CATÓLICA
ESCOLA SUPERIOR DE BIOTECNOLOGIA
PORTO

ANALYSIS OF X-RAY RADIOGRAPHY IMAGES OF PEAR FRUIT USING DEEP LEARNING NETWORKS

by

Beatriz Ferreira

November 2024

CATÓLICA
ESCOLA SUPERIOR DE BIOTECNOLOGIA
PORTO

ANALYSIS OF X-RAY RADIOGRAPHY IMAGES OF PEAR FRUIT USING DEEP LEARNING NETWORKS

Thesis presented to *Escola Superior de Biotecnologia of the Universidade Católica Portuguesa* to fulfill the requirements of Master of Science degree in Food Engineering

by

Beatriz Ferreira

Supervisor (KU Leuven): **Professor Bart Nicolai**
Tutors (Universidade Católica Portuguesa): **Professor Cristina Silva** & **Professor Pedro Rodrigues**

November 2024

Abstract

One of the major challenges facing the international pear production sector is the codling moth (Cydia pomonella), a pest that not only causes significant damage during production and harvest but also enters the international supply chain, leading to economic consequences such as import restrictions, consumer dissatisfaction, and potential health hazards. Current inspection methods are destructive and rely on random sampling, making them inefficient, labor-intensive, and prone to increasing food waste due to decisions being made at the batch level after testing only a limited number of samples.

To address this, the implementation of Vision Transformer (ViT) models was explored in this thesis. Three pre-trained ViT architectures were used: first, fine-tuning a pre-trained model, then, training only the first and last layers while freezing others, and lastly simplifying the model by retaining only initial transformer layers. Additionally, a custom ViT model was optimized using the Optuna framework to fine-tune hyperparameters trained from scratch. These efforts aimed to improve pest detection using X-ray images of pears. Furthermore to assess the impact of patch size on ViT' performance, was compared the performance of ViT models with patch sizes of 16 and 32 across all methods.

It was concluded that the pre-trained ViT-B/16 model with all parameters frozen except for the first layers of the architecture and the last transformer, produced the best results with a balanced accuracy of 72,8%, a training Loss-SENS and LossGRAND-SENS of 0.0033, 0.0030 and a validation Loss-SENS and LossGRAND-SENS of 0.0017 and 0.0013, respectively. Despite the success of ViTs in image classification tasks in other studies, they did not outperform a CNN-based model, EfficientNet6, in this study on the pear dataset. Factors such as differences in augmentation techniques, training splits, and the inherent complexity of ViT architectures likely influenced these results. This reinforces the idea that ViTs typically require larger datasets and more precise tuning to reach optimal performance, highlighting their sensitivity to data quantity and model adjustments.

This research identifies key challenges in pest detection. It addresses these issues by comparing the performance of ViTs and CNNs on small datasets, emphasizing the need for fine-tuning strategies tailored to specialized tasks. The thesis lays the groundwork for future advancements in pest detection, providing solutions to improve model robustness and accuracy in challenging real-world conditions.

Keywords: X-ray CT, Codling Moth, Vision Transformer, Deep Learning, Transfer Learning

Resumo

Um dos principais desafios enfrentados pelo setor de produção internacional de peras é a traça da maçã (Cydia pomonella), uma praga que não apenas causa danos significativos durante a produção e colheita, mas também entra na cadeia de abastecimento internacional, levando a consequências econômicas, como restrições de importação, insatisfação dos consumidores e potenciais riscos à saúde. Os métodos de inspeção atuais são destrutivos e baseiam-se em amostragens aleatórias, tornando-os ineficientes, exigentes em termos de mão de obra e propensos a aumentar o desperdício de alimentos devido a decisões tomadas ao nível do lote após testar apenas um número limitado de amostras.

Para abordar essa questão, nesta tese foi explorada a implementação de modelos Transformador de Visão (ViT). Assim aplicou-se três arquiteturas ViT pré-treinadas: o Fino-Ajuste de um modelo pré-treinado, o treino dos Parâmetros apenas das primeira camada e último transformador enquanto as outras permaneciam congeladas. Por último a simplificação do modelo, mantendo apenas as camadas iniciais do transformador. Além disso, um modelo Personalizado ViT foi otimizado usando a estrutura Optuna para ajustar hiperparâmetros treinados do zero. Esses métodos visaram melhorar a deteção de pragas usando imagens Raios-X de peras. Além disso, para avaliar o impacto do tamanho dos fragmentos no desempenho dos ViTs, foi comparado o desempenho dos modelos ViT com tamanhos de fragmentos de 16 e 32 em todos os métodos.

Concluiu-se que o modelo pré-treinado ViT-B/16, onde todos os parâmetros exceto as primeiras camadas da arquitetura e o último transformador foram descongelados, produziu os melhores resultados. Obteve assim uma balanced accuracy de 72,8%, uma Loss-SENS e LossGRAND-SENS no treino de 0,0033 e 0,0030, e uma Loss-SENS e LossGRAND-SENS na validação de 0,0017 e 0,0013, respetivamente. No entanto, apesar dos modelos ViTs terem mostrado sucesso em tarefas de classificação de imagens noutros estudos, neste caso específico não superaram o modelo CNN, EfficientNet6, para a mesma base de dado de images raio-x das peras. Fatores como diferenças nas técnicas de aumento de dados, divisões de treino e a complexidade das arquiteturas ViT influenciaram esses resultados. Isso reforça a ideia de que os ViTs geralmente requerem bases de dados maiores e ajustes mais precisos para atingir seu desempenho ideal.

Este estudo, identifica os principais desafios na deteção de pragas. Aborda esses problemas comparando o desempenho dos ViTs e CNNs em pequenos conjuntos de dados, enfatizando a necessidade de estratégias de Ajuste-Fino adaptadas a tarefas especializadas. Estabelecendo assim as bases para futuros avanços na deteção de pragas, fornecendo soluções para melhorar a robustez e a precisão dos modelos em condições desafiadoras do mundo real.

Palavras-chave: Aprendizagem por Transferência, Aprendizagem Profunda, Raios-X TC, Traça da maçã, Transformador de Visão

Acknowledgements

This thesis represents a journey filled with numerous challenges and moments of growth, but above all, it has been a rewarding learning experience that has shaped both my academic and professional development, preparing me for the next steps in my professional life.

I would like to extend my deepest gratitude to my daily supervisor, Jiaqi He, without whom this thesis would not have been possible. Her constant availability, quick responses to my emails, and willingness to help me correct and improve my code were invaluable and most importantly, I am profoundly grateful for her patience during my learning process.

Additionally, I would like to thank my two coordinators Prof. Cristina Silva and Prof. Pedro Rodrigues, who each contributed their expertise to the realization of this thesis. A special thanks goes to Prof. Bart Nicolai for allowing me the opportunity to complete my thesis at his research group and to Prof. Pieter Verbovenfor for helping me throughout this journey. The experience of working in this environment has been incredibly rewarding.

I am also deeply thankful to my host university, KU Leuven for providing me with excellent working conditions and opportunities throughout my stay. The support I received during this time was instrumental to the successful completion of my work.

Lastly, I would like to thank my family and friends, particularly my parents, for making it possible for me to pursue my studies and my Erasmus experience, without their unwavering support, none of this would have been possible.

Beatriz Ferreira

"What you get by achieving your goals
is not as important as what you become
by achieving your goals."

Zig Ziglar

Contents

List of Figures

List of Tables

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
Adam	Adaptive Moment Estimation
AdamW	Adaptive Moment Estimation Weight Decay
AI	Artificial Intelligence
CA	California
CASH	Combined Algorithm Selection and Hyperparameter
CCD	Charge-Couple Device
CM	Codling Moth
CNN	Convolutional Neural Network
CT	Computer Tomography
CV	Computer Vision
DL	Deep Learning
EU	European Union
FCL	Fully Connected Layers
FN	False Negatives
FP	False Positives
FPN	Feature Pyramid Network
GB	Gigabytes
IPM	Integrated Pest Management
MA	Massachusetts
ML	Machine Learning
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
MSA	Multi Self-Attention
MT	Million Tons
NLP	Natural Language Processing
P. Communis	Pyrus Communis
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
RGB	Red, Green, Blue
RMSProp	Root Mean Square Propagation
SGD	Stochastic Gradient Descent
SL	Supervised Learning
SSL	Semi-Supervised Learning
TP	True Positives
TN	True Negatives
USA	United States Of America
USL	Unsupervised Learning
ViT	Vision Transformers
ViT-B	Vision Transformer Base
XCT	X-ray Computed Tomography

Chapter 1

Introduction

The spread of invasive species may pose great threats to the economy and safety in the food industry. The codling moth (*Cydia pomonella L.*) is one of the worst invasive species in the world and the most destructive pear pest.

This pest is present in every region where pears are cultivated, especially in warmer, dryer regions. Although the flight capacity of this pest is limited, they can spread over long distances by transporting infested fruit. During its larval stage, the small insect causes irreversible damage to the fruit by feeding on the flesh and creating tunnels that reach the core, resulting in significant internal harm and making the crop unmarketable.

Insect detection is still predominantly through visual inspection or destructive testing of random samples, representative of all batches. These traditional methods are inadequate, time-consuming, labour-intensive and contribute to higher food waste. Additionally, much of the insect damage is not visible in the early stages, increasing the likelihood that undetected fruit reaches the supply chain and/or contaminates healthy pears. Furthermore, climate change directly affects crops through floods and unseasonal climatic conditions, aggravating the situation of pear production losses. The rise in the planet's temperature exacerbates the pest issue, leading to an even greater proliferation of pests (Adedeji et al. (2020), Salama et al. (2021), Mahendiran et al. (2022)). To address this issue, X-ray-based technologies have been developed to visualize the internal structure of food products and, when coupled with deep learning algorithms, can detect pears that are infected. This creates an efficient, automatic and non-destructive method, and reduces the waste of non-infected pears.

This thesis aims to develop a ViT classification model to accurately detect codling moths in 2D X-ray radiographs of infected pears (He et al. (2024)). These 2D radiographs were obtained as simulations of CT scans from an X-ray CT system. To this end, two primary methods were explored: fine-tuning pre-trained ViT models and training a ViT model from scratch with randomly initialized weights. For fine-tuning, the pre-trained ViT-B model, sourced from ImageNet-1K, was used as the base. Two architectures were tested, differing in patch size. Additionally, to simplify the model while leveraging pre-trained weights, two techniques were implemented. First, all layers except the initial two were replaced with identity layers, making only the first layers trainable.

Second, another approach froze all layers except specific ones necessary for the task. This approach aimed to reduce model complexity while still benefiting from the pre-trained weights. For the model trained from scratch, a custom Vision Transformer was optimized using Optuna to find the best architecture, which was then compared to the fine-tuned pre-trained models.

The decision to explore Vision Transformers arises from recent advancements showing that ViTs have surpassed the previously dominant Convolutional Neural Network (CNN) architectures in achieving state-of-the-art performance in image classification tasks (Elyan et al. (2022), Li et al. (2022)). These studies indicate that ViT's ability to learn global image features makes it particularly suited for tasks requiring fine detail and complex spatial relationships.

This thesis consists of five chapters. This first chapter is an introduction to the work done and goals for this research. Chapter 2 includes a literature review of the codling moth, an overview of the current state of the pear fruit market and its challenges, non-destructive inspection methods and reviews of deep learning techniques including ViT and CNN. Chapter 3 details the methods implemented in the model and briefly describes the data acquisition process. Chapter 4 presents the results and chapter 5 the discussion. Finally, chapter 6 offers conclusions, predictions, and suggestions for future work.

Chapter 2

State of the art

This chapter provides an overview of pear fruit, highlighting its importance in the food industry and identifying the primary causes of production losses, namely codling moth and climate change. The life cycle, host preference and management of the codling moth, its global distribution, and its impact on pear production losses are discussed. Furthermore, this chapter reviews the methods for detecting codling moth infestation, comparing destructive and non-destructive techniques, with a particular emphasis on X-ray imaging. The application of deep learning (DL) in computer vision tasks is then introduced, highlighting its relevance. The chapter discusses Convolutional Neural Networks (CNNs), the most commonly used DL architecture for image classification, and Vision Transformers (ViT), which have demonstrated state-of-the-art performance in recent studies in similar tasks.

2.1 Pear Fruit

'Pear' is the common name for at least 20 recognized species of trees with thousands of cultivars and it's also the name of the pomaceous fruit of these trees. The pear tree and shrub are species of the genus Pyrus within the rose family (Rosaceae).

The fruit is globally consumed and recognized as the oldest human-cultivated plant. The likely origin of pears can be traced to 65 to 55 million years ago, specifically in the mountainous areas of southwestern China, but over the years, the pear species spread in both eastward and west directions (Wu et al. (2018)). Now, pears comprise two major types, the 'European' or 'Western' pears, and the 'Asian' pears. The two, display wide morphological and physiological variability, as well as broad adaptations to wide agro-ecological ranges (James-Martin et al. (2015)). Pyrus communis, the primary cultivated variety of European pears, produces fruits characterized by the classic pear shape, featuring soft and smooth flesh, a sparse presence of stone cells, and a robust aroma and flavour profile. On the other hand, the predominant cultivated species in Asia, such as *P. pyrifolia, P. bretschneideri, P. sinkiangensis, and P. ussuriensis*, yield round-shaped fruits with a crisp texture, elevated sugar levels, low acidity, minimal fragrance, and a subtle flavour (Song et al. (2014)).

In 2018, Jun Wu constructed a phylogenetic tree using 420 single-copy conserved genes from nine plant species, showing complex relationships among the large numbers of pear species as shown in Figure 2.1. The four currently recognised cultivated pear species (indicated in blue boxes) have been domesticated from three wild species (Wu et al. (2018)).

