch 11/20
21/21 [==============================] - 94s 4s/step - loss: 0.0564 - accuracy: 0.9877 - val_loss: 0.1265 - val_accuracy: 0.9537 - lr: 6.2500e-07
Epoch 12/20
21/21 [==============================] - 94s 4s/step - loss: 0.0564 - accuracy: 0.9853 - val_loss: 0.1731 - val_accuracy: 0.9444 - lr: 3.1250e-07
```

Figure 21: VGG19 Evaluation Metrics

These statistics depict the model's progression during the training phase, emphasizing the growth in accuracy and the fall in loss over time. A graphical depiction of the training and validation accuracy and loss over the epochs is provided in Figure 22 below.

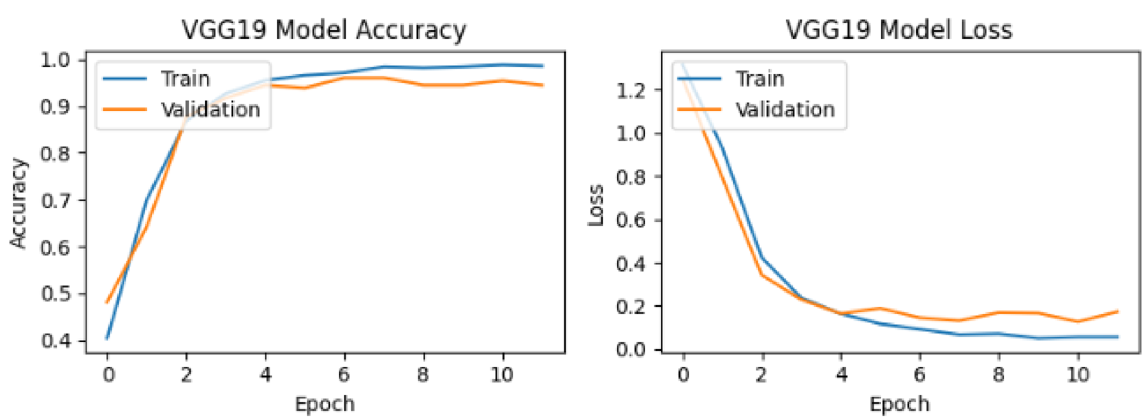


Figure 22: VGG19 Training vs Validation Accuracy and Loss Plots

The classification report for VGG19 is set out in Table 5 below.

Table 5: VGG19 Model Classification Report

Class	Precision	Recall	F1-score	Support
good	0.94	0.94	0.94	84
oil	0.97	0.94	0.96	80
scratch	0.93	0.95	0.94	80
stain	0.93	0.94	0.93	80

The VGG19 model, with an overall accuracy of 94.16%, displayed a range of precision and recall rates across the classes. The model's capacity to accurately classify the images is clearly depicted in the confusion matrix, as shown in Figure 23 below.

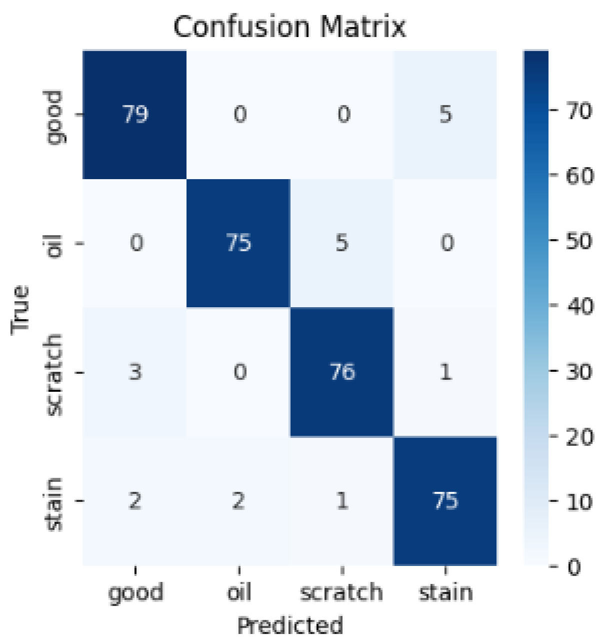
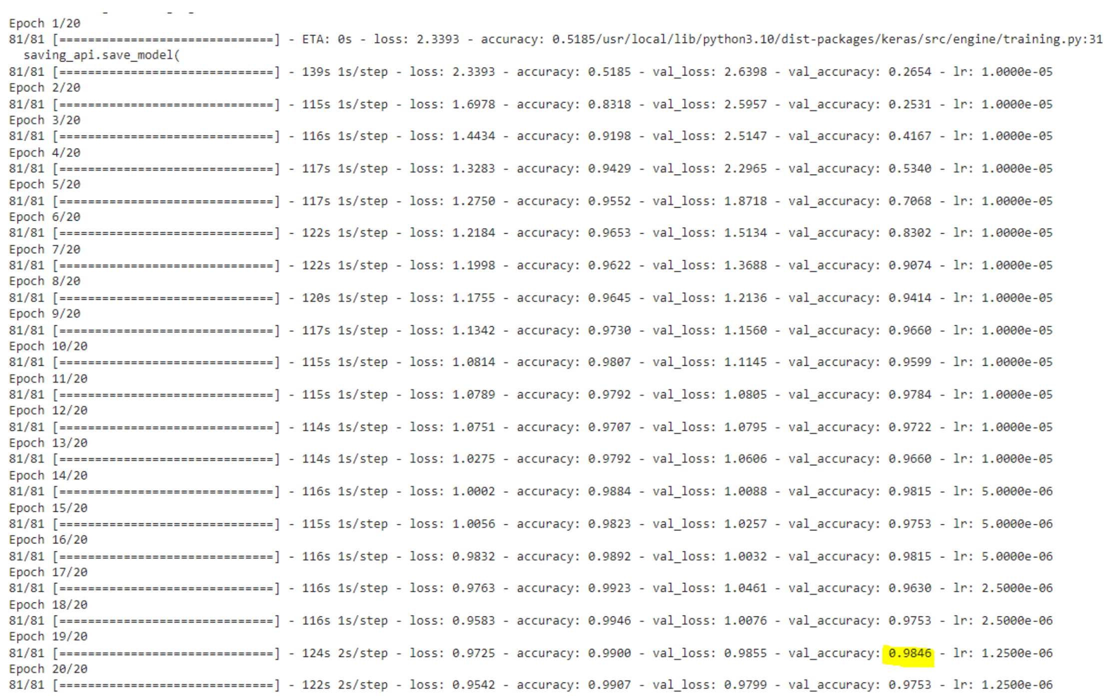


Figure 23: VGG19 Model Confusion Matrix

5.2.5 Xception Evaluation

A detailed examination of the Xception model was performed, utilizing an array of metrics and visual aids. The model's training and validation accuracy and loss were monitored over each epoch. The final results from this monitoring are shown in Figure 24 below.



```
Epoch 1/20
81/81 [==============================] - ETA: 0s - loss: 2.3393 - accuracy: 0.5185/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:31
  saving_api.save_model(
81/81 [==============================] - 139s 1s/step - loss: 2.3393 - accuracy: 0.5185 - val_loss: 2.6398 - val_accuracy: 0.2654 - lr: 1.0000e-05
Epoch 2/20
81/81 [==============================] - 115s 1s/step - loss: 1.6978 - accuracy: 0.8318 - val_loss: 2.5957 - val_accuracy: 0.2531 - lr: 1.0000e-05
Epoch 3/20
81/81 [==============================] - 116s 1s/step - loss: 1.4434 - accuracy: 0.9198 - val_loss: 2.5147 - val_accuracy: 0.4167 - lr: 1.0000e-05
Epoch 4/20
81/81 [==============================] - 117s 1s/step - loss: 1.3283 - accuracy: 0.9429 - val_loss: 2.2965 - val_accuracy: 0.5340 - lr: 1.0000e-05
Epoch 5/20
81/81 [==============================] - 117s 1s/step - loss: 1.2750 - accuracy: 0.9552 - val_loss: 1.8718 - val_accuracy: 0.7068 - lr: 1.0000e-05
Epoch 6/20
81/81 [==============================] - 122s 1s/step - loss: 1.2184 - accuracy: 0.9653 - val_loss: 1.5134 - val_accuracy: 0.8302 - lr: 1.0000e-05
Epoch 7/20
81/81 [==============================] - 122s 1s/step - loss: 1.1998 - accuracy: 0.9622 - val_loss: 1.3688 - val_accuracy: 0.9074 - lr: 1.0000e-05
Epoch 8/20
81/81 [==============================] - 120s 1s/step - loss: 1.1755 - accuracy: 0.9645 - val_loss: 1.2136 - val_accuracy: 0.9414 - lr: 1.0000e-05
Epoch 9/20
81/81 [==============================] - 117s 1s/step - loss: 1.1342 - accuracy: 0.9730 - val_loss: 1.1560 - val_accuracy: 0.9660 - lr: 1.0000e-05
Epoch 10/20
81/81 [==============================] - 115s 1s/step - loss: 1.0814 - accuracy: 0.9807 - val_loss: 1.1145 - val_accuracy: 0.9599 - lr: 1.0000e-05
Epoch 11/20
81/81 [==============================] - 115s 1s/step - loss: 1.0789 - accuracy: 0.9792 - val_loss: 1.0805 - val_accuracy: 0.9784 - lr: 1.0000e-05
Epoch 12/20
81/81 [==============================] - 114s 1s/step - loss: 1.0751 - accuracy: 0.9707 - val_loss: 1.0795 - val_accuracy: 0.9722 - lr: 1.0000e-05
Epoch 13/20
81/81 [==============================] - 114s 1s/step - loss: 1.0275 - accuracy: 0.9792 - val_loss: 1.0606 - val_accuracy: 0.9660 - lr: 1.0000e-05
Epoch 14/20
81/81 [==============================] - 116s 1s/step - loss: 1.0002 - accuracy: 0.9884 - val_loss: 1.0088 - val_accuracy: 0.9815 - lr: 5.0000e-06
Epoch 15/20
81/81 [==============================] - 115s 1s/step - loss: 1.0056 - accuracy: 0.9823 - val_loss: 1.0257 - val_accuracy: 0.9753 - lr: 5.0000e-06
Epoch 16/20
81/81 [==============================] - 116s 1s/step - loss: 0.9832 - accuracy: 0.9892 - val_loss: 1.0032 - val_accuracy: 0.9815 - lr: 5.0000e-06
Epoch 17/20
81/81 [==============================] - 116s 1s/step - loss: 0.9763 - accuracy: 0.9923 - val_loss: 1.0461 - val_accuracy: 0.9630 - lr: 2.5000e-06
Epoch 18/20
81/81 [==============================] - 116s 1s/step - loss: 0.9583 - accuracy: 0.9946 - val_loss: 1.0076 - val_accuracy: 0.9753 - lr: 2.5000e-06
Epoch 19/20
81/81 [==============================] - 124s 2s/step - loss: 0.9725 - accuracy: 0.9900 - val_loss: 0.9855 - val_accuracy: 0.9846 - lr: 1.2500e-06
Epoch 20/20
81/81 [==============================] - 122s 2s/step - loss: 0.9542 - accuracy: 0.9907 - val_loss: 0.9799 - val_accuracy: 0.9753 - lr: 1.2500e-06
```

Figure 24: Xception Evaluation Metrics

These results reveal the model's improvement during the training phase, displaying the increase in accuracy and the reduction in loss over time. A visual illustration of the training and validation accuracy and loss over the epochs is given in Figure 25 below.

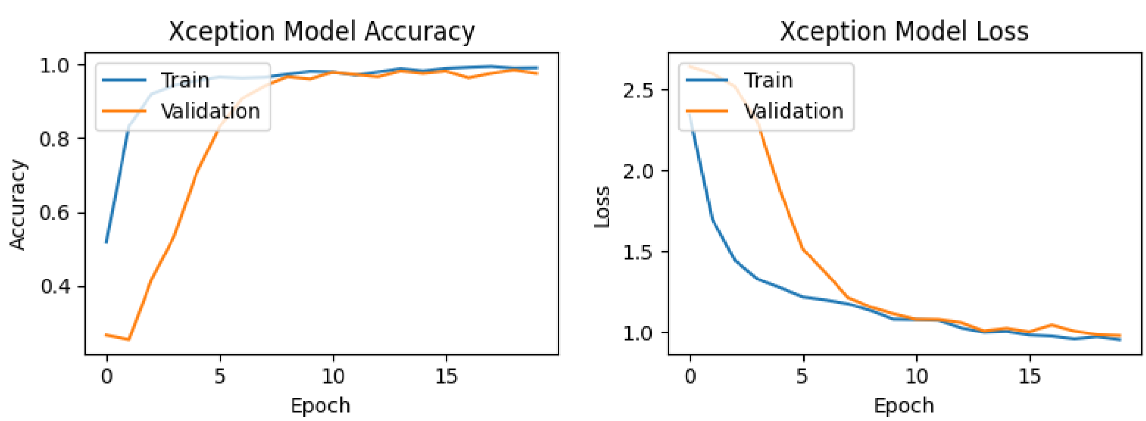


Figure 25: Xception Training vs Validation Accuracy and Loss Plots

The classification report for Xception is displayed in Table 6 below.

Table 6: Xception Model Classification Report

Class	Precision	Recall	F1-score	Support
good	0.98	0.94	0.96	84
oil	0.99	0.99	0.99	80
scratch	0.94	0.99	0.96	80
stain	0.96	0.95	0.96	80

The Xception model, with an overall accuracy of 98.45%, demonstrated various precision and recall rates across different classes. The model's proficiency in accurately classifying the images is clearly illustrated in the confusion matrix, presented in Figure 26 below.