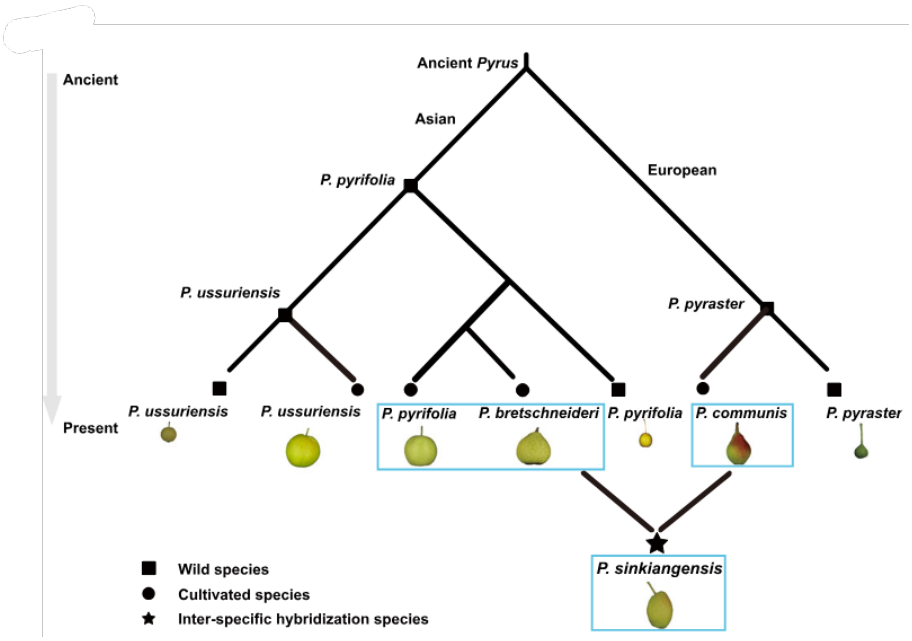

Figure 2.1: Genetic relationships and divergence times of Asian and European pear species, adapted from Wu et al. (2018)

.

Despite these variations, the fundamental structure of the pear remains consistent across all species, as illustrated in Figure 2.2. The outermost layer of the pear, known as the peel or epicarp, serves as a protective barrier against the external environment. Variations in the texture and colour of the epicarp are key characteristics to distinguish different pear species. Some species have thinner peels, while others possess thicker ones. The peel colour can range from green to yellow, red, or brown. Beneath the peel, the fruit is divided into pulp (or mesocarp) and core (or endocarp) which houses the seeds.

Pears surpass all other fruits for their high content of digestive-regulating nutrients including fibre, fructose, and sorbitol (James-Martin et al. (2015)). In general, pears (*P. Communis*) contain 85% water, 14% carbohydrates (comprising sugar and fructose) and 2% fibre, along with vitamins, minerals, and phenolics. Furthermore, pears, particularly the peel, are rich in several phytonutrients, especially phenolic acids which have been associated with multiple health benefits related to diabetes, cardiovascular disease, and obesity (James-Martin et al. (2015)). In conclusion, pears are a nutritionally dense fruit that offers numerous health benefits, contributing to their widespread consumption and significant global production, their versatility in cooking and suitability for various dietary needs, ensure pears remain a popular choice in diets worldwide.

From the pear production perspective, all commercial varieties in Europe belong to the species

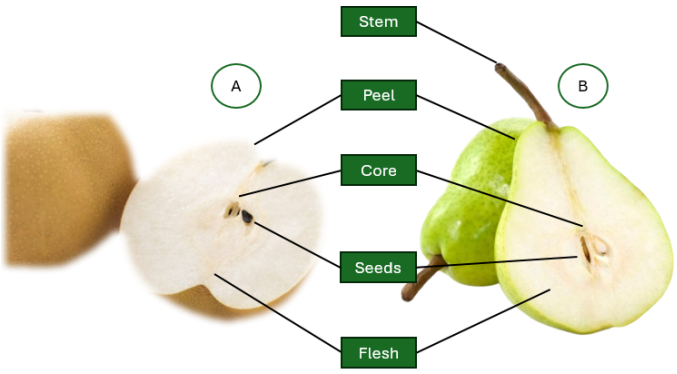

Figure 2.2: Pear's morphology; A- "*Pyrus pyrifolia*"; B- "*Pyrus communis*"

Pyrus communis L., with the Conference variety maintaining its position as the primary variety, representing 53% of total pear production in the EU. The top three world regions are China, Europe and North America. The estimated global pear production for the 2023/24 period anticipates a marginal increase of 300,000 tons, reaching a total of 25.2 million tons (MT) (USDA (2013)). This upturn is primarily attributed to substantial gains in production within China, which effectively counterbalance losses incurred in the European Union (EU) due to weather-related factors. In the EU, the estimated pear production for the 2023/24 season is expected to decline by 249,000 tons, primarily due to severe losses in Italy, the leading producer within the EU. Multiple damaging weather events are forecasted to reduce pear harvesting in Italy by over 60%, significantly contributing to the decrease in total EU production. On a more optimistic side, the forecast of China's pear production in 2023/24 period indicates a substantial rise of 600,000 tons, culminating in a total of 19.6 MT. This positive trajectory marks a recovery from the previous year's frost-damaged crop (USDA Foreign Agricultural Service (2024)). However, Spain and Belgium have expanded their pear cultivation areas in recent years, mainly focusing on the Conference variety. This choice is driven by the variety's adaptability to various climates, high yield and productivity, consumer preference and disease resistance. As a result, Belgium now accounts for nearly 11% of the total pear production in Europe (Organization et al. (2020)) and over 1% globally.

2.2 Damage: main cause of production losses

2.2.1 Climate Change

Climate change is one of the biggest challenges the world faces today, potentially affecting the adaptability of temperate fruit trees, which account for about 48% of global fruit production (Salama et al. (2021)). In 2012, Sibylle Stoeckli (Stoeckli et al. (2012)) reported that in European temperate regions, the anticipated proliferation of agricultural pests and diseases is predicted to extend into areas previously unaffected under climate change scenarios. Rising temperatures, in particular, create more favorable conditions for pests to thrive and spread, exacerbating the problem. Moreover, phenological shifts induced by climatic variability can disrupt plant-pollinator

interactions, potentially resulting in pollination depression or pollinator starvation. This, in turn, poses a threat to the fruit yield in crops that lack self-fertility (De Ollas et al. (2019)).

2.2.2 Insect Pest

Insect pests are recognized as significant contributors to pears waste, causing approximately 10-20% of yield losses in major crops worldwide. This problem is more pronounced in developing countries, where these pests are responsible for around 50% of agricultural and economic losses (Mahendiran et al. (2022)). According to a study (Galinato et al. (2011)), insect damage and the cost of controlling them have risen substantially in recent years. Pear growers spend as much as $14 million annually on insecticides. Increasing costs associated with pest control, including heightened pesticide resistance, fewer pest management options, and rising chemical prices have made pear production more challenging.

The primary insect and mite pests in pear are codling moth and pear psylla (Husain et al. (2018)). When it comes to Pear psylla, this pest diminishes the value of pears as the honeydew secreted by leaf-feeding nymphs drips onto the fruit, causing evident surface damage. Those damages can be identified using an RGB camera, and the affected fruits are often rejected (Westigard et al. (1981)).

2.2.2.1 Codling moth (*Cydia pamonella*)

The Codling Moth (Cydia pomonella) stands as a prominent pest in pome fruit cultivation, predominantly affecting apples and pears and is found in most parts of the world where these fruits are grown, thereby leading to substantial economic losses on a global scale (Maggi & Chreil (2023)). This insect can be a serious pest in pears, especially in the warmer, dryer areas of the India and North Pacific region. Orchards subjected to repeated infestations may face enduring repercussions. The persistent application of pesticides, for example, may lead to heightened costs and pose potential environmental risks. Furthermore, without effective management, codling moth populations can develop resistance to insecticides, thereby adding complexity to control initiatives and aggravating economic losses (Joshi et al. (2020)). A comprehensive grasp of the biology, distribution and management strategies associated with this insect pest is imperative for the formulation of efficacious control measures. Such measures not only serve to minimize environmental impacts but also aim to optimize crop production sustainably.

Host Preference and Distribution The Codling Moth (CM) exhibits a broad host range, affecting diverse apple and pear cultivars, along with wild and ornamental fruit trees. Its distribution is influenced by a combination of climatic conditions, host availability, and trade patterns. This pest is prevalent in temperate regions globally, with noteworthy impacts on fruit production in regions such as North America, Europe, and parts of Asia as shown in Figure 2.3.

In North America, the CM is recognized as a significant threat to apple and pear orchards, with various control strategies employed to mitigate economic losses. In Europe, pests are widespread,

urgent for the need for rigorous pest management practices. Additionally, parts of Asia, where favorable climatic conditions exist, also experience substantial economic impact due to the Codling Moth's presence (Jiang et al. (2018)).

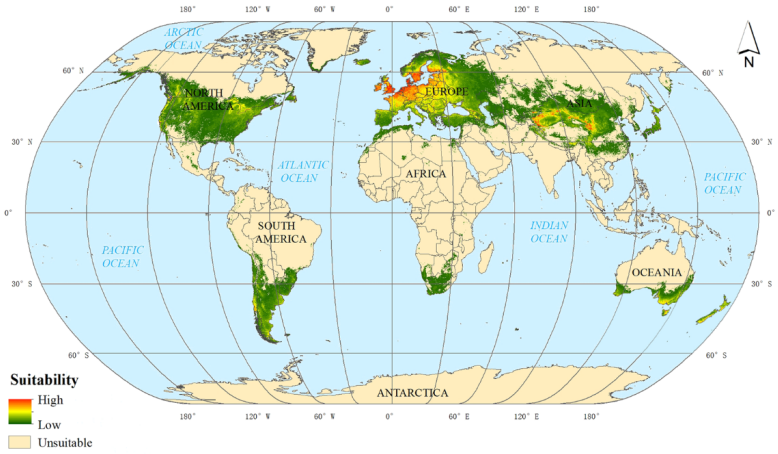

Figure 2.3: Global potential distribution of codling moth using Maxent; adapted from Jiang et al. (2018)

Biology and Life Cycle The codling moth belongs to the family Tortricidae (Lepidoptera). This moth undergoes a comprehensive metamorphic process encompassing four distinct stages: egg, larva, pupa, and adult. As temperatures rise above 10 ℃ (50°F) in the spring, the first generation of moths emerges, mates, and the female moth begins to lay eggs on immature fruits or foliage near the host plant. Each egg is about the size of a pin head and is translucent, gradually darkening as the egg nears hatching. The eggs hatch in six to fourteen weeks, depending on temperature. At that time, the insect has a pink body with a black head and are approximately 1/10 inch in length. Codling moth's have between one and four generations per year, depending on temperature and other climatic factors, and each female can lay 30 to 70 eggs (Husain et al. (2018)). After the eggs hatch, the young larvae seek out and bore into the fruit and begin to feed on the flesh, causing internal damage. Larval development occurs within the fruit, they also may tunnel into the core of the fruit. When fully grown, they exit the fruit to find suitable sites for pupation. In the form of a cocoon, they continue the life cycle in the soil or on debris under the tree. The larvae pupate inside their cocoons in early spring and emerge as adult moths from mid-March to early April (Kramer (2012)). Upon emerging from the pupal stage, adult moths engage in mating and instigate new generations, giving rise to multiple cycles within a single growing season. Adults are about 1/2 to 3/4 inch long and their forewings are dark greyish with waxy lines with a copper-coloured eye-like circle toward the margin. The codling moth's appearance blends well with most tree bark, making it difficult to detect (Maggi & Chreil (2023)). Understanding the complexities of codling moth reproductive behavior is crucial for developing effective strategies to manage and control their populations. It is important to recognize that the absence of visible signs does not necessarily indicate the pest's absence. Therefore, employing methods capable of penetrating the fruit is essential for accurate assessment and detection.

Management Codling Moth larvae induce damage by tunnelling through the fruit, rendering it unmarketable and prone to secondary infections. Infested fruits display characteristic signs, including entry holes, internal feeding tunnels, the presence of frass (insect excrement), and premature dropping (Maggi & Chreil (2023)). Additionally, during the initial phases of infestations, almost no external signs are visible meaning that the possibility of infiltration into the supply chain is high. The damage inflicted by CM is substantial and can result in noteworthy economic losses, that arise from direct yield reduction, increased management costs, and post-harvest quality issues. Additionally, Codling Moth infestations may result in trade barriers due to quarantine restrictions imposed by importing countries to prevent the pest from reaching consumers (Fan et al. (2022)).

The pear moth (*Cydia pomonella*) is a serious pest of pears. Most of the lesions on pears at the beginning of the season occur at the end of the fruit's calyx. As the season progresses, all pears become increasingly vulnerable to the pear moth (Kadoić Balaško et al. (2020)). In terms of chemical control, insecticide resistance represents a significant challenge. Growers and consultants usually avoid treating consecutive generations of pear moths with the same materials or modes of action since this pest adapts well to insecticides from generation to generation (Westigard et al. (1981)). In biological control methods involve the use of general predators such as bats and spiders, which feed on adult moths. Additionally, arthropod predators like spiders and ground beetles, as well as vertebrate predators such as birds and mice, feed on mature larvae seeking overwintering sites (Kadoić Balaško et al. (2020)).

Cultural control practices include removing brush, debris and infested fruit from the orchard. Employing best practices for harvest bin placement and treatment, such as hot water treatment, tarping, and fumigation, in the orchard helps minimize codling moth infestation (Kadoić Balaško et al. (2020)). Additionally, due to the recent global need to reduce chemical residues in agricultural commodities, integrated pest management (IPM) has emerged as a method to combine several pest management techniques (Sumedrea et al. (2015)). Furthermore, other techniques have been recently studied to suppress or eliminate CM activity but still do not guarantee the total elimination of CM-infested fruits in addition to the possible fruit deterioration in accompanying heat treatments (Wang et al. (2001)).

2.3 Detection Methods- Codling Moth

Despite chemical and biological control, codling moth's infection still occurs. The presence of the codling moth significantly challenges the production and harvesting stages leading to considerable damage. Moreover, these insects can inadvertently infiltrate the international supply chain, causing potential health risks. Given the global food supply chain, many nations maintain a zero-tolerance policy towards invasive species, exemplified by the rejection of entire batches upon the detection of even a single specimen (Khaled et al. (2022)), precise detection is crucial to avoid exacerbating food waste issues, economic losses, and environmental concerns (Rady et al. (2017)).

This section provides an overview of the current most used strategies for detecting pests in fruits, covering both traditional and non-destructive methods and discusses the advantages and

disadvantages of various techniques.

2.3.1 Traditional Manual Methods

Manual detection techniques, known for their simplicity and lack of requirement for specialized equipment, continue to be widely used. These methods allow for a detailed examination of each sample, facilitating accurate pest identification. However, manual detection relies on random sampling and physical examination, where inspectors search the fruit surface for damage, holes, excrement, or other indications of pest presence. This approach is highly subjective, relying heavily on the inspector's visual assessment, making it time-consuming and labour-intensive. As a result, it is less practical for large volumes of samples and can be costly, inaccurate, and inefficient. Furthermore, the effectiveness of this method depends on the skill and experience of the inspector, leading to potential variability in pest detection outcomes (Liang et al. (2021), Adedeji et al. (2020)).

2.3.2 Non-destructive Methods

The non-destructive approach allows the identification and assessment of pest presence or damage without causing harm to the fruit itself (Adedeji et al. (2020)). These methods will help detect the presence of infested fruits before packing and transporting them to local or international markets. Additionally, such methods, due to their reliance on physical properties that closely correlate with specific quality factors of fruits and vegetables have the potential to provide accurate and effective judgment about the stage of the codling moth insect and the degree of infestation (Khaled et al. (2022)).

Non-destructive methods prove to be more effective than traditional conventional methods. Besides, non-destructive methods preserve the integrity of the fruit tissue while enabling the assessment of internal structures and quality, reducing food waste. The table below 2.1, offers a comprehensive overview of the main non-destructive methods currently applied in the food industry.

2.4 X-ray

2.4.1 X-ray generation

The discovery of X-ray technology by Wilhelm Conrad Roentgen in 1895 marked a revolutionary advancement in medicine, physics, and materials science. Since then, the application of X-rays has been especially notable in the clinical sector (Do Huh & Kim (2020)). The X-ray radiation falls within the electromagnetic spectrum, positioned between gamma rays and ultraviolet radiation. It emits energies ranging from 200 electron volts (eV) to 1 million electron volts (MeV) and is distinguished by its relatively short wavelengths, ranging from 0.01 to 10 nanometers (nm) (Van De Looverbosch et al. (2020)).

The generation of this high-energy electromagnetic radiation occurs when fast moving electrons are stopped suddenly by impact on a metal target converting the electrons energy into X-ray

Table 2.1: Overview of non-destructive methods to detect insects in fruits

Method	Principle	Advantages	Disadvantages
Visible light sensing	Detects external or surface defects using electromagnetic waves in the visible spectrum (380–750 nm) Adedeji et al. (2020), Liu et al. (2017)	Simple, cost-effective, and accurate (93.4–100% classification) Blasco et al. (2007)	Cannot detect internal defects; Surface color can interfere with detection Cubero et al. (2011)
Nuclear Magnetic Resonance	High-frequency NIR can penetrate the whole fruit and detect chemical changes Adedeji et al. (2020), Moscetti et al. (2014)	Measures multiple quality attributes without sample prep Rady et al. (2017)	High moisture content in fruits can interfere with detection Saranwong et al. (2011)
Hyperspectral Imaging Systems (HRI)	Captures detailed spectral data for each pixel, often using machine learning for classification Lu et al. (2017), El-Mesery et al. (2019)	Detects both internal defects and surface features El-Mesery et al. (2019)	Data requires extensive analysis; Expensive equipment El-Mesery et al. (2019)
Acoustic	Detects larvae movement and feeding sounds in post-harvest fruits Mankin et al. (2002)	Inexpensive, sensitive, useful for early detection Sutin et al. (2019)	Distinguishing insect sounds from noise can be difficult Pinhas et al. (2008)
Thermal Imaging	Detects surface temperature variations due to pest activities Adedeji et al. (2020)	Easy to handle, portable, detects pest-induced temperature changes Pathmanaban et al. (2019)	High-resolution imaging can be costly, dependent on environment Adedeji et al. (2020)

(1%) and heat (99%).The process is typically facilitated within an X-ray tube, that consists of two electrodes sealed in a vacuum (Figure 2.4); the negative electrode (a tungsten filament); and a positive electrode (a flat metal target). The filament at the cathode is heated, thereby transferring thermal energy to the electrons. When the thermal energy exceeds the binding energy of an electron to the cathode, the electron is liberated from the cathode. After, the free electrons are accelerated towards the anode, by applying an acceleration voltage. Upon collision, the electrons change energy and emit X-ray (Omar et al. (2020), Wilkinson (2012)). Furthermore, when the anode is tilted, the X-rays are generated in the desired direction

In the agriculture sector, the incorporation of X-ray technology has evolved gradually over the decades, fostering a transition in agricultural management from labour-intensive practices to more efficient X-ray-based methods (Mathanker et al. (2013)). Moreover, a variant of X-rays known as soft X-ray, characterized by wavelengths ranging from 0.1 to 10 nm, possesses diminished penetration power while retaining the ability to unveil internal density changes. According to a study (Neethirajan et al. (2007)), soft X-rays are particularly suitable for agricultural products, with the advantage of rapid image acquisition, typically requiring only a few seconds (3–5 s). Another study (Qian et al. (2021)) develop a soft X-ray imaging technique to effectively and non-destructive evaluate the internal quality of citrus, addressing the limitations of traditional methods. This approach yielded remarkably high accuracy, achieving 96.2% detection success. The soft X-ray, demonstrated expertise in discerning internal density variations, making it suitable for analysis of agricultural products.

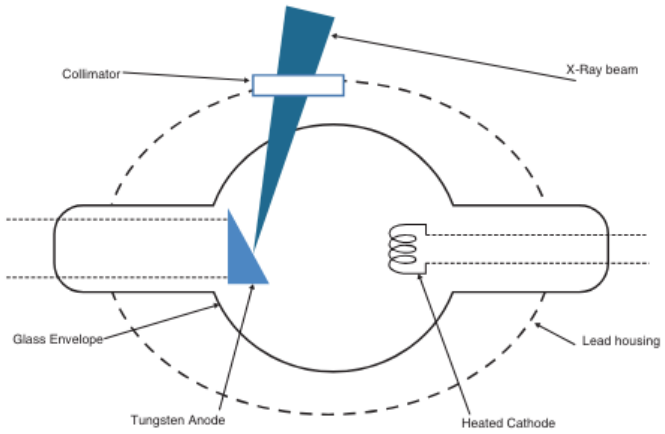

Figure 2.4: X-ray generation and production of a beam for use in imaging; adapted from Taylor (2023)

2.4.2 X-ray Imaging

X-ray imaging is an imaging technology that uses X-rays to create representations of an object. The major techniques are X-ray radiography and X-ray computer tomography, resumed in Table 2.2. When the X-ray beam interacts with an object the energy can be transmitted, scattered (Compton scattering) or absorbed (photoelectric collision) (Kotwaliwale et al. (2014)). This reduction of bean intensity, called attenuation, occurs when photons interact with the atoms in the sample (Ekramirad et al. (2016)). The attention of X-ray radiation describes how easily a material can be penetrated by a beam of light and depending on the composition of the sample, will result in different levels of attenuation. In medical radiography, materials with higher atomic numbers and densities, such as bones and metals, attenuate X-rays more strongly than materials with lower atomic numbers and densities, such as soft tissues (Curry et al. (1990), Kotwaliwale et al. (2014)).

Creating an X-ray image involves capturing the attenuated X-ray radiation with a detector, such as a film or digital sensor. The detector records the pattern of transmitted X-rays, which are then converted into a digital image. Areas of the object that attenuate the X-rays more strongly appear darker, while areas that attenuate the X-rays less strongly appear brighter on a grayscale image (Kotwaliwale et al. (2014), Ekramirad et al. (2016)). The intensity reduces exponentially as it moves away from the X-ray generator and it's given by the Lambert-Beer Law:

$$I = I_0 e^{-\mu_m \rho z} \tag{2.1}$$

I is the intensity of photons exiting through a body; I_0 is Initial intensity (before entering the material); μ_m is mass attenuation coefficient in mm^2/g; ρ is material density in g/mm^3(M L^{-3}) and z is the thickness mm (L) through which the X-rays pass

2.4.3 X-ray Radiography

In radiography, unabsorbed X-rays passing through an object are captured by a detector and converted into an electronic signal, producing a two-dimensional representation of the object. Radiographs display the cumulative linear attenuation coefficients along the X-ray path from source to detector as described by the Lambert-Beer law (2.1).

X-ray radiography incorporates two main approaches: line sensor radiography and area sensor radiography (scanner), as demonstrated in Figure 2.5. Area sensor radiography utilizes two-dimensional detector arrays to produce cross-sectional images of the object. In contrast, line sensor radiography employs a linear array of detectors that capture X-ray attenuation data along a single line as the object moves through the X-ray beam. This method enables rapid image acquisition, making it ideal for real-time industrial inspections (Kim et al. (2014), Van Dael (2017)).

The detectors in both approaches convert the X-rays into visible light or electronic signals, which are then captured by a Charge-Coupled Device (CCD) camera. The CCD camera serves as the imaging sensor in the radiography system, allowing the X-ray images to be viewed on a computer monitor or printed for diagnostic purposes (Kotwaliwale et al. (2014)). These images are compiled through controlled scanning and algorithmic reconstruction, providing detailed spatial information crucial for medical diagnostics and the analysis of complex components.

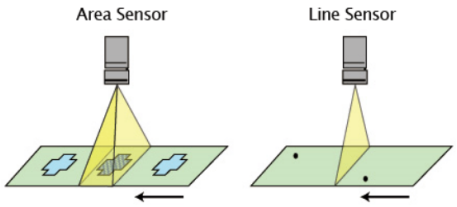

Figure 2.5: Two types of X-ray radiography systems; X-ray with line sensor (left) and area sensor scanner (right), adapted from Van Dael (2017)

The application of this method in the food industry faces several challenges. For high-water-content materials, the methods shows limited penetration. Furthermore, the radiography technique can only acquire a two-dimensional plane of a three-dimensional body, resulting in the superposed of object details. This causes structures resembling defects in the X-ray images, known artifacts, thereby complicating the image analysis process (Yulianti et al. (2018)). Additionally, the implementation of X-ray imaging systems can be costly, potentially limiting their widespread adoption in the fruit industry. Nevertheless, the utilization of X-ray imaging in fruit inspection represents a promising approach for non-destructive quality assessment (Adedeji et al. (2020)).

2.4.4 X-ray Computed Tomography

In X-ray radiography, a significant limitation arises due to the loss of depth information in a 2D projection. To address this challenge, the technique of Computed Tomography (CT) was introduced in the 1970s (Hounsfield (1973)).

In comparison to radiography systems, XCT systems uses X-ray rotating round a sample to acquire images "projections", as shown Figure 2.6. The collected 2D projections are used to calculate the X-ray intensity projections and are aligned and processed. In the next step, using specialized algorithms and computational techniques, the X-ray intensity projections are reconstructed into a stack of virtual cross-sectional images. Each cross-sectional image represents a thin slice of the sample, capturing its internal features in detail. This intermediate step involves aligning and processing the 2D projections and the slice is subdivided into a matrix of volume elements (voxels).

The image is reconstructed by a computer as a corresponding matrix of picture elements (pixels). Each pixel represents the average linear attenuation coefficient of a voxel, with its gray scale value reflecting the proportion of X-rays absorbed in that specific area. CT numbers are assigned to each pixel in the image by a computer algorithm that uses the measurements of the transmitted X-rays as data (Wilkinson (2012)). The increasing availability, speed and reliability of computed

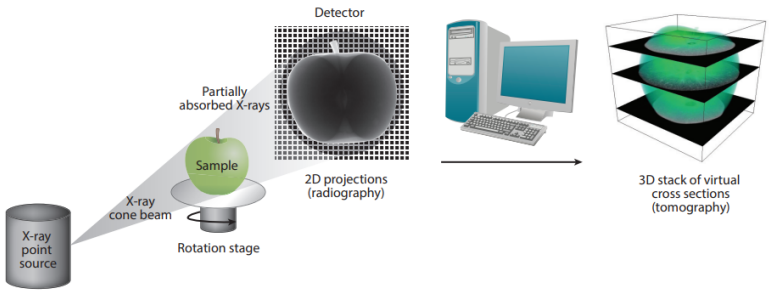

Figure 2.6: Schematics illustrating the working principle of CT scanning and imaging, adapted from Wang et al. (2018)

tomography has resulted in its widespread use in urological imaging (Wilkinson (2012)). Consequently, many methods have been developed to maximize performance.

Van De Looverbosch (Van De Looverbosch et al. (2020)) conducted a study using X-ray CT scans on *Pyrus communis L. cv. 'Cepuna'* and *Pyrus communis L. cv. 'Conference'* pear varieties. The X-ray allowed for the visual differentiation of various parts of the fruit. These differences are possible because different parts of the pear have different densities. The lower-density regions of the pear (e.g., cavities, holes) appear as black areas (Hanke et al. (2008)).

Nevertheless, future technology needs to address some XCT disadvantages for broader application in the food industry. Duo to the complex nature of X-ray CT data analysis, specialized training is required, and the processing times are longer compared to other techniques. These factors can lead to delays in fruit inspection and distribution processes if applied in the fruit industry (Ketcham & Carlson (2001)). Furthermore, the X-ray scanning equipment is expensive to acquire and maintain, making it economically unfeasible for small-scale fruit producers or farmers (Adedeji et al. (2020), Olakanmi et al. (2023)).

Table 2.2: Overview and differences between X-ray radiography and X-ray CT

Type	Principle	Process	Use
X-ray radiography	X-ray radiography captures a single 2D projection of the internal structures of the body onto a film or digital detector. It relies on the differential absorption of X-rays by different tissues in the body	A focused beam of X-rays is passed through the object body being examined. The X-rays are absorbed or scattered by the tissues they encounter. With denser tissues (e.g., bone) absorbing more X-rays and appearing whiter on the image	In medicine: physically describing the skeleton, including fractures, luxation, bone disease, and the location of foreign matters, is useful for guiding the surgery Jiang et al. (2018)
X-ray Computed Tomography (CT)	X-ray computed tomography (CT) generates detailed cross-sectional images of the body by rotating an X-ray source and detector around the patient. It utilizes computer processing to reconstruct these cross-sectional images (tomographic slices) from multiple X-ray projections taken from different angles	As the X-ray source rotates around the patient, it emits X-ray beams from various angles. The detector measures the intensity of the X-rays that pass through the body at each angle. A computer then processes this information to create detailed cross-sectional images of the internal structures	Food industry: accurately characterise food products to establish an intrinsic relationship between micro-structure and food quality Lim & Barigou (2004)

2.5 Deep Learning

2.5.1 Introduction to Machine Learning

Intelligence is the ability to process information such that we can use it to inform future decisions or actions. In artificial intelligence (AI), AI is the ability to build artificial algorithms that can process information to advise future decisions.