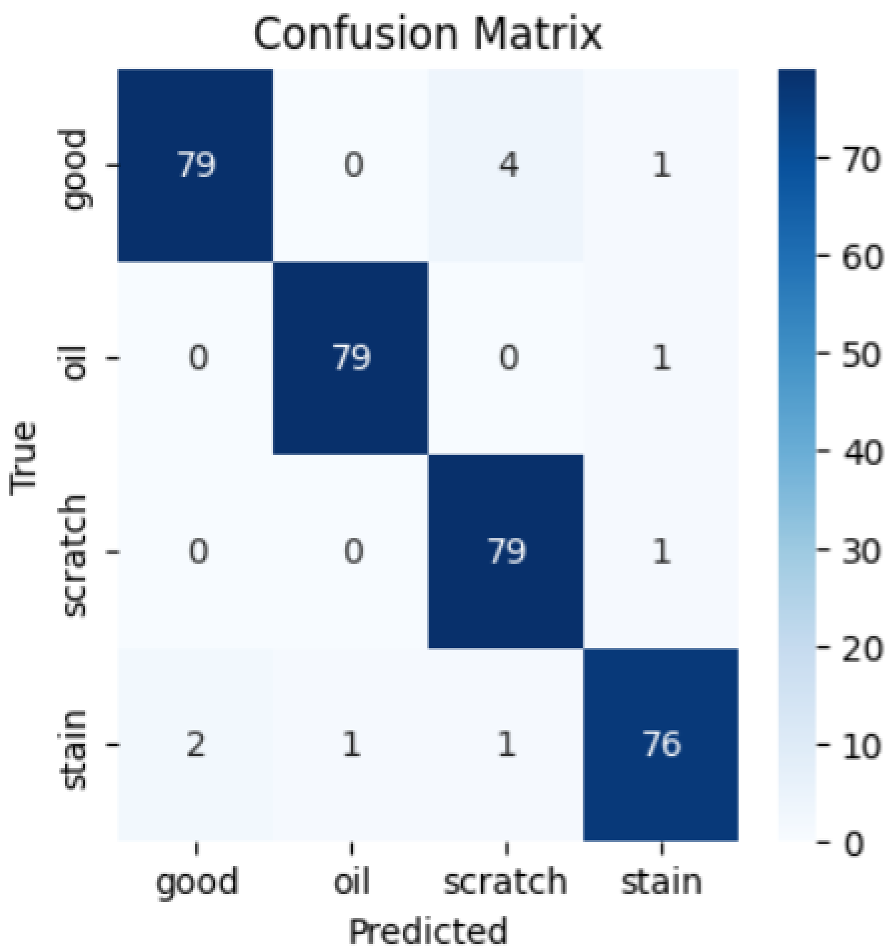


Figure 26: Xception Model Confusion Matrix

5.3 Comparative Analysis of Models

In this section, a comparative analysis is performed on the DenseNet201, InceptionV3, VGG16, VGG19, and Xception models based on their performance metrics, and classification capabilities. Each model's performance was measured using various metrics, including accuracy, class-wise precision, recall, and F1-score. The results are summarized in the below Table 7.

Table 7: Summary of Evaluation Metrics of all Models

Model	Accuracy	Class	Precision	Recall	F1-score	Support
DenseNet201	50.61%	good	0.56	0.68	0.62	84
		oil	0.3	0.28	0.29	80
		scratch	0.49	0.47	0.48	80
		stain	0.51	0.45	0.48	80
InceptionV3	68.52%	good	0.75	0.71	0.73	84
		oil	0.6	0.36	0.45	80
		scratch	0.6	0.66	0.63	80
		stain	0.62	0.84	0.71	80
VGG16	95.06%	good	0.93	0.96	0.95	84
		oil	0.99	0.96	0.97	80
		scratch	0.95	0.91	0.93	80
		stain	0.95	0.97	0.96	80
VGG19	94.16%	good	0.94	0.94	0.94	84
		oil	0.97	0.94	0.96	80
		scratch	0.93	0.95	0.94	80
		stain	0.93	0.94	0.93	80
Xception	98.45%	good	0.98	0.94	0.96	84
		oil	0.99	0.99	0.99	80
		scratch	0.94	0.99	0.96	80
		stain	0.96	0.95	0.96	80

Upon comparing these results, it becomes evident that the Xception model, with the highest accuracy of 98.45%, outperformed the other models in terms of accuracy.

When considering class-wise precision, recall, and F1-score, the Xception model again proved to be the most effective, demonstrating its superior ability to classify images correctly across all classes.

Finally, a review of the confusion matrices revealed that the Xception model also excelled in minimizing misclassifications, further attesting to its robust performance in image classification tasks.

In conclusion, the Xception model emerged as the top performer in this comparative analysis, showcasing its effectiveness in both overall and class-wise accuracy, precision, recall, and F1-score.

5.5 Summary

This chapter conducted a thorough evaluation and comparison of the DenseNet201, InceptionV3, VGG16, VGG19, and Xception models. Each model was critically assessed on various performance metrics, providing a detailed understanding of their respective capabilities.

The Xception model emerged as the standout performer in this analysis, excelling in terms of accuracy, precision, recall, and F1-score. This superior performance makes it a strong candidate for future applications in image classification tasks.

However, it's important to remember that the optimal model can vary depending on the specific requirements and constraints of each application. While the Xception model was the superior performer in this study, other models may be more suitable in different contexts.

This extensive analysis has provided valuable insights into the performance of different deep learning models in image classification tasks.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Introduction

This chapter serves as the culmination of the research journey, summarizing the key findings, implications, and contributions of the study. It revisits the initial research questions, providing a detailed correlation between the set objectives and the achieved outcomes. This section further offers a comprehensive review of the findings from the evaluation of the deep learning models and reflects on the overall contributions made to the field of screen defect identification. Finally, it outlines potential avenues for future exploration and research, setting the stage for subsequent investigations in the domain.

6.2 Discussion and Conclusion

This research embarked on an exploration to answer three crucial research questions concerning the application of transfer learning, data augmentation, and the combination of regularization and dropout methods in enhancing mobile screen defect detection and classification.

The study conducted a thorough testing and evaluation of five distinct deep learning models, namely DenseNet201, InceptionV3, VGG16, VGG19 and Xception. The results revealed that the Xception model, with an accuracy of 98.45%, outperformed the others. This finding provides strong evidence that transfer learning can significantly enhance the accuracy of mobile screen defect detection and classification.

Importantly, this research extends upon the work of (Selvi et al., 2021) by further examining the application of Convolutional Neural Networks (CNNs) for the detection, analysis, and estimation of damage in mobile phone screens. While (Selvi et al., 2021) achieved an accuracy of 85% using pre-trained models such as Densenet, Resnet, and Squeezenet, as well as custom models for damage classification in mobile glass, they noted potential improvements by considering the complex constraints and assumptions of deep learning algorithms. This research has addressed this by exploring more sophisticated models and techniques to enhance accuracy and performance, which has resulted in a significant improvement in accuracy.