Machine Learning (ML), a prevailing branch of artificial intelligence, focuses on developing algorithms and statistical models that enable computers to learn from and make predictions, or decisions based on given data. In this way, the machine can learn automatically from a given dataset and be able to make a proper decision (test) by itself (Chollet (2017)). Machine learning (ML) encompasses three primary learning paradigms: (i) supervised learning (SL), (ii) unsupervised learning (USL), and (iii) semi-supervised learning (SSL). In supervised learning, models are trained using labelled information to make predictions. Conversely, unsupervised learning utilises unlabelled data to extract features from input data, and lastly, semi-supervised learning combines elements of both supervised and unsupervised learning. This approach utilizing a limited amount of labelled data alongside a larger pool of unlabelled data, with application in image processing, and bio-informatics (Camacho et al. (2018)).

Machine Learning is present in two discernible conceptual sub-fields, Classic Machine Learning and Deep Learning. The Shallow Machine Learning (Classic Machine Learning) predominantly relies on the extraction and manipulation of handcrafted features from input data, facilitating the training of models to perform specific tasks without the need for extensive computational resources (Goodfellow et al. (2016), Janiesch et al. (2021)). In medical imaging, traditional machine learning models often struggle with differences in image quality caused by varying acquisition parameters like spatial resolution. To address this, extensive pre-processing of data is typically required before extracting features. Furthermore, this can reduce accuracy and performance when the model is applied to real-world scenarios where conditions may differ from those encountered during the training process (Castiglioni et al. (2021)).

Perceptron is fundamentally rooted in the architecture and function of neural networks. The perceptron simulates the way biological neural systems process information. It consists of input values (x), weights(w), bias (b), and an activation function (f). It processes inputs by applying the weights and bias, then passes the result through an activation function to produce an output.

As shown in Figure 2.7, a neural network consists of interconnected neural layers, that transmit and transform data through weighted connections. The process begins with the input layer, where the raw data is received, and finishes with the output layer, which generates the final prediction (y). Hidden layers, located between the input and output layers, perform computations and pass the information to the next layer. Activation functions (f) like ReLU, Sigmoid, or tanh are applied within these layers to introduce non-linearity. This allows the network to capture complex patterns and relationships in the data (Ünal & Başçiftçi (2022), Janiesch et al. (2021)).

Furthermore, it's imperative to recognize that these networks lack prior knowledge of their environment or context. This implies that neural networks are randomly initialized and necessitate

a deliberate training process (Yong et al. (2020)).

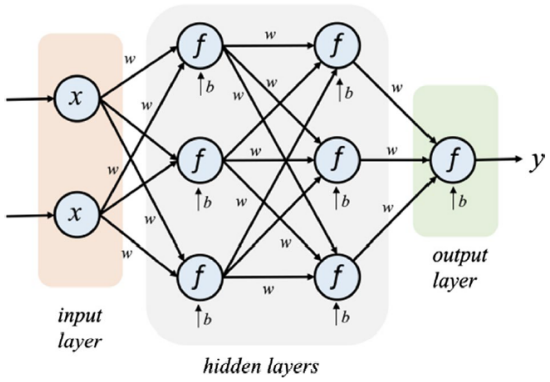

Figure 2.7: A typical Artificial Neural Network with two hidden layers and a single output, adapted from Ünal & Başçiftçi (2022)

The second sub-field, Deep neural networks typically consists of more than one hidden layer, organized in deeply nested network architectures. They may use advanced operations (e.g., convolutions) or multiple activation functions, rather than relying on a single, simple activation function. These characteristics allow deep neural networks to be fed with raw input data and automatically discover a representation that is needed for the corresponding task. Accordingly, DL is particularly useful in domains with large and high-dimensional data (e.g., text, image, video), which is why deep neural networks outperform shallow ML algorithms (Ünal & Başçiftçi (2022)).

In the context of SL, training a neural network involves minimising a loss function. This involves an optimization process with two main steps: calculate gradients through backpropagation and update new weights using optimization algorithms Figure 2.8.

During each training iteration (or batch), the input data is forwarded through the network to compute the predicted output. This output is then compared to the actual target values using a loss function, which quantifies the discrepancy between the real value and the model's prediction. Following, the Backpropagation calculates the gradient of the loss function concerning each weight, allowing for weight updates. These updates are performed using optimization algorithms like Adam (Adaptive Moment Estimation) or SGD (Stochastic Gradient Descent) which use the gradient to adjust the weights. The magnitude of this adjustment is determined by the learning rate, a predetermined hyperparameter. Optimization algorithms like Adam and RMSProp dynamically adjust the learning rate and incorporate momentum, which accelerates convergence towards the minimum loss function value and improves training efficiency (Yuan et al. (2021), Shamsudin (2013)).

However, to ensure optimal performance and computational efficiency, it is crucial to determine the appropriate point to stop the training process of a neural network. This is typically achieved through early stopping, a technique that monitors the model's performance on a validation dataset. Training is halted to prevent overfitting when the validation loss stops improving for a specified number of consecutive epochs (the patience parameter). Validation involves using

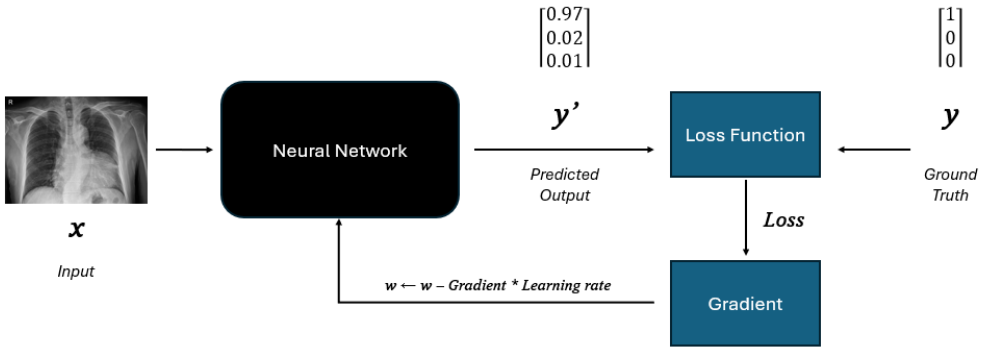

Figure 2.8: The representation of a neural network training process; adapted from Shamsudin (2013)

a separate dataset to fine-tune hyperparameters and confirm the model's ability to generalize to unseen data. Following this, the model undergoes a testing phase. That phase evaluates the final model's performance on a completely independent dataset, providing an unbiased assessment of its real-world applicability and accuracy (Bentoumi et al. (2022), Castiglioni et al. (2021)). As illustrated (Figure 2.9), the data is divided into the training phase, and the rest is unseen data, which evaluates the model's performance. Implementing those steps ensures the model robustly on new, unseen data.

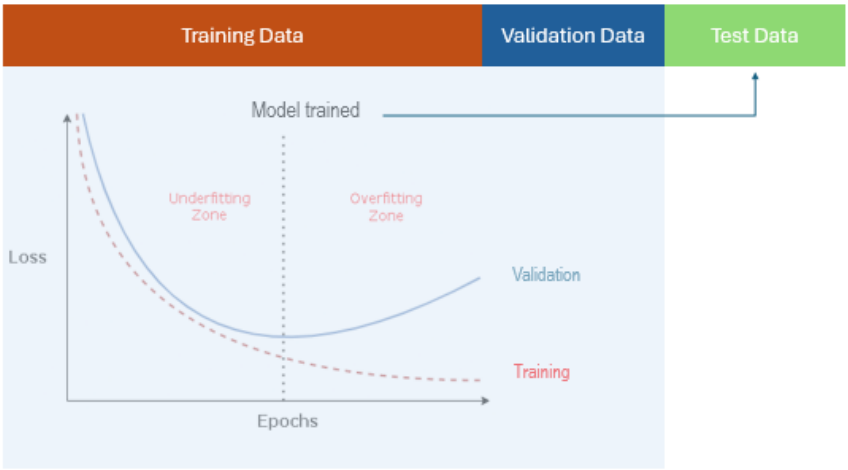

Figure 2.9: Representation of models phases and when to stop training phase; adapted from Janiesch et al. (2021)

2.5.2 Deep Learning in Computer Vision

Computer Vision (CV) is concerned with giving the computer the ability to process and analyse visual content such as 2D, videos, and 3D images. This field can be delineated into three primary task categories: object recognition/detection, image segmentation, and image classification.

To tackle the drawback of ineffective feature extraction in traditional machine learning, DL has been developed. DL methods can learn features (underlying representations) of the input images without the need for manual feature extraction or engineering. Because of that, DL models are considered the most common and top-performance algorithms in handling many CV tasks (Chai & Li (2019)). This introduction has notably strengthened various applications of CV, especially within the biomedical field, where significant advancements in medical applications have been observed (Schwab et al. (2020)).

In computer vision's primary tasks, image recognition detection models aim to accurately identify and locate objects within an image. This is done by defining a bounding box or mask to indicate which pixels in the image represent the object of interest (Elyan et al. (2022)). A study (Pang et al. (2019)) use a DL detection algorithm (You Only Look Once (YOLO)) to identify and classify cholelithiasis (gallstones)and obtain an accuracy of 92%.

Image segmentation, on the other hand, refers to a pixel-wise classification task that segments an image into areas with the same attributes (e.g., anatomical parts in the image). While object detection methods often produce a bounding box defining the region of interest, segmentation methods will produce a pixel mask for that region (Elyan et al. (2022)). Ranjbarzadeh (Ranjbarzadeh et al. (2021)) combined preprocessing techniques focused on small image regions with a Cascade Convolutional Neural Network (C-ConvNet/C-CNN) to robustly segment brain tumor in multimodal MRI images. The study obtains Dice scores over 0.90, demonstrating the efficacy of DL models in segmentation.

Image Classification tasks are considered the most common problems in CV. The classification task aims to find a function $h(x)$ that maps an input image X to Y (Sarvamangala & Kulkarni (2022)). Traditionally, it was used to classify an image, a computer-aided diagnosis (CAD) system for medical image analysis (Doi (2007)). However, since the implementation of CNN-based methods, the field of image classification has significantly advanced. Besides, convolutional neural network techniques can efficiently process/train large sizes of datasets and then use them in sensitive or unrepeatable application scenarios.

In the agriculture sector, 2D X-ray projections were analyzed using a Convolutional Neural Network (a DL model) for image classification tasks to differentiate between bigerm and single seeds (Van De Looverbosch et al. (2020)). An excellent average scores higher then 0.90 was achieved. This development can be largely attributed to the improvement that took place at the algorithm levels (especially CNN methods), the progress in computing power, and the availability of large volumes of medical images and related data in the public domain (Elyan et al. (2022)).

2.6 Image Classification - Methods

2.6.1 Convolutional Neural Network (CNN)

Within DL methods, the Convolutional Neural Network (CNN) is the most used for image-processing tasks (Castiglioni et al. (2021)). One of the distinguishing features of CNN architecture compared

to Fully Connected Layers is the concept of local connectivity. Local Connectivity means that each neuron in the hidden layer only sees a certain of inputs in the previous layer (Sarvamangala & Kulkarni (2022), Castiglioni et al. (2021)).

The convolutional layers are considered the main building blocks in CNNs. These layers are connected to a small local region of the input data. Convolutional filters, also known as kernels, are then applied to the input data by sliding them across the input feature maps. Each adaptive kernel performs a dot product computation between the filter weights and the input data at each position (Figure 2.10). Depending on the input data, one, two or three-dimensional kernels can be employed, and the weights are updated during the training phase. That way, this method can extract imaging features tailored to the investigated task (Sarvamangala & Kulkarni (2022)). Furthermore, this process generates a new set of feature maps, which represent spatially localized patterns or features present in the input data. The convolutional layers are then followed by activation functions (e.g. RELU) that introduce non-linearity to the data. Furthermore, to reduce feature map resolution, overfitting and training time, pooling layers are usually inserted between successive convolutional layers in CNN architecture. The pooling layers operate independently on every feature channel, aggregating data of a local region (e.g., rectangle) and transforming them into one single value. In the max pooling operation, an example present in Figure 2.10, is selected the highest value in a particular feature map region (Chai et al. (2021), Guo et al. (2016)). The last layer in CNN architectures is a typically fully connected Layer. In this layer, each unit is densely connected to all the units of the previous layer, and its purpose is to use these features for classifying the input image into various predefined classes Figure 2.11.

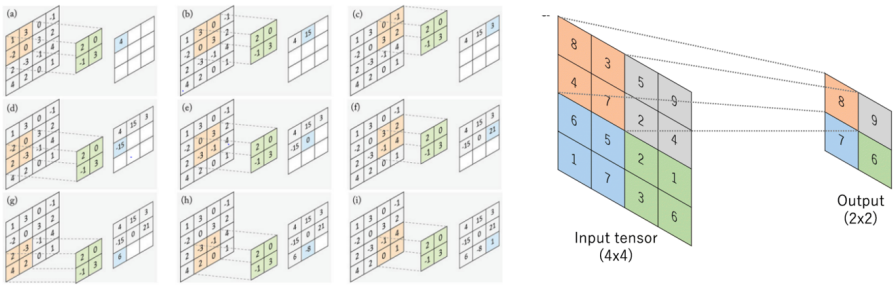

Figure 2.10: Convolutional Layer Architectures: Operation between a filter 2x2 aligns within a 4x4 input feature map with filter step of 1 (left); Max polling process (right); adapted from Khan et al. (2018)

As the CNN progresses through multiple layers of convolution and pooling, each subsequent layer of kernels operates on a larger receptive field. This is because the pooling operation effectively aggregates information from larger regions of the input, combining information from multiple adjacent pixels into a single value. Consequently contrary to other methods, CNN can extract features from increasingly larger regions of the input image, enabling the capture of more global spatial relationships and semantic information. This hierarchical feature extraction is crucial for tasks like image classifications (Chai et al. (2021), Guo et al. (2016)).

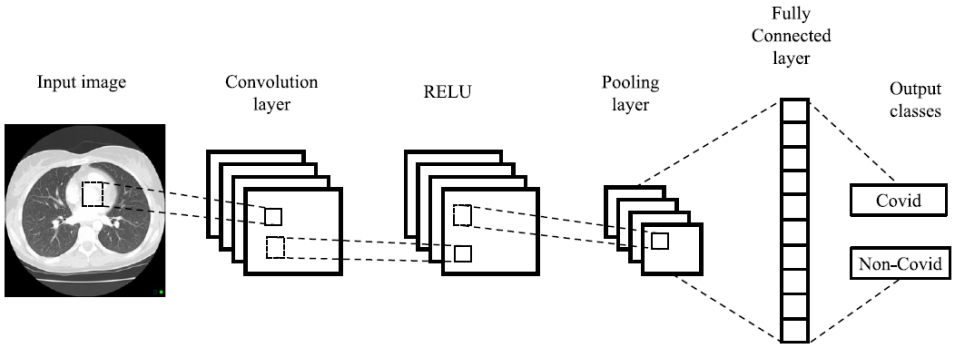

Figure 2.11: Convolutional Layer Architectures: Operation between a filter 2x2 aligns within a 4x4 input feature map with filter step of 1 (left); Max polling process (right); adapted from Kugunavar & Prabhakar (2021)

The CNN models are more focused on detecting local details rather than long-range semantic relationships within and across images, as convolution operates on a fixed-size window. Additionally, max-pooling can lead to a loss of important information (Nijhawan et al. (2022)).

Despite these limitations, CNN models have proven highly valuable in the medical domain. Numerous CNN-based network structures have been developed and applied to various image classification tasks. Some of these are summarized in Table 2.3.

Table 2.3: CNN classification architectures summarized, adapted from Chai et al. (2021)

	Year	Architecture	Model Size	Limitation
AlexNet Chai et al. (2021), Krizhevsky et al. (2012)	2012	5 convolutional layers some followed by max-pooling layers 3 fully-connected layers with 1000-way softwax	Total parameters 62.3 million. Total size 249.2 million bytes (MB)	Space for accuracy improvement
VGGNet Chai et al. (2021), Simonyan & Zisserman (2014)	2014	13/19 convolutional layers with small convolution filters (3x3) in all layers followed by a max pooling layer, 3 fully connected layers	Total parameters VGG16 138 million or VGG19 144 million. Total size 552 million bytes (MB), 574.8 million bytes (MB)	Evaluating the network requires a lot of computation
GoogleNet & Inception Chai et al. (2021), Szegedy et al. (2015)	2015	9 "Inception Modules" - consisting of 1x1, 3x3 and 5x5 convolutions, as well as max-pooling operations, performed in parallel. Applying smaller kernels promotes a reduction in the input features, 21 convolutional layers and 1 fully connected	Total parameters 6,8 million. Total size 27,2 million bytes (MB)	Inception's complexity makes it hard to adapt efficiently to new use cases
ResNet Chai et al. (2021), Kastner & Heinzl (2015)	2016	ResNet allows the network to bypass (avoid) one or more layers by having a Residual Block and Shortcut connection. 152 convolutional layers with max-pooling layers	Total parameters 21,8 million. Total size 87,2 million bytes (MB)	Suffer from overfitting
DenseNet Chai et al. (2021), Huang et al. (2017)	2017	3 dense blocks - all layers are connected directly with each other has L(L+1)/2 connections (L numbers of layers), 2 Transition layer	Total parameters 7,98 million. Total size 31,92 million bytes (MB)	
EfficientNet Chai et al. (2021), Tan & Le (2019)	2018	EfficientNet employs a compound scaling method, uniformly adjusting depth, width, and resolution, resulting in improved performance across various tasks while maintaining efficiency, 8 kinds of network	Total parameters: depend on variant (e.g., EfficientNet-B0 5,3 million). Total size: depend on variant (e.g., EfficientNet-B0 21,4 MB)	The search cost for grid search is high

2.6.2 Vision Transformers

Transformer was first applied to Natural Language Process (NLP) tasks (Vaswani et al. (2017)) and since then Transformer-based models have achieved significant breakthroughs in NLP and AI applications. Tom Brown (2020) Brown et al. (2020) pre-trained a massive transformer-base model called GPT-3 (short for Generative Pre-trained Transformer 3) using 175 billion parameters and achieved strong performance on different types of downstream natural language tasks without requiring fine-tuning. Given this remarkable performance, an increasing number of researchers are advocating for transformer-based models to improve visual tasks (Han et al. (2022), Jamil et al. (2023), Parvaiz et al. (2023), Li & Tanone (2022)).

In computer vision, CNN models have long been considered the benchmark for image classification tasks. However, there is growing evidence in recent literature indicating a potential shift in this paradigm, with the emergence of Vision Transformers (ViT) (Li & Tanone (2022)). The Vision Transformer applies a pure transformer directly to sequences of image patches, also called tokens, to classify the full image (Dosovitskiy et al. (2020)). The transformer learns by measuring the relationship between pairs of input tokens by applying a self-attention mechanism to extract intrinsic features (Parikh et al. (2016), Bahdanau et al. (2014)). The self-attention mechanism allows the model to weigh the significance of different input tokens relative to each other. The weights are determined dynamically based on the similarity between the tokens. Furthermore, this mechanism allows the model to focus more on relevant tokens while attending less to irrelevant ones, improving its ability to process sequences effectively and capture long-range dependencies and relationships within the input sequence and excessive computer power (Han et al. (2022), Jamil et al. (2023)).

In ViTs, architectures to track attention links between two embedded tokens take several steps. As illustrated in Figure 2.12, the primary step is to divide an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. Learnable embedding is applied to the sequence to provide a compact and semantically representation of input data to the model, in other words serves as the image representation that passes to each layer. The position embedding is essential because the self-attention layers are not able to inherently capture the positional information of the token in an image. Finally, to perform classification, the standard approach is to add an extra learnable "classification token" (*) to the sequence (Dosovitskiy et al. (2020), Li & Tanone (2022)). The Transformer encoder is made up of alternating layers of multiheaded self-attention (MSA) and multi-layer perceptron blocks (MLP). MLP blocks are fundamental components of neural network architectures. For each neuron, MLP applies a linear transformation (such as matrix multiplication) to its input followed by a non-linear activation function (such as ReLU or sigmoid) (Dosovitskiy et al. (2020)). Conversely, MSA is the key component that distinguishes transformer models from other architectures and greatly enhances their power (Han et al. (2022)).

The attention mechanism's holistic view of the image allows transformers to capture complex relationships between distant pixels or patches, a task that is challenging for CNN models which

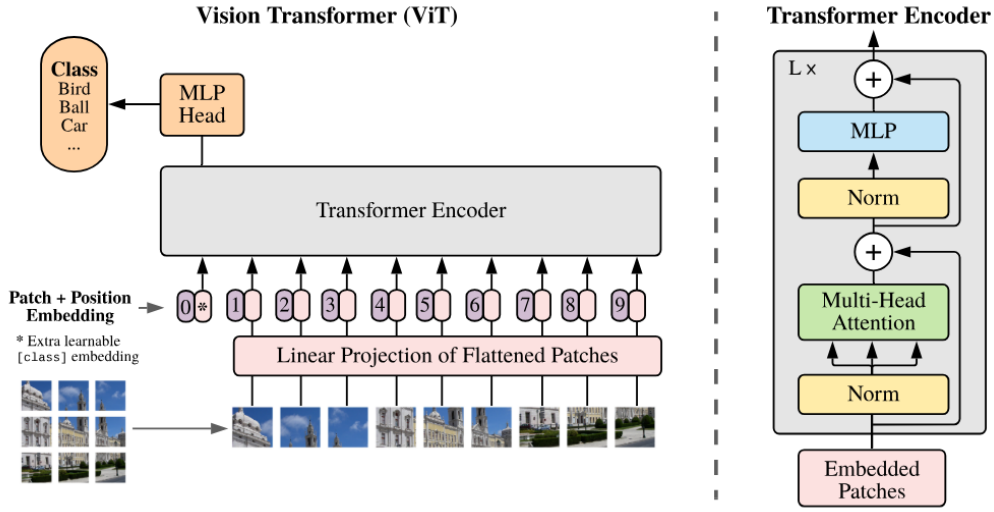

Figure 2.12: Vision transformer architecture; adapted from Dosovitskiy et al. (2020)

primarily attend to local characteristics (Dosovitskiy et al. (2020), Jamil et al. (2023)). Furthermore, convolution layers treat all image pixels equally, regardless of their importance. Despite the recognition of transformers for their efficacy in computer vision applications, they also come with certain disadvantages. Because transformers lack some inductive biases inherent to CNNs such as translation equivariance and locality, they do not generalise well when trained on insufficient amounts of data (Han et al. (2022)). However, the authors found that when pre-trained at sufficient scale, transformers achieve excellent results on tasks with fewer data points. As highlighted in studies (Dosovitskiy et al. (2020)), depends on a critical pre-training phase with large volumes of curated data. Hugo Touvron (2021) (Touvron et al. (2021)) addressed this challenge with a data-efficient training approach for image transformers, eliminating the need for extensive data sets. Their method, trained on ImageNet using a single computer in less than 3 days, achieved an impressive top-1 accuracy of 83.1% (single-crop) on ImageNet with no external data.

Additionally, in the ViT architecture, input images are divided into fixed-size patches, which are then linearly projected into high-dimensional vectors before being fed into the transformer layers, this linearization potentially disregards detailed local features within the patch. As a result, the original ViT architecture may not be optimized for tasks that require precise local feature extraction.

Researchers have proposed various solutions to address the need for enhanced modelling of local information. One approach involves employing a Hybrid Architecture, such as Convolutional Vision Transformers (CvT), which combines convolutional layers for local feature extraction in initial stages and transformer layers for capturing global dependencies in subsequent stages (Wu et al. (2020), Li & Tanone (2022)). In 2020, Bichen Wu (Wu et al. (2020)) adopted ResNet model as a convenient baseline extracting low-level features with convolutional layers and employing vision transformers to replace the last stage of convolutions. With an advanced training approach, ViT improves ResNet accuracy on ImageNet by 4.6 to 7 points and enhances semantic segmen-

tation performance by 0.35 points compared to convolutional FPN modules (Wu et al. (2020)). Aligning with these findings, research indicates that integrating convolutional layers with ViTs can yield superior outcomes in numerous computer vision applications compared to ViT models without CNN layers (Naseer et al. (2021), Coccomini et al. (2022)). Additionally, other studies affirm that using a CNN teacher provides better performance than using a transformer (Touvron et al. (2021)).

In the field of image classification, particularly in medical analysis, the effectiveness of feature extraction methods is extremely important, as is the preservation of relative positional information between different features (Parvaiz et al. (2023)). However, in CNN as the network gets deeper into its architecture there tends to be a decrease in the retention of this second feature. Nonetheless, researchers have demonstrated that replacing the convolutional layer with attention has improved performance (Bahdanau et al. (2014)) and research has been done using ViT to help detect diseases through radiologic examinations (Park & Kim (2022), Khaled et al. (2022)).

Alternatively, models like Tokens-to-Token ViT (TNT) introduce hierarchical structures within transformer layers. TNT divides patches into sub-patches and incorporates a novel transformer-in-transformer architecture. As shown in Figure 2.13, this variant of the Vision Transformer (ViT) architecture utilized the inner transformer block to model relationships between sub-patches and an outer transformer block for patch-level information exchange (Han et al. (2022), Touvron et al. (2021)). This means that in TNT the patches are viewed as visual sentences and one flow operates across the visual sentences and the other processes the visual words inside each sentence. This approach was proven to enhance the model's efficacy in capturing intricate local features in recent studies (Li & Tanone (2022), Luo et al. (2022)).

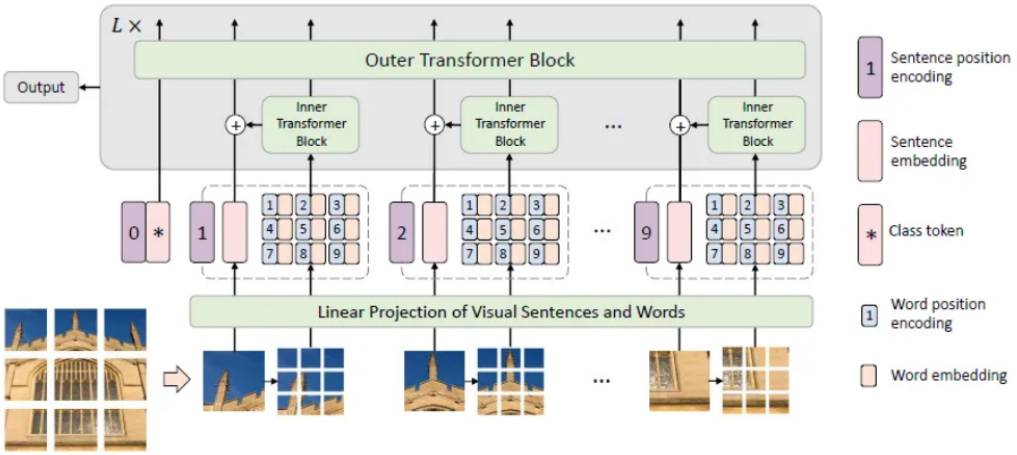

Figure 2.13: Transformer-iN-Transformer (TNT) framework; adapted from Han et al. (2022)

2.7 Related Work

This section provides an overview of studies that have successfully integrated Vision Transformer architectures with X-ray techniques in agricultural applications. Despite several initial attempts to

use transformers for image classification tasks (Chen et al. (2020)), their performance had, until recently, been inferior compared to convolutional networks (CNNs) in computer vision. However, hybrid architectures combining CNNs with self-attention mechanisms had already shown competitive results (Bello et al. (2019)).

In 2020, the Vision Transformer (ViT) architecture closed the gap with state-of-the-art performance on ImageNet for image classification using raw image patches as input. In October of that year, Dosovitskiy et al. introduced the ViT in the paper "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" (Dosovitskiy et al. (2020)). Since then, the ViT architecture has been fine-tuned for image recognition benchmarks and various downstream tasks, including pest detection in agricultural products and image classification tasks.

In 2021, Touvron et al. built upon the ViT model with their Data-efficient Image Transformers (DeiT) (Touvron et al. (2021)). This model, based on the ViT framework and improvements such as those in the PyTorch library (Wightman (2024)), reported a 6.3% improvement in top-1 accuracy over previous ViT models. DeiT introduced a distillation token to the input sequence, which interacts with class and patch tokens through self-attention, allowing the model to learn from a teacher network, typically a CNN. This distillation approach leveraged the inductive biases of CNNs to improve performance.

In 2022, Nijhawan et al. developed a hybrid approach for food image classification using a public dataset aimed at harmonizing food classification (Nijhawan et al. (2022)). This method utilized a Transformer architecture with self-attention to extract image patches, eliminating the need for convolution layers. Hand-crafted features were also computed, providing supplementary information for classification. The model achieved an impressive accuracy of 94.63%, specificity of 95.23%, sensitivity of 84.42%, and a kappa coefficient of 0.93, outperforming state-of-the-art food recognition systems using either hand-crafted or CNN-based features.