Moreover, the research emphasized the effectiveness of data augmentation techniques in broadening the range of damage types that the system can detect, thereby increasing the system's versatility and robustness. This aligns with the study by (Mahyoub et al., 2023) which successfully employed data augmentation to improve class balance and model performance in car damage detection.

In addition, the study showcased the importance of combining regularization and dropout methods in enhancing the model's robustness against overfitting, which is critical for its performance in real-world applications. This finding resonates with the research by (Farhadi et al., 2022), which found that a model combining dropout and elastic net regularization outperformed others in mitigating overfitting.

In conclusion, this research has made significant strides in improving mobile screen defect detection and classification through the application of transfer learning, data augmentation, and regularization and dropout methods. The study's findings not only contribute to the existing body of literature but also provide practical insights that can be leveraged to enhance quality control in the mobile phone industry. Looking forward, there is still potential for further research in this field, particularly in exploring other deep learning models and techniques that could further enhance the accuracy and robustness of mobile screen defect detection and classification.

6.3 Contributions

This research has made several significant contributions to the field of screen defect identification and classification. It has not only established a robust comparison framework for different deep learning models but also illuminated their capabilities and performance characteristics in the specific context of screen defect identification.

Furthermore, the study demonstrated the real-world applicability of these models and highlighted the benefits of techniques like data augmentation, transfer learning, and regularization in enhancing model performance and robustness. The research thus contributes valuable insights and tools that can be used in both academic and industrial settings for screen defect identification.

6.4 Future Work

While this study has made significant strides in leveraging deep learning for screen defect identification, there is still room for further exploration. Future research could involve investigating other deep learning architectures, experimenting with larger and more diverse datasets, or exploring ensemble methods.

One promising avenue for future work could be to extend the classification task to not only identify the type of screen damage but also to classify the intensity or severity of the damage for each class. This could provide more nuanced and detailed information that could be valuable for downstream tasks such as repair estimation.

Further work could also focus on developing more advanced data augmentation techniques or other strategies to enhance the models' learning capabilities and robustness.

6.5 Summary

In conclusion, this chapter has provided a comprehensive summary of the findings of the research, discussing the application of transfer learning, data augmentation, and regularization in improving the accuracy of mobile screen defect detection and classification. It highlighted the performance of the Xception model and its accuracy in the classification task. The chapter also discussed the significant contributions of the research to the field of screen defect identification and outlined potential avenues for future work. This includes the investigation of other deep learning architectures, the use of larger and more diverse datasets, exploring ensemble methods, and extending the classification task to include the intensity of screen damage.

References

1. Alarcón, F., Cortés-Pellicer, P., Pérez-Perales, D., & Mengual-Recuerda, A. (2021). A reference model of reverse logistics process for improving sustainability in the supply chain. *Sustainability (Switzerland)*, *13*(18). https://doi.org/10.3390/su131810383
2. Ali, L., Alnajjar, F., Jassmi, H. Al, Gochoo, M., Khan, W., & Serhani, M. A. (2021). Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors*, *21*(5), 1–22. https://doi.org/10.3390/s21051688
3. Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2020). *Transfer Learning-based Road Damage Detection for Multiple Countries*. http://arxiv.org/abs/2008.13101
4. Arya, D., Maeda, H., Kumar Ghosh, S., Toshniwal, D., Omata, H., Kashiyama, T., & Sekimoto, Y. (2020). Global Road Damage Detection: State-of-the-art Solutions. *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, 5533–5539. https://doi.org/10.1109/BigData50022.2020.9377790
5. Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabbouj, M., & Inman, D. J. (2021). A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. In *Mechanical Systems and Signal Processing* (Vol. 147). Academic Press. https://doi.org/10.1016/j.ymssp.2020.107077
6. Bravo, R., Segura, M. G., Temowo, O., & Samaddar, S. (2022). How Does a Pandemic Disrupt the Benefits of eCommerce? A Case Study of Small and Medium Enterprises in the US. *Journal of Theoretical and Applied Electronic Commerce Research*, *17*(2), 522–557. https://doi.org/10.3390/jtaer17020028
7. Charisma, R. A., & Adhinata, F. D. (2023). Transfer Learning with Densenet201 Architecture Model for Potato Leaf Disease Classification. *ICCoSITE 2023 - International Conference on Computer Science, Information Technology and Engineering: Digital Transformation Strategy in Facing the VUCA and TUNA Era*, 738–743. https://doi.org/10.1109/ICCoSITE57641.2023.10127772
8. Chen, Z., Wang, D., Wang, Y., Lin, S., Jia, H., Lin, P., Liu, Y., & Chen, L. (2023). Research and Implementation of Road Damage Detection Algorithm Based on Object Detection Network. *2023 4th International Seminar on Artificial Intelligence, Networking and*

Information Technology, AINIT 2023, 446–450. https://doi.org/10.1109/AINIT59027.2023.10212930

9. Dyk, T., & Drahansky, M. (2023). Tank Barrel Surface Damage Detection Algorithm. *2023 9th International Conference on Military Technologies, ICMT 2023 - Proceedings*. https://doi.org/10.1109/ICMT58149.2023.10171286
10. Farhadi, Z., Bevrani, H., & Feizi-Derakhshi, M. R. (2022). Combining Regularization and Dropout Techniques for Deep Convolutional Neural Network. *IEEE Global Energy Conference, GEC 2022*, 335–339. https://doi.org/10.1109/GEC55014.2022.9986657
11. Feng, C., Zhang, H., Wang, S., Li, Y., Wang, H., & Yan, F. (2019). Structural Damage Detection using Deep Convolutional Neural Network and Transfer Learning. *KSCE Journal of Civil Engineering*, *23*(10), 4493–4502. https://doi.org/10.1007/s12205-019-0437-z
12. Ferreira, B., & Reis, J. (2023). A Systematic Literature Review on the Application of Automation in Logistics. *Logistics*, *7*(4), 80. https://doi.org/10.3390/logistics7040080
13. Genchev, S. E., Glenn Richey, R., & Gabler, C. B. (2011). Evaluating reverse logistics programs: A suggested process formalization. *The International Journal of Logistics Management*, *22*(2), 242–263. https://doi.org/10.1108/09574091111156578
14. Gur, O. (n.d.). *Aircraft Damage Detection Using Metamodeling Techniques*. https://www.researchgate.net/publication/339942342