In 2022, Li et al. proposed an automatic pest identification method based on Vision Transformers, emphasizing its application in precision agriculture (Li & Tanone (2022)). To avoid overfitting, they applied various data enhancement techniques, such as Histogram Equalization, Laplacian, Gamma Transformation, CLAHE, Retinex-SSR, and Retinex-MSR. After training the ViT network on the enhanced dataset, they achieved a test recognition accuracy of 96.71% on the Plant_Village dataset, which was about 1.00% higher than the results using traditional CNN models like GoogleNet and EfficientNetV2.

In 2023, Hoang et al. introduced a novel approach combining image filtering and ViT learning for plant insect pest and disease recognition (Hoang & Pham (2023)). They transformed input images into the frequency domain and applied a trainable global dynamic filter kernel before passing the data to the ViT module. This preprocessing step, show in Figure 2.14 improved accuracy without significantly increasing the number of parameters or training time. Their model outperformed traditional CNN-based models in terms of accuracy, recall, specificity, and F1 score, with a remarkable 99.91% accuracy on a dataset of 2862 samples covering three pest categories on coffee plants.

Later in 2023, Tempelaere et al. conducted a study to identify a robust deep learning model

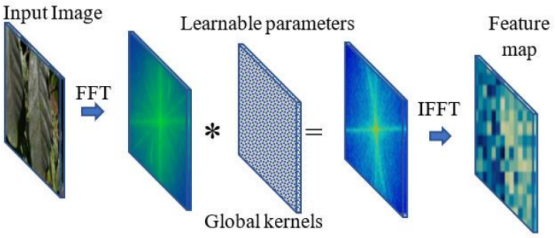

Figure 2.14: Demonstration of filter processing for high levels of feature map; adapted from Hoang & Pham (2023)

for X-ray sorting in non-destructive agricultural methods (Tempelaere et al. (2023)). A ViT-B/16 model with a 16x16 input patch size was tested using a learning rate of 10^{-5} and AdamW optimizer for 50 epochs. However, the model achieved an accuracy of only 73%, which was unsatisfactory compared to CNN models tested on the same dataset.

At the beginning of 2024, Fu et al. introduced an improved Vision Transformer model for automating the recognition of crop pests (Fu et al. (2024)). The dataset used consisted of 4,562 X-ray images of apple and grape leaves, where the model was tasked with identifying various types of pests and diseases. The model incorporates techniques aimed at capturing better spatial relationships and interactions between image features across different patches, allowing for more precise pest detection. One of the significant improvements includes optimizing the patch tokenization process to retain more relevant spatial features. Additionally, the ViT model was fine-tuned on the specific dataset of pest-infected leaves. The ViT achieved high classification accuracy and recall rates of 80-90% across most categories.

In the same year, Barman et al. developed an Android application for real-time tomato leaf disease detection (Barman et al. (2024)). They compared the Vision Transformer (ViT) model with Inception V3, a CNN-based model, for tomato leaf disease detection. They used a dataset containing 10,010 RGB images, which were divided into 10 categories, including both healthy and diseased tomato leaves. The ViT model significantly outperformed the CNN model, achieving an accuracy of 97.37% compared to 89.24% for Inception V3. The ViT model used was pre-trained on the ImageNet dataset and then fine-tuned on the specific dataset of tomato leaves. The application, ViT-SmartAgri, implements the pre-trained ViT model for real-time detection of tomato leaf diseases.

Additionally, Gao et al. proposed a data augmentation and feature enhancement method using ViT (AlsmViT) to distinguish similar food shapes with different nutritional values (Gao et al. (2024)). Tested on the Food-101 and Vireo Food-172 datasets, the AlsmViT-L model achieved accuracies of 95.17% and 94.29% respectively, demonstrating the power of ViT in precise food classification tasks.

Finally, in June 2024, Raza et al. introduced a Feature Enhancement Vision Transformer (FEViT) for classifying citrus fruits via X-ray CT scans (Raza et al. (2024)). The proposed FEViT model, which incorporates CNN blocks parallel to the Transformer encoder, achieved outstand-

ing performance, with the FEViTLarge-32 model obtaining an F1-score of 98.42%, accuracy of 99.25%, recall of 98.37%, and precision of 98.46%.

Chapter 3

Materials & Methods

This chapter provides a comprehensive overview of the materials and methods used in this thesis. The first two sections begin with a brief summary the X-ray CT acquisition and processing, and radiographic dataset simulation. In the next part, all the architectures selected for this thesis will be thoroughly explained. The next part outlines the hyperparameters, and techniques implemented to tailor the model to the specific challenges and requirements of the dataset used in this thesis. Finally, the last part of that section presents the performance analysis used.

3.1 X-ray CT scans of pear fruit

A total of 360 pears were divided into four groups of which three were systematically infested with different instar phases of codling moth (*Cydia pomonella*) Figure 3.1.

Figure 3.1: RGB images of cut-open pears corresponding to the healthy group (left), group L1-L2 and L3-L4 (middle) and group L5 (right). The larva is indicated with a red circle; adapted from He et al. (2024)

Afterwards, the CT dataset (16-bit) of pear fruit was acquired by a gantry CT system (SOMATOM DefinionFlash, Siemens, Germany) at UZ Leuven Hospital in Leuven, Belgium in a previous work (He et al. (2024)). The system operated at 100kV with a voxel size of 0.9766x0.9766x0.3000 mm^3. Afterwards, the CT data were processed in MATLAB (R2022b, The Mathworks, Natick, MA, USA). The volumes were resampled to a 0.9700x0.9700x0.9700 mm3 voxel size using cubic interpolation, the background was removed with a global Otsu threshold (Otsu et al. (1975)), and the pears were rotated to align their principal axis with the X-axis. Finally, the pear volumes were padded to the same size with background voxels, resulting in a volume of 224x224x224 voxels for

each pear (Figure 3.2). More details on the exact data acquisition, annotation workflow, and CT data pre-processing can be found in He et al. (2024).

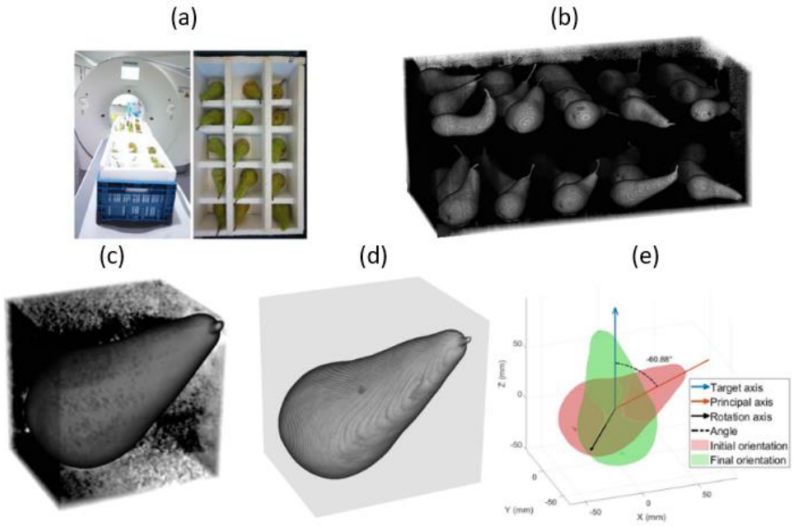

Figure 3.2: Visualizations of the results of each step of the pre-processing step. (a) The boxes containing the pears and styrofoam partitions on the CT X-ray system; (b) X-ray CT scan of the pears without the frame of the box; (c) Individual pear X-ray CT scan with noise; (d) Same X-ray CT without noise; (e) Rotation schematic followed to standardize the orientation of the pears; adapted from He et al. (2024)

3.2 Generation of Simulated Radiographic Datasets

Given the data-intensive requirements of deep learning, was decided to leverage the limited available 3D X-ray CT scans to simulate radiographs using a virtual X-ray system (He et al. (2024)). The projection process was performed using an ASTRA toolbox in MATLAB (Van Aarle et al. (2016)). The virtual X-ray system was configured with a 0.5 m distance between the X-ray source and the conveyor belt, and a 0.05 m distance between the conveyor belt and the line detector. The line detector had a width of 0.128 m and comprised 256 pixels. Additionally, the conveyor belt speed was set to 0.27 m/s, and the detector's sampling rate was adjusted to 540 Hz. To obtain radiography projections, pears were rotated to align the calyx-stem axis perpendicularly to the source-detector axis. Then it was placed in three orientations (0°, 60°, and -130°), each rotated nine times around the X-axis in a 30° rotation, producing 27 projections per pear. Furthermore, 12 pears did not develop infections and were discarded, resulting in a total of 9396 projections: 2430 for the control group and 6966 for the infested groups. All projections were 256x256 pixels. The work was done by (He et al. (2024)).

Subsequently, the created data set was randomly divided into train and test. The training had 90% of the data, with 10% of those images used for validation and 10% for test. The training phase had 278 pears, corresponding to 72 health and 206 infected. The validation and test sets had each 35 pears, corresponding to 9 healthy and 26 infected. The allocation of images into training

and testing sets was random, maintaining a clear distinction between the datasets to ensure the validity of the model's assessment. However, the random splits were performed on the sample level such that all projections of any pear were assigned to the same set. The dataset makes a distinction between radiographs corresponding to 'control' pears and 'infested' pears. For clarity and consistency throughout this thesis, the dataset described above will be referred to as the "peardataset."

3.3 Network architecture

The Vision Transformer (ViT) has recently emerged as a strong alternative to CNNs in computer vision, particularly excelling with large datasets (Dosovitskiy et al. (2020)). In this thesis, two methods were employed to implement the ViT architecture on the peardataset. The first method used and modified the Vision Transformer (ViT) model with weights pretrained on ImageNet dataset from torchvision. The second method involved developing a custom ViT architecture from scratch using the PyTorch framework, enabling adjustments to the model's architecture to better match the complexity and specific requirements of the peardataset.

3.3.1 ViT models from torchvision

To identify a robust classification model, various Vision Transformer architectures were evaluated, including ViT-B/16, ViT-B/32, ViT-L/16, and ViT-L/32. Table 3.1, presents the top-1 and top-5 accuracy performance of these architectures when trained on the expansive ImageNet-1k dataset.

Table 3.1: Overview of different architectures and their respective performance on the ImageNet-1K dataset, accuracy in (%), adapted from Sayak (2021)

	TOP-1 Accuracy	**TOP-5 Accuracy**	**Trainable Parameters**	**Size Patches**
ViT-B/16	81.0	95.3	86,859 496	16x16
ViT-B/32	75.9	92.5	88,297 192	32x32
ViT-L/16	79.7	94.6	304,715 752	16x16
ViT-L/32	77.0	94.6	306,632 680	32x32

The four different architectures discussed are part of the Vision Transformer (ViT) neural network family introduced by (Dosovitskiy et al. (2020)). The models "ViT-B" (Vision Transformer Base) and "ViT-L" (Vision Transformer Large) denote different variants primarily distinguished by size and complexity. On the other hand between ViT/16 and ViT/32, the differences are the patch size, with 16x16 and 32x32 pixel patches, respectively.

The ViT-B/16 and ViT-B/32 models were both chosen for this study to analyze how patch size influences model performance. The ViT-B/16 model was selected for its balanced trade-off between accuracy and computational efficiency, with 86,567,656 trainable parameters, as shown in Table 3.1. On the other hand, the ViT-B/32 model, with its larger 32x32 pixel patches, was

included to explore how different patch sizes affect the model's ability to capture fine-grained details versus broader contextual information (Dosovitskiy et al. (2020)).

Larger models, such as the ViT-L variants, were deliberately not chosen due to concerns about their complexity relative to the size of the dataset. Given the dataset's moderate size, there was a significant risk that using a larger model could lead to overfitting, reducing the model's ability to generalize to new data. Thus, the ViT-B/16 and ViT-B/32 models were deemed more appropriate, offering a good balance between model complexity and the available data. Additionally, three different simplified models were explored, where certain blocks were removed to further optimize performance. Further explanation of this simplified approach is provided in this section.

Before integrating these models into the training pipeline, several adjustments were made. For all models, the conventional three-channel input, representing RGB data, was modified to a single-channel input to utilize grayscale images. Additionally, the final layer of each architecture was adjusted to output a single node for binary classification. Furthermore, the original projection layer was replaced with a convolutional layer to make the model more efficient in capturing fine details (Mao et al. (2022)). The pre-trained ViT-B/16 model had **85,406,209 trainable parameters** with a kernel size of 16x16 and a stride of 16x16, while the ViT-B/32 model was adjusted to have a kernel size and stride of 32x32 to align with its architectural design with **85,883,137 trainable parameters**. Both models were employed using PyTorch framework, specifically version 1.13.1.

3.3.2 CustomViT model

To better align the model's complexity with the dataset's characteristics, a custom Vision Transformer model, referred to as "CustomViT," was developed. This custom model allowed for precise adjustments to the architecture, including the number of attention heads, layers, and depth, tailored to the specific needs of the dataset. The CustomViT model was built using the PyTorch framework, specifically version 1.13.1, with a single-channel input, and a single output node for binary classification. To better understand the differences between the two methods, the table 3.2 shows an overview of the principle differences between the fine-tunned models and the from-scratch ViT model.

3.4 Training, validation and testing the model

The training, validation and test were performed on a Windows system with a Xeon W-2275 4.60 GHz processor (Intel, Santa Clara, CA, USA), a 16 GB NVIDIA Quadro RTX 5000 GPU (NVIDIA, Santa Clara, CA, USA) and 256 GB of RAM. Each training cycle was performed for 60 or 200 epochs with an epoch corresponding to a full iteration over all training set samples. Following each epoch in the training cycle, an extra iteration was performed to evaluate the model's performance using the validation set. A weighted binary cross-entropy loss function was employed to address class imbalance in the binary classification model (25% control, 75% infested). The GeLU (Gaussian Error Linear Unit) activation function was used, and class weights were calculated based on the ratio of the majority to the minority class to ensure balanced learning.

Table 3.2: Architectures Overview for the three architectures used in this thesis

Architecture	**ViT-B/16**	**ViT-B/32**	**CustomViT**
Method	Fine-Tuning	Fine-Tuning	From Scratch
Patch Size	16x16	32x32	-
Embedding Dim	768	768	-
Depth	12	12	-
Heads	12	12	-
MLP Dimension	3072	3072	-
Initial Layer	2D Convolution Layer	2D Convolution Layer	Patch Extraction
Final Layer	1 channel	1 channel	1 channel
Initial Weights	ImageNet-1K	ImageNet-1K	Random Initialized
Activation Function	GELU function	GELU function	GELU function
Loss Function	Binary Cross Entropy with Logits	Binary Cross Entropy with Logits	Binary Cross Entropy with Logits

To further enhance model performance, transfer learning was employed with the ViT-B/16 and ViT-B/32 models. All trainable parameters were initialized from models pre-trained on the ImageNet dataset and then fine-tuned on the current dataset, except for the customVit model, where the weights were initialized randomly (Yosinski et al. (2014)). Additionally, data augmentation techniques were applied to the training data to mitigate overfitting (Chen et al. (2021)). These techniques included horizontal and vertical flipping with a probability of 0.5.

3.4.1 Architectures and Hyperparameters

During the training phase of all models, gradients are calculated through backpropagation and updated using the AdamW optimizer. AdamW was chosen for its ability to handle sparse gradients and varying parameter scales, common in deep learning models. By combining momentum and adaptive learning rate techniques with decoupled weight decay, AdamW facilitates faster convergence and more efficient handling of noisy gradients (Loshchilov (2017)). Additionally, its fast initial convergence, similar to Adam, provides an advantage over non-adaptive optimization methods (e.g., SGD), at the cost of possibly lower final accuracy (Wilson et al. (2017)).

Complementing the AdamW optimizer, a learning rate scheduler was implemented to dynamically adjust the learning rate based on validation performance. Specifically, if the validation accuracy plateaued for 10 consecutive epochs, the scheduler reduced the learning rate by a factor of 0.1, with a lower bound set at 10^{-8}. This mechanism ensured that the model remained responsive to learning while avoiding the risks of overshooting optimal solutions.

3.4.1.1 ViT models with different transfer learning strategies

In the context of optimizing pre-trained ViT models, three transfer learning strategies were used.

In the first method, the original ViT-B model, which consists of 12 encoder layers, was simplified by retaining only the first two transformer layers, while replacing the remaining layers with Identity layers that simply passed the input directly to the output. Despite this reduction in active layers, the model still preserved its high-dimensional embedding space and large MLP expansion ratio, resulting in **15,003,649** trainable parameters.

The second method was divided into two distinct approaches. The first, Selective Fine-Tuning, involved fine-tuning only the first and last layers of the ViT-B model while freezing all other layers to retain the pre-trained weights. This approach updated just **198,145** parameters, concentrating on the convolutional layer and the classification head. In the second approach, Extended fine tuning was applied. Here, both the first layer and the entire 12th transformer layer, with the final layer normalization and classification head, were unfrozen for fine-tuning. This allowed the model to better adapt to the new task by fine-tuning **7,877,377** parameters to effectively balance the retention of pre-trained knowledge with task-specific learning.

3.4.1.2 CustomViT model

In developing the CustomViT model, due to the large search space and the arbitrariness of finding promising combinations manually, hyperparameter optimization can be a tedious process. To address this problem, an optimization framework designed for hyperparameters optimization called Optuna was implemented. Optuna offers two key advantages for hyperparameter optimization: a flexible sampling strategy that supports discrete, continuous, and categorical distributions, and a dynamic pruning strategy that efficiently halts unpromising trials early. This ensures that computational resources are concentrated on the most promising hyperparameter configurations, which is particularly beneficial for complex models like the ViT where each trial can be computationally expensive (Akiba et al. (2019)).

For this thesis, Optuna was used to optimize several hyperparameters of the CustomViT model, including the patch size, dimension of the embedding space, depth of the transformer, number of attention heads, the dimension of the MLP head, learning rate, and batch size. The specific ranges for these hyperparameters were chosen based on typical values used in the literature: patch size: 16, 32, dimension of embedding space: 128 to 512, depth: 2 to 10, number of heads: 2 to 8, MLP dimension: 256 to 1024, learning rate: log-uniform distribution between 10^{-5} and 10^{-3}, and batch size: 16, 32. The objective function for the optimization process was the validation accuracy of the model, which was minimized across 50 trials, with each trial consisting of training the CustomViT model for 60 epochs. The AdamW optimizer was used for training, along with a learning rate scheduler that reduced the learning rate when the validation accuracy did not improve for 10 consecutive epochs. The final results were saved and exported for further analysis, allowing for the selection of the best-performing hyperparameter configuration. This automated hyperparameter tuning process provided a systematic and efficient approach to identifying the most effective combination of hyperparameters, ultimately leading to improved model performance while minimizing the manual effort required.

3.5 Robustness and Sensitivity metric

In this study, Loss-SENS and LossGrad-SENS metrics, proposed by Yeh et al. in an article titled "On the Importance of Gradient-Based Attribution for Understanding Model Robustness", were employed to assess the robustness of the optimal ViT model (Heo et al. (2023), Finlay & Oberman (2021)).

Loss-SENS measures the sensitivity of the model's loss to perturbations in the input, where lower values indicate higher resilience against noise and adversarial attacks. **LossGrad-SENS** focuses on the stability of the gradients, assessing how the gradients of the loss function change under noisy conditions. A lower LossGrad-SENS value suggests smooth and consistent gradients, signifying that the model is less vulnerable to adversarial perturbations and noise, enhancing its robustness.

$$\text{Loss-SENS}(\mathscr{L}, f, \gamma, x, y) = \max_{\|\delta\| \leq \gamma} |\mathscr{L}(f(x+\delta), y) - \mathscr{L}(f(x), y)| \tag{3.1}$$

$$\text{LossGrad-SENS}(\nabla_x \mathscr{L}, f, \gamma, x, y) = \max_{\|\delta\| \leq \gamma} \frac{\|\nabla_x \mathscr{L}(f, x+\delta, y) - \nabla_x \mathscr{L}(f, x, y)\|}{\|\nabla_x \mathscr{L}(f, x, y)\|} \tag{3.2}$$

where, $\mathscr{L}$ represents the loss function, f refers to the target model that processes the input x, where y denotes the true label associated with x, δ indicates the perturbation or noise applied to the input, constrained by the adversarial budget γ, $\nabla_x \mathscr{L}$ stands for the gradient of the loss concerning the input x, which helps in assessing the model's sensitivity to input noise

These metrics enable a comprehensive evaluation of the model's performance beyond accuracy, examining how predictions and gradient behavior remain stable under various conditions. This approach addresses limitations of traditional accuracy metrics by offering a deeper understanding of model robustness and sensitivity to perturbations (Heo et al. (2023)).

3.6 Test Evaluation

Confusion Matrix

In AI, especially in classification tasks, a confusion matrix can directly reflect the classification results of the model. It is presented in the form of a table, which shows the prediction results of each category item from a vertical view and the actual category of each category item from a horizontal view (Fu et al. (2024)). This tool easily provides the values of key indicators: True Positives (TP), True Negatives (TN), False Negatives (FN), and False Positives (FP).

Correctly identified positive (infected) and negative (healthy) cases are referred to as true positives (TP) and true negatives (TN), respectively. Conversely, incorrect predictions are categorised as false negatives (FN) when a positive case is predicted as negative, and false positives (FP) when

a negative case is predicted as positive 3.3. This table is used in classification problems to assess

Table 3.3: Confusion Matrix

		Prediction	
		Negative	**Positive**
Actual	**Negative**	True negative (TN)	False positive (FP)
	Positive	False negative (FN)	True positive (TP)

where errors in the model were made. With the values from the confusion matrix is possible to calculate various metrics. The Accuracy measures how often the model is correct (3.3). The precision of the positives predicted, tells what percentage is truly positive (3.4). The Sensitivity (also called Recall) measures how good the model is at predicting positives (3.5). Finally, the F-score is the "harmonic mean" of precision and sensitivity. It considers both false positive and false negative cases (3.6) (Sanchez et al. (2021)). Given the imbalanced nature of the dataset, where one class significantly outnumbers the other, it was necessary to calculate the balanced accuracy to provide more coherent results (3.7). Balanced accuracy adjusts for the imbalance by averaging the recall obtained in each class, thus giving a more accurate measure of the model's performance across both classes (Pedregosa et al. (2023)).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3.3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.5}$$

$$\text{F1 score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{3.6}$$

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}}\right) \tag{3.7}$$

Chapter 4

Results

This chapter presents and analyzes the experimental work aimed at developing a ViT classifier to automatically classify X-ray radiographs of pears as infected or healthy. Two ViT training methods were explored: fine-tuning a pre-trained model and a custom ViT model was optimized using the Optuna framework. Additionally, for the fine-tuning models was also training only the first and last layers while freezing others, and simplifying the model by retaining only initial transformer layers. The results of these approaches, including robustness metrics and the standard balanced accuracy, F1 score, precision and recall, are presented.

4.1 Fine-tuning Model

4.1.1 Hyperparameter tuning & Outcomes

Hyperparameter tuning plays a crucial role in optimising machine learning models for better performance (Jin (2022)). Fine-tuning as a special case of transfer learning aims to improve the performance on the target data by transferring the knowledge from a large-scale pre-training data to a target domain. It involves using the weights from a pre-trained model as the starting point for the new task, which helps the model adapt more quickly and effectively to the new data (Liu et al. (2022)).

In this study, as described in section 3.3, ViT-B/16 and ViT-b/32 were fine-tuned on the pear-dataset for 60 epochs, using the fixed training and validation set, while varying the batch size (16, 32) and the initial learning rate ($10^{-3}, 10^{-4}, 10^{-5}$). The evolution of the loss over the epochs during training was compared between the different model architectures and hyperparameters. More specifically, both the final loss value as well as how fast the loss stabilizes to reach that final value were compared. This comparison showed the following optimal hyperparameters: batch size and initial learning rate are 32 and 10^{-4} for ViT-B/16, and 16 and 10^{-4} in the case of ViT-B/32, respectively. The loss plots of these two models are depicted in Figure 4.1. This Figure also shows that, both architectures are showing signs of overfitting.

Those two models, trained with their optimal hyperparameters, were evaluated on the independent test dataset according to their accuracy, balanced accuracy, recall, precision and F1-score,

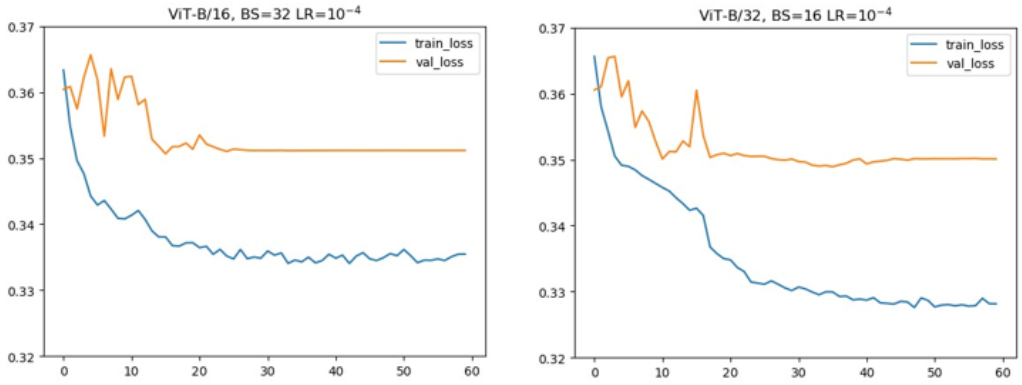

Figure 4.1: Evolution of loss over the epochs during training for the ViT-B/16 (left) and ViT-B/32 (right) model with the optimal hyperparameters: batch size of 32 and initial learning rate of 10^{-4} for ViT-B/16, and batch size of 16 and initial learning rate 10^{-4} in the case of ViT-B/32

shown in table 4.1. The ViT-B/16 and ViT-B/32 models reach a final balanced accuracy of respectively 71,5% and 73,2%. While the ViT-B/32 model demonstrated slightly better prediction performance, these results are below the expected benchmarks based on values reported in the table 3.1.

Table 4.1: Model Performance Metrics (in %) of the trained ViT-B/16 and ViT-B/32 model with the optimal hyperparameters: initial learning rate and batch size of respectively 32 and 10^{-4} for ViT-B/16, and 16 and 10^{-4} in the case of ViT-B/32 along 60 epochs. (TN, True Negative; FP, False Positive;TP, True Positive; FN, False Negative; Acc, Accuracy; Bal. Acc, Balanced Accuracy)

Model	**Acc**	**Bal. Acc**	**Precision**	**Recall**	**F1 scores**	**TN**	**TP**	**FP**	**FN**
ViT-B/16	70.6	71.5	88.3	69.7	77.9	73.25	69.66	26.75	30.34
ViT-B/32	73.2	73.2	88.8	73.2	80.2	73.25	73.22	26.75	26.78

4.1.2 Modify Architecture

In response to initial signs of sub-optimal performance, modifying the architecture of pre-trained models was necessary to reduce complexity and enhance adaptability to the peardataset

The ViT-B/16 and ViT-B/32 models were modified by replacing certain encoder layers with identity layers, as described in section 3.3. This modification aimed to reduce the complexity of the models while preserving the initialized weights from ImageNet-1K. The models were trained for 60 epochs, using fixed training and validation sets, with the same batch size and learning rate combinations as in the fine-tuning experiments. The evolution of the loss over the epochs was tracked and compared across different configurations. Specifically, we evaluated how the reduction in model complexity impacted the final loss value and the rate at which the loss stabilized. The analysis identified the optimal configuration concerning both architectures, leading to the selection of batch size 32 and learning rate 10^{-3} for ViT-B/16, and batch size 16 and learning rate 10^{-3} for ViT-B/32. The corresponding loss plots for these models are shown in Figure 4.2.