15. Irungu, J., Oladunni, T., Denis, M., Ososanya, E., & Muriithi, R. (2023). A CNN Transfer Learning -Electrocardiogram (ECG) Signal Approach to Predict COVID-19. *2023 15th International Conference on Computer and Automation Engineering, ICCAE 2023*, 367–371. https://doi.org/10.1109/ICCAE56788.2023.10111114
16. Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V., & Kaur, M. (2021). Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*, *39*(15), 5682–5689. https://doi.org/10.1080/07391102.2020.1788642
17. Janse, B., Schuur, P., & De Brito, M. P. (2010). A reverse logistics diagnostic tool: The case of the consumer electronics industry. *International Journal of Advanced Manufacturing Technology*, *47*(5–8), 495–513. https://doi.org/10.1007/s00170-009-2333-z

18. Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). *Supervised Machine Learning: A Brief Primer*. www.elsevier.com/locate/bt
19. Kaur Chatrath, S., Batra, G. S., & Chaba, Y. (2022). Handling consumer vulnerability in e-commerce product images using machine learning. *Heliyon*, *8*(9). https://doi.org/10.1016/j.heliyon.2022.e10743
20. Khan, M. H. M., Hussein Sk Heerah, M. Z., & Basgeeth, Z. (2021, October 7). Automated detection of multi-class vehicle exterior damages using deep learning. *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2021*. https://doi.org/10.1109/ICECCME52200.2021.9590927
21. Kyu, P. M., & Woraratpanya, K. (2020, July 1). Car Damage Detection and Classification. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3406601.3406651
22. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. https://doi.org/10.1038/nature14539
23. Li, M., Zhou, A. W., & Ai, B. (2021). An improved FCM algorithm for glass defect extraction of mobile phone screen cover plate. *2021 International Conference on Electronic Information Engineering and Computer Science, EIECS 2021*, 501–504. https://doi.org/10.1109/EIECS53707.2021.9588010
24. Mahyoub, M., Natalia, F., Sudirman, S., Liatsis, P., & Jasim Al-Jumaily, A. H. (2023). Data Augmentation Using Generative Adversarial Networks to Reduce Data Imbalance with Application in Car Damage Detection. *Proceedings - International Conference on Developments in ESystems Engineering, DeSE*, *2023-January*, 480–485. https://doi.org/10.1109/DeSE58274.2023.10100274
25. Mangiaracina, R., Marchet, G., Perotti, S., & Tumino, A. (2015). A review of the environmental implications of B2C e-commerce: a logistics perspective. *International Journal of Physical Distribution and Logistics Management*, *45*(6), 565–591. https://doi.org/10.1108/IJPDLM-06-2014-0133
26. Matt, D. T., Modrák, V., & Zsifkovits, H. (2020). Industry 4.0 for smes: Challenges, opportunities and requirements. In *Industry 4.0 for SMEs: Challenges, Opportunities and Requirements*. Palgrave Macmillan. https://doi.org/10.1007/978-3-030-25425-4

27. Movsessian, A., García Cava, D., & Tcherniak, D. (2021). An artificial neural network methodology for damage detection: Demonstration on an operating wind turbine blade. *Mechanical Systems and Signal Processing*, *159*. https://doi.org/10.1016/j.ymssp.2021.107766
28. Pham, V., Nguyen, D., & Donan, C. (2022). Road Damage Detection and Classification with YOLOv7. *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, 6416–6423. https://doi.org/10.1109/BigData55660.2022.10020856
29. Pham, V., Pham, C., & Dang, T. (2020). Road Damage Detection and Classification with Detectron2 and Faster R-CNN. *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, 5592–5601. https://doi.org/10.1109/BigData50022.2020.9378027
30. Richards, & Gwynne. (n.d.). *Warehouse Management i*.
31. Schlüter, M., Lickert, H., Schweitzer, K., Bilge, P., Briese, C., Dietrich, F., & Krüger, J. (2021). AI-enhanced Identification, Inspection and Sorting for Reverse Logistics in Remanufacturing. *Procedia CIRP*, *98*, 300–305. https://doi.org/10.1016/j.procir.2021.01.107
32. Selvi, K. T., Thamilselvan, R., Pratheksha, K., Praveena, N., & Ragunanthan, S. (2021). Glass Damage Classification in Mobile Phones using Deep Learning Techniques. *Proceedings of the 5th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2021*, 1111–1119. https://doi.org/10.1109/ICECA52323.2021.9676076
33. Shihavuddin, A. S. M., Rashid, M. R. A., Maruf, M. H., Hasan, M. A., Haq, M. A. ul, Ashique, R. H., & Mansur, A. Al. (2021). Image based surface damage detection of renewable energy installations using a unified deep learning approach. *Energy Reports*, *7*, 4566–4576. https://doi.org/10.1016/j.egyr.2021.07.045
34. Srinivasan, K., Garg, L., Datta, D., Alaboudi, A. A., Jhanjhi, N. Z., Agarwal, R., & Thomas, A. G. (2021). Performance comparison of deep cnn models for detecting driver's distraction. *Computers, Materials and Continua*, *68*(3), 4109–4124. https://doi.org/10.32604/cmc.2021.016736
35. Sruthy, C. M., Kunjumon, S., & Nandakumar, R. (2021). Car Damage Identification and Categorization Using Various Transfer Learning Models. *Proceedings of the 5th*

International Conference on Trends in Electronics and Informatics, ICOEI 2021, 1097–1101. https://doi.org/10.1109/ICOEI51242.2021.9452846

36. Thomas, J. S., Ejaz, S., Ahmed, Z., & Hans, S. (2023). Optimized Car Damaged Detection using CNN and Object Detection Model. *Proceedings of 3rd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2023*, 172–174. https://doi.org/10.1109/ICCIKE58312.2023.10131804
37. Widjojo, D., Setyati, E., & Kristian, Y. (2022). Integrated Deep Learning System for Car Damage Detection and Classification Using Deep Transfer Learning. *Proceeding - IEEE 8th Information Technology International Seminar, ITIS 2022*, 21–26. https://doi.org/10.1109/ITIS57155.2022.10010292
38. Xie, Y., Ebad Sichani, M., Padgett, J. E., & DesRoches, R. (2020). The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra*, *36*(4), 1769–1801. https://doi.org/10.1177/8755293020919419
39. Yongfa Lv, Ling Ma, & Huiqin Jiang. (n.d.). *A_Mobile_Phone_Screen_Cover_Glass_Defect_Detection_MODEL_Based_on_Small_Samples_Learning*.
40. Zhang, C., Chang, C. chen, & Jamshidi, M. (2020). Concrete bridge surface damage detection using a single-stage detector. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 389–409. https://doi.org/10.1111/mice.12500

APPENDIX A: RESEARCH PROPOSAL

Abstract

This research proposal addresses the imperative of optimizing reverse logistics for warehouses catering to mobile carrier customers/vendors. through the development of an advanced mobile screen damage detection and classification system. Reverse logistics, encompassing the intricate processes of managing returned mobile phones, faces challenges amplified by the exponential growth of e-commerce and evolving environmental regulations. Specifically, the prevalent issue of glass damage incurs significant costs and delays within the supply chain, necessitating innovative solutions.

My primary objective is to design and implement a scalable, robust, and diversified system capable of identifying and classifying three specific mobile screen damage types: oil, scratch, and stain. To achieve this, the research adopts a comprehensive workflow spanning dataset acquisition, preprocessing, transformation/augmentation, modeling, evaluation, and result interpretation. Transfer learning, data augmentation, and regularization/dropout methods constitute pivotal components of my methodology, enhancing the system's adaptability and performance.

In the context of reverse logistics, the proposed system emerges as a transformative solution to the challenges associated with manual inspection and processing of returned items. By focusing on specific damage types, system not only addresses immediate industry concerns but also aligns with the broader global push towards sustainable supply chains. The integration of advanced machine learning techniques, namely transfer learning and data augmentation, contributes to the efficiency of reverse logistics practices, surpassing traditional manual methods.

In summary, this research underscores the significance of adopting advanced technological solutions to enhance the efficiency of reverse logistics in the mobile industry. The proposed system's scalability, adaptability, and focus on specific damage types position it as a model for sustainable and efficient supply chain practices. Beyond the mobile industry, the potential impact of this research extends to diverse sectors engaged in reverse logistics, contributing to the evolution of artificial intelligence applications in supply chain management.

1. Background

Consumer behavior has undergone a paradigm shift due to the rapid expansion of e-commerce and an increased focus on environmental sustainability, prompting businesses to reassess their strategies in reverse logistics (Richards and Gwynne, n.d., 2014). This shift is particularly evident in the reverse logistics operations of warehouses catering to mobile carrier customers/vendors, where the effective management of returned mobile phones poses intricate challenges. The traditional approach to reverse logistics involves manual inspections, repairs, and recycling procedures within warehouses, leading to resource-intensive processes that occupy substantial space and require a significant workforce.

Amid these challenges, the identification and mitigation of glass damage emerge as pivotal concerns for both mobile carrier customers and vendors. Glass damage not only incurs substantial costs and delays in the supply chain but also aligns with the evolving demands of an environmentally conscious market. Consequently, a critical reassessment of operational practices is imperative to enhance overall efficiency and reduce operational costs.

In response to these challenges, the adoption of automated strategies, particularly leveraging machine learning algorithms, has gained prominence. These automated approaches demonstrate superior cost and time efficiency, offering transformative solutions for the identification of glass damage in warehouse settings. As the demand for such strategies continues to grow, there is a pressing need for advanced systems capable of not only identifying but also classifying specific types of damage. This research addresses this critical gap by proposing the development of a mobile screen damage detection and classification system. The aim is to revolutionize current warehouse procedures, contributing to a more sustainable and efficient supply chain within the mobile carrier industry.

The overarching goal of this research is to develop an advanced mobile screen damage detection and classification system, optimizing reverse logistics processes for warehouses catering to mobile carrier customers/vendors.

2. Related Research

Deep learning methods, a subset of machine learning techniques, are characterized by their ability to learn from large amounts of data and extract complex patterns. These methods employ artificial neural networks, which are inspired by the structure and function of the human brain, to model complex relationships between input data and desired outputs. Deep learning has revolutionized various fields, including computer vision, natural language processing, and speech recognition, and has proven to be particularly effective in tasks involving image and pattern recognition. Deep learning methods have shown substantial progress in diverse areas such as car damage detection (Sruthy et al., 2021; Thomas et al., 2023; Widjojo et al., 2022), road damage identification (Arya, Maeda, Ghosh, et al., 2020; Chen et al., 2023; Pham et al., 2020, 2022), COVID-19 detection from ECG images (Irungu et al., 2023), tank barrier surface damage detection (Dyk & Drahansky, 2023), and structural damage identification (Feng et al., 2019). These methods have also proved to be valuable in the warehouse logistics and e-commerce industry, specifically in addressing the challenge of mobile phone screen surface defect segmentation. Deep learning techniques are emerging as promising solutions to accurately identify and classify these defects, thereby enhancing quality control efficiency in these industries. Deep learning methodologies commonly adopted in these fields include Faster Region-Based Convolutional Neural Networks (Faster R-CNN), You Only Look Once (YOLO) (Pham et al., 2020), Single Shot Detection (SSD), and various pre-trained models with transfer learning such as ResNet, DenseNet, SqueezeNet, MobileNetV2, Inception V3, Xception, VGG16, and VGG19 etc (Selvi et al., 2021; Sruthy et al., 2021; Widjojo et al., 2022). These methodologies highlight the versatility and broad applicability of deep learning across various domains, providing valuable insights for this research.

The study by (Yongfa Lv et al., n.d., 2019) presents a model that leverages Deep Convolutional Generative Adversarial Networks (DCGAN) and Faster R-CNN for detecting defects in mobile phone screen cover glass. Although this model is designed specifically for small sample learning and is effective, it exhibits limitations in speed and scalability when dealing with larger sample sizes.

Li et al., (2021) further contributes to the field by proposing an improved Fuzzy C-Means (FCM) clustering method for extracting defects from mobile phone screen covers. This method, similar to the model presented by (Yongfa Lv et al., n.d., 2019), shows efficacy but is currently limited to small-batch inspections. This highlights a scalability issue that is common in these studies, and the adaptability of this method to larger-scale, production line applications is yet to be confirmed.

Transfer learning has emerged as a powerful technique for addressing complex problems in various domains, including damage detection and classification. Recent studies have demonstrated the effectiveness of transfer learning in classifying and detecting damage in various infrastructure components, such as smartphones, vehicles and hydro-junction infrastructure. (Selvi et al., 2021) proposed a CNN-based system for classifying the level of damage in mobile glass using various pre-trained models such as Densenet, Resnet, Squeezenet, as well as custom models. The system achieves an accuracy of 85%, but the study concludes that the model could be improved in the future by considering the various complex constraints and assumptions of deep learning algorithms.

Feng et al., (2019) applied transfer learning to detect damage in hydro-junction infrastructure using a deep convolutional neural network. Their method achieved a high detection accuracy of 96.8%, surpassing traditional techniques. These studies underscore the flexibility and potential of transfer learning across various domains.

Pham et al., (2020)study focuses on the detection and classification of road damages using Detectron2's implementation of Faster R-CNN. The experiments were conducted using the Global Road Damage Detection Challenge 2020 dataset. The results show the X101-FPN base model for Faster R-CNN was effective and adaptable across different countries. However, despite promising visualizations, the F1 scores were low, indicating room for improvement. Similar to this research on glass defect detection and classification, this study involves the application of machine learning models for damage detection. The use of pre-trained models for transfer learning in this study is analogous to my approach.

Khan et al., (2021) demonstrated the effectiveness of CNNs in classifying various vehicle damage types, with MobileNet and VGG19 achieving accuracies of 70% and 50% respectively. In a subsequent study, (Sruthy et al., 2021) explored the application of CNNs for the detection, analysis, and estimation of various types of car damage. Utilizing transfer learning-based models from the Keras library, such as InceptionV3, Xception, VGG16, VGG19, ResNet50, and MobileNet

, they aimed to predict and classify damage. Their analysis revealed that MobileNet exhibited the highest accuracy, reaching 97.28% in predicting and classifying damage types, and also demonstrated a faster training speed compared to other models. Building on these insights, Widjojo et al., (2022) conducted experiments highlighting the potential of Mask R-CNN in detecting damaged car parts. A simple modification on the CNN classification model resulted in a 9% average increase in the F1 score. Notably, MobileNetV2 emerged as the top-performing classification model, boasting an impressive F1 score of up to 91%.

Together, these studies form a comprehensive basis for understanding the application of transfer learning in damage detection. They provide insightful pathways for future investigations specifically focused on the detection and classification of defects in mobile phone screens using various pre-trained models.

Farhadi et al., (2022)research addresses overfitting in deep learning due to complex layer structures. It explores a combination of L1, L2, Elastic Net-regularization, and Dropout methods, comparing the performance of this combined model with a CNN model without regularization. The study found that a model combining Dropout and Elastic Net regularization outperformed others. In context of my work on mobile phone screen defect detection and classification, this study's approach to mitigate overfitting using combined regularization and Dropout methods offers valuable insights.

Mahyoub et al., (2023) study examines the use of deep learning for automatic car damage detection and addresses the challenge of limited datasets with data augmentation techniques. Specifically, Generative Adversarial Networks (GANs) were used to increase the size and improve the class balance of the dataset, which resulted in better model performance. In relation to my research on mobile phone screen defect detection and classification, this study's application of data augmentation highlights a potential strategy to overcome similar dataset limitations.

The existing literature on mobile screen defect detection and classification has highlighted the potential of deep learning methods. Nonetheless, challenges persist, such as the scalability of models to larger sample sizes and the detection of a diverse array of damage types. Transfer learning emerges as a promising approach to address these challenges. For instance, (Feng et al., 2019) demonstrated that a deep convolutional neural network with transfer learning could accurately detect damage in hydro-junction infrastructure. Data augmentation is another significant technique that enhances the performance of deep learning models. (Mahyoub et al., 2023) used data augmentation and GANs to increase the size and improve the class balance of a dataset for automatic car damage detection, leading to improved model performance. Furthermore, to tackle overfitting due to complex layer structures, a combination of L1, L2, Elastic Net-regularization, and Dropout methods can be employed. (Farhadi et al., 2022) compared the performance of a model using these combined methods with a CNN model without regularization. They found that the model with combined regularization and dropout methods outperformed the others. This approach offers valuable insights for my work on glass damage detection, and I aim to implement these techniques to mitigate overfitting.

Based on these findings, my research will focus on developing a mobile screen damage detection and classification system that is scalable, can detect a diverse range of damage types, and is robust to overfitting. I plan to use transfer learning, data augmentation, and a combination of regularization and dropout methods to achieve these goals.

3. Research Questions

1. How can transfer learning be applied to improve the accuracy of mobile screen defect detection and classification?
2. In what ways can data augmentation enhance the diversity of damage types that the system can detect?
3. How can a combination of regularization and dropout methods be implemented to make the system robust to overfitting?

4. Aim and Objectives

The aim of this research is to develop a scalable, robust, and diversified mobile screen damage detection and classification system, specifically focusing on the detection and classification of oil, scratch, and stain defects.

The research objectives are formulated based on the aim of the study, which are as follows:

- To design and implement a damage detection algorithm capable of identifying and classifying three specific types of mobile screen damages: oil, scratch, and stain.
- To apply transfer learning techniques to improve the accuracy of these specific damage detections.
- To utilize data augmentation methods to increase the diversity of these specific screen damage types the system can identify and classify.
- To incorporate a combination of regularization and dropout methods to enhance the system's robustness against overfitting.
- To evaluate the effectiveness of the proposed system through rigorous testing, particularly focusing on its performance with a high-resolution dataset.
- To analyse and document the findings, with a specific focus on the types of defects present in the dataset.

5. Significance of the Study

The study holds significant importance in the contemporary landscape of reverse logistics for mobile carriers, driven by the surge in e-commerce and heightened environmental regulations. In response to these challenges, businesses face the need to optimize processes related to the management of returned mobile phones. Manual inspection and processing of returned items, a common practice in warehouses, not only occupy substantial space and workforce but also lead to inefficiencies in the supply chain. Addressing these issues, the proposed mobile screen damage detection and classification system seeks to automate and streamline these warehouse procedures. By specifically targeting the identification of glass damage, a prevalent concern for both warehouse customers and vendors, the research aims to introduce a more efficient, cost-effective, and time-saving alternative to traditional manual methods.

The expected outcome of this research is the development of a scalable, robust, and diversified mobile screen damage detection and classification system. The system, designed to identify and categorize specific damage types such as oil, scratch, and stain, is anticipated to surpass the accuracy and efficiency of manual methods. Leveraging transfer learning, data augmentation, and regularization/dropout methods, the system aims to enhance its adaptability to various damage scenarios and exhibit robust performance against overfitting. The successful implementation of this system holds the potential to revolutionize reverse logistics operations for mobile carriers, offering cost reductions, minimized delays, and contributing to a more sustainable and efficient supply chain.

On a broader scale, the implications of this research extend nationally and internationally. At the national level, the study has the potential to significantly influence reverse logistics practices within the mobile industry. The efficient handling of reverse logistics is not only crucial for cost optimization but also aligns with global environmental regulations, emphasizing sustainability. The proposed automated system could serve as a model for other industries facing similar reverse logistics challenges. Internationally, the research contributes to the broader field of artificial intelligence and machine learning applications in supply chain management. The development of advanced detection systems for specific damage types extends beyond the mobile industry, potentially impacting practices in various sectors engaged in reverse logistics. This study aligns with the global push towards technological innovation in supply chain sustainability, making it relevant and impactful on an international scale.

6. Scope of the Study

6.1 In Scope

This study is designed to focus on the development and implementation of a mobile screen damage detection and classification system tailored for reverse logistics operations in the warehouses. The primary components within the scope of this research include the design and implementation of a damage detection algorithm capable of identifying and classifying three specific types of mobile screen damages: oil, scratch, and stain. The study will leverage transfer learning techniques to enhance the accuracy of these specific damage detections. Furthermore, data augmentation methods will be employed to increase the diversity of the identified screen damage types, and a combination of regularization and dropout methods will be incorporated to enhance the system's robustness against overfitting. The evaluation of the proposed system's effectiveness will be conducted through rigorous testing, with a specific focus on its performance with a high-resolution dataset. Findings will be documented and analysed, with particular attention to the types of defects present in the dataset.

6.2 Out of Scope

This study acknowledges certain aspects that fall outside its defined scope. Firstly, it does not encompass a comprehensive analysis of the broader reverse logistics processes beyond the specific focus on mobile screen damage detection. Additionally, the study does not delve into the intricacies of hardware or software components of mobile devices beyond the scope necessary for damage identification. The research does not extend to the development of physical robotics for automated handling of damaged mobile devices in a warehouse setting. Furthermore, while the proposed system aims to detect and classify specific damage types, it does not address the repair or recycling processes associated with the identified damages.

6.3 Reason for Defining the Scope:

Defining the scope is crucial to ensure the research remains focused, feasible, and attains its specific objectives. By concentrating on mobile screen damage detection and classification, the study can provide a deep and meaningful analysis within a manageable framework. This focused approach enables the development of a system that directly addresses the identified challenges in reverse logistics for mobile carriers, providing practical solutions. Defining the scope also ensures that the research remains feasible within the constraints of time, resources, and technical requirements, facilitating a more successful and impactful outcome.

7. Research Methodology

7.1 Workflow

The research will be conducted following a comprehensive workflow that incorporates multiple stages of data handling and modelling. These stages include dataset acquisition, data preprocessing, data transformation/augmentation, modelling, evaluation, and result interpretation. Each phase is intrinsically linked, and findings from each stage will inform adjustments and improvements in the other stages. For instance, the preprocessing might be adjusted based on the model's performance during the evaluation stage, fostering an iterative and dynamic research process.

7.2 Dataset Description

The research will utilize a dataset comprising 1200 high-resolution images (1920×1080) of three types of mobile screen damages: oil, scratch, and stain. Each damage type is represented by 400 images. Additionally, the dataset includes 20 non-defective images. The images have been collected via an industrial camera and are annotated, provided in the PASCAL VOC format. In addition to the existing images, data augmentation techniques will be employed to generate more images. The dataset and associated resources are publicly available on Kaggle at (https://www.kaggle.com/datasets/girish17019/mobile-phone-defect-segmentation-dataset/data / https://github.com/jianzhang96/MSD/)

7.3 Data Preprocessing

Data preprocessing is a crucial phase in the machine learning pipeline, where raw data is transformed into a format that can be readily used by predictive models. This process can include multiple stages. Firstly, images are resized to ensure uniform input to the model. Secondly, pixel values are normalized to a standard scale between 0 and 1, which enhances computational efficiency and promotes stability during model training. To add more variety to the dataset and enhance the model's generalization ability, data augmentation techniques such as rotation, zooming, shifting, and flipping are applied. Also, a part of the dataset (20% in this case) is set aside for validation purposes to measure the model's performance and prevent overfitting. These preprocessing steps are essential to prepare the data for effective training with machine learning models.

7.4 Transformation

Data augmentation is an essential phase in this research, aimed at artificially expanding the size and variability of the dataset. This process is achieved by applying several transformations to the pre-processed images, generating different versions of the same image, each reflecting various scenarios or perspectives of mobile screen damage.

The augmentation techniques applied in this research include image rotation, flipping, zooming, and shifting. These techniques are implemented using TensorFlow's `ImageDataGenerator` function. Through these augmentation techniques, the model is exposed to a wide array of damage types and perspectives, thereby enhancing its ability to generalize and detect a diverse range of damage types accurately. This process increases the robustness of the model, making it less likely to overfit to the training data and more capable of performing well on unseen data.

7.5 Modelling Techniques

This research adopts a modelling phase that leverages the power of transfer learning, using pre-trained models to enhance the accuracy of the damage detection system. Utilizing models that have been pre-trained on large image datasets allows us to benefit from the complex feature extraction capabilities these models have learned. Specifically, I will consider using a variety of pre-trained models such as InceptionV3, Xception, VGG16, VGG19, ResNet50, and MobileNet. These models have demonstrated robust performance in image classification tasks due to their sophisticated architectures. They have been trained on large-scale image databases like ImageNet, learning to identify a multitude of features, which can be beneficial for my task of mobile screen damage detection.

In addition to using transfer learning, the model will incorporate regularization and dropout methods to enhance its robustness and prevent overfitting. Regularization adds a penalty to the loss function, discouraging overly complex models that overfit the training data. Dropout, on the other hand, randomly ignores selected neurons during training, making the model less dependent on any single neuron and thus reducing overfitting.

The combination of transfer learning with models such as InceptionV3, Xception, VGG16, VGG19, ResNet50, and MobileNet, along with regularization and dropout methods, aims to achieve a balance between the ability to learn from the data and the robustness of the model. This balance is crucial in ensuring that model can generalize well to new, unseen data, thereby enhancing its practical applicability.

7.6 Evaluation Metrics

In the evaluation phase, the model's performance is assessed using evaluation metrics such as accuracy, precision, recall, and F1 score. The model's ability to detect each type of defect (oil, scratch, stain) is evaluated separately to ensure effectiveness across all categories. The model's robustness against overfitting is also assessed by comparing its performance on the training and validation sets.

7.7 Result Interpretation

Finally, after the evaluation phase, the results are interpreted. Findings from each phase are thoroughly documented and analysed. This provides insights into the strengths and weaknesses of the system, informs potential improvements, and contributes to the body of knowledge in the field of mobile screen damage detection and classification.

8. Requirements Resources

8.1 Hardware requirements

The research will be primarily conducted on a HP EliteBook 840 G6 laptop. This laptop is equipped with an Intel(R) Core(TM) i7-8665U CPU, which operates at 1.90GHz, has 4 cores, and 8 logical processors, making it capable of handling multiple tasks simultaneously. Furthermore, the laptop has 32.0 GB of installed physical memory (RAM), which allows for efficient handling of large datasets and complex computations.

This hardware configuration provides a strong foundation for the initial stages of the research, such as data preprocessing and augmentation. It also allows for the development and testing of machine learning models locally.

However, the process of training deep learning models, especially with large image datasets and complex architectures, can be computationally demanding. For this reason, the research will also leverage Google Colab for these computationally-intensive tasks. Google Colab provides access to high-performance GPUs, which can significantly accelerate the model training process.

In summary, the combination of the local hardware (HP EliteBook 840 G6) and the cloud-based GPU resources from Google Colab will provide a robust and flexible environment for conducting the research effectively and efficiently.

8.2 Software requirements

Python: Python is the primary programming language for this research due to its readability, ease of use, and extensive support for scientific computing and machine learning libraries.

TensorFlow: TensorFlow is an open-source machine learning library developed by Google. It will be used to build and train the machine learning models, utilizing its capabilities for high-performance numerical computation.

Keras: Keras is a user-friendly neural network library written in Python. Built on top of TensorFlow, it provides a convenient way to define and train almost any kind of deep learning model. Keras will be used for implementing the neural network architecture of model.

Google Colab: Google Colab is a cloud-based platform that provides a coding environment for Python and access to free GPU resources. The research will leverage Google Colab for computationally-intensive tasks such as training the deep learning models.

Other Libraries: Additional Python libraries like NumPy, Pandas, Matplotlib, and Scikit-learn may be used for tasks like data manipulation, analysis, and visualization.

This combination of hardware and software resources will provide the necessary environment to carry out the research effectively and efficiently.

APPENDIX B: CODE REPOSITORY

The code used in this research for the implementation and evaluation of the deep learning models is made publicly available for the purpose of reproducibility and further research exploration. You can find the code repository on GitHub at the following link:

https://github.com/bhogasena/Mobile-Screen-Damage-Classification