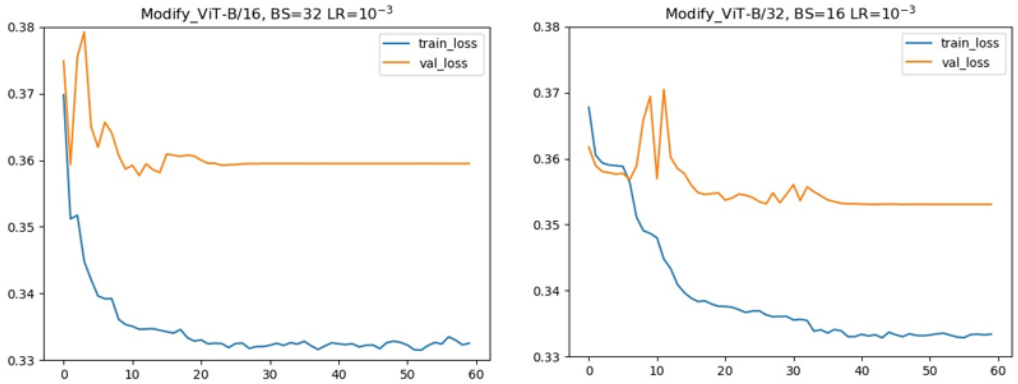

Figure 4.2: Evolution of loss over the epochs during training for the Modify ViT-B/16 (left) and Modify ViT-B/32 (right) model with the optimal hyperparameters: batch size of 32 and initial learning rate of 10^{-3} for ViT-B/16, and batch size of 16 and initial learning rate 10^{-3} in the case of ViT-B/32, along 60 epochs

Those two models, trained with their optimal hyperparameters, were evaluated on the independent test dataset according to their accuracy, balanced accuracy, recall, precision and F1-score, shown in table 4.2. The Modify ViT-B/16 and Modify ViT-B/32 models reach a final balanced accuracy of respectively 70,8% and 73,1%. This shows a similar performance as the previous fine-tuning pre-trained models.

Table 4.2: Model Performance Metrics (in %) of the Modify ViT-B/16 and Modify ViT-B/32 models. (TN, True Negative; FP, False Positive; TP, True Positive; FN, False Negative; Acc, Accuracy; Bal. Acc, Balanced Accuracy)

Model	**Acc**	**Bal. Acc**	**Precision**	**Recall**	**F1 scores**	**TN**	**TP**	**FP**	**FN**
Modify ViT-B/16	70.8	69.3	86.1	72.4	78.6	66.26	72.36	33.74	27.64
Modify ViT-B/32	73.1	69.5	85.4	76.9	81.0	62.14	76.92	37.86	23.08

4.1.3 ViT models as feature extractors

In another approach, selective unfreezing of layers within the pre-trained models was employed to further refine their adaptability to the peardataset. As outlined in Section 3.3, this method involved keeping most layers frozen while selectively unfreezing and fine-tuning specific layers, such as the first and last layers, and in another configuration, an additional deeper transformer layer.

First and last layer

The models with only the first and last layer unfrozen were initially trained for 60 epochs with various combinations of batch size (16, 32) and initial learning rate ($10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$). For models with the lowest initial learning rate (10^{-6}), the validation curve did not stabilise, prompting an extension of training to 200 epochs for both batch size combinations. The resulting loss values

and stabilisation rates were analyzed to determine the most effective unfreezing strategy. For the ViT-B/16 model, the two best configurations were: batch size 32, learning rate 10^{-5}, and 60 epochs, which achieved higher precision and balanced accuracy; and batch size 16, learning rate 10^{-6}, and 200 epochs, which showed slightly better accuracy, recall, and F1 score, as shown in table 4.3 and Figure 4.3.

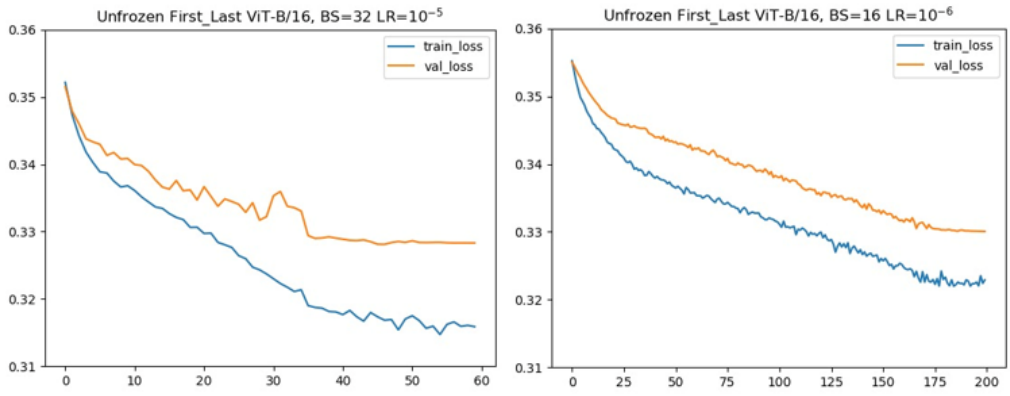

Figure 4.3: Evolution of loss over the epochs during training for the Unfrozen First_last ViT-B/16 BS=32, LR=10^{-5}, 60 epochs(left) and Unfrozen First_last ViT-B/16 BS=16, LR=10^{-6}, 200 epochs (right)

Table 4.3: Model Performance Metrics (in %) of the optimal Unfrozen First_last ViT-B/16 models. (TN, True Negative; FP, False Positive; TP, True Positive; FN, False Negative; Acc, Accuracy; Bal. Acc, Balanced Accuracy)

Unfrozen First_last ViT-B/16	**Acc**	**Bal. Acc**	**Precision**	**Recall**	**F1 scores**	**TN**	**TP**	**FP**	**FN**
BS:32 LR:10^{-5} 60	68.1	72.9	91.3	63.1	74.6	82.72	63.11	17.28	36.89
BS:16 LR:10^{-6} 200	68.6	72.5	90.6	64.4	75.3	80.66	64.39	19.34	35.61

For the model Unfrozen First_last ViT-B/16, while the model 1 (BS:32, LR:10^{-5}, 60 epochs), has slightly better precision and balanced accuracy, the differences are minimal. Given the similar test performance metrics across both models, the model 2 (BS:16, LR:10^{-6}, 200 epochs), with fewer signs of overfitting (Figure 4.3) is likely to perform more consistently on unseen data. Therefore, prioritizing Model 2 for its better generalization makes sense in this context (Goodfellow et al. (2016)).

The optimal hyperparameters were identified as a batch size of 16, an initial learning rate of 10^{-6} and number epochs 200 for Unfrozen First_last ViT-B/16, and a batch size of 16, an initial learning rate of 10^{-6} and number epochs 200 for Unfrozen First_last ViT-B/32. The corresponding loss plots for these models are shown in Figure 4.4.

The two models, each trained with their optimal hyperparameters, were assessed on the independent test dataset, with evaluation metrics including accuracy, balanced accuracy, recall, precision, and F1-score, as detailed in Table 4.4. The Unfrozen First_last ViT-B/16 and Unfrozen First_last ViT-B/32 models achieved final balanced accuracies of 72.5% and 72.2%, respectively.

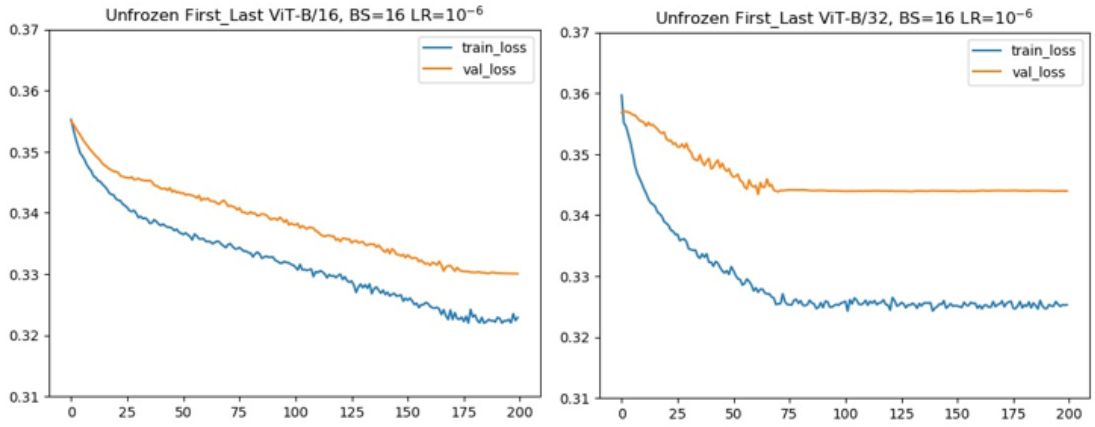

Figure 4.4: Evolution of loss over the epochs during training for the Unfrozen First_last ViT-B/16 (left) and Unfrozen First_last ViT-B/32 (right) model with the optimal hyperparameters: batch size of 16, an initial learning rate of 10^{-6} and 200 epochs for both models

When compared to the previously obtained results, selectively unfreezing the first and last layers can yield competitive performance, especially in the ViT-B/16 model, while the ViT-B/32 model shows a slight trade-off in balanced accuracy.

Table 4.4: Model Performance Metrics (in %) of the Unfrozen First_last ViT-B/16 and Unfrozen First_last ViT-B/32 models with the optimal hyperparameters: batch size of 16, initial learning rate of 10^{-6} and number epochs 200 for ViT-B/16, and batch size of 16, an initial learning rate of 10^{-6} and number epochs 200 in the case of ViT-B/32. (TN, True Negative; FP, False Positive; TP, True Positive; FN, False Negative; Acc, Accuracy; Bal. Acc, Balanced Accuracy)

Unfrozen First_last Model	**Acc**	**Bal. Acc**	**Precision**	**Recall**	**F1 scores**	**TN**	**TP**	**FP**	**FN**
ViT-B/16	68.6	72.5	90.6	64.4	75.3	80.66	64.39	19.34	35.61
ViT-B/32	69.7	72.2	89.5	67.1	76.7	77.37	67.09	22.63	32.91

First layer and Last Transformer layers

The models with only the first and last transformer layers unfrozen were initially trained for 60 epochs using various combinations of batch size (16, 32) and initial learning rates (10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}). Similar to the previous models, those trained with the lowest initial learning rate (10^{-6}) did not show stabilization in the validation curve within the initial 60 epochs. This necessitated extending the training to 200 epochs for both batch-size configurations. The loss values and stabilisation rates from these training sessions were analyzed to determine the most effective strategy for unfreezing and fine-tuning the transformer layers.

The optimal hyperparameters for the Unfrozen First and Last_Transformer layer ViT-B/16 were identified as a batch size of 32, an initial learning rate of 10^{-6}, and 200 epochs. For the Unfrozen First and last_Transformer layer ViT-B/32, the optimal configuration was determined to

be a batch size of 32, an initial learning rate of 10^{-5}, and 60 epochs. The corresponding loss plots for these models are shown in Figure 4.5.

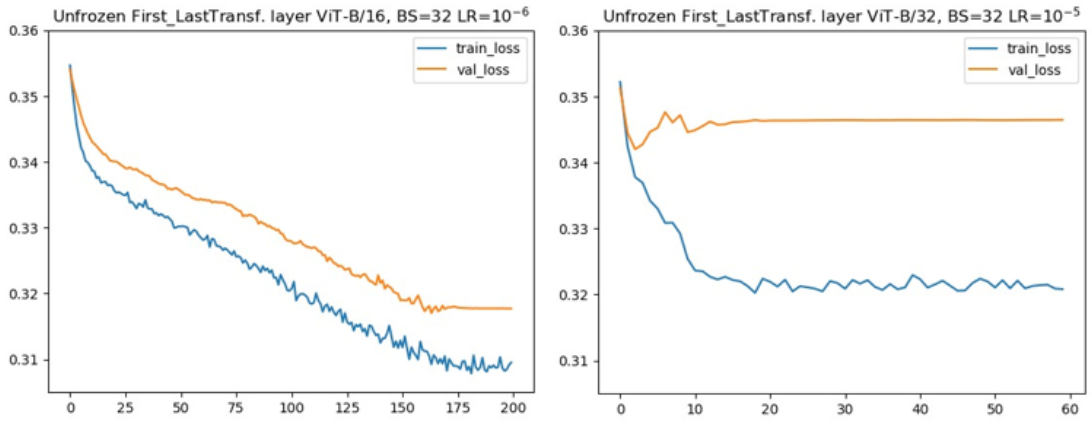

Figure 4.5: Evolution of loss over the epochs during training for the Unfrozen First and Last_Transformer layer ViT-B/16 (left) and Unfrozen First and Last_Transformer layer ViT-B/32 (right) model with the optimal hyperparameters: batch size of 32, an initial learning rate of 10^{-6} and 200 epochs for ViT-B/16, and batch size of 32, an initial learning rate of 10^{-5} and 60 epochs for the ViT-B/32

Next, the independent test dataset was used to evaluate the two models, each trained with their optimal hyperparameters, with metrics including accuracy, balanced accuracy, recall, precision, and F1-score. The Unfrozen First and Last Transformer ViT-B/16 model achieved a final balanced accuracy of 72.8%, while the ViT-B/32 model reached 68.7%. These results are demonstrated in Table 4.5.

Table 4.5: Model Performance Metrics (in %) of the Unfrozen First Last Transformer ViT-B/16 and Unfrozen First Last Transformer ViT-B/32 models with the optimal hyperparameters: batch size of 32, initial learning rate of 10^{-6} and number epochs 200 for ViT-B/16, and batch size of 32, an initial learning rate of 10^{-5} and number epochs 60 in the case of ViT-B/32. (TN, True Negative; FP, False Positive; TP, True Positive; FN, False Negative; Acc, Accuracy; Bal. Acc, Balanced Accuracy)

Unfrozen First_Last Transformer Model	**Acc**	**Bal. Acc**	**Precision**	**Recall**	**F1 scores**	**TN**	**TP**	**FP**	**FN**
ViT-B/16	67.7	72.8	91.4	62.4	74.2	83.13	62.39	16.87	37.61
ViT-B/32	66.1	68.7	87.6	63.4	73.6	74.07	63.39	25.93	36.61

4.2 CustomViT model

To analyse the impact of hyperparameters on the performance of the ViT models and discuss the best combination of hyperparameters, a hyperparameters optimization framework, Optuna, was used. The framework examined seven different hyperparameters: the patch size (16, 32), embedding dimension (128 to 512, with step of 128), number of layers(2 to 10 with step of 2), number of attention heads(2 to 8 with step of 2), hidden layer size (256 to 1024, step of 256), initial learning rate (10^{-5} to 10^{-3}), and batch size (16,32), more information in 3.4.1.2.

Figure 4.6 displays the results of the hyperparameter optimization search conducted using Optuna. This parallel coordinate plot consists of eight axes, with the leftmost axis representing the validation accuracy, and the other seven axes corresponding to the examined hyperparameters. Each line in the plot represents a completed run, connecting the specific values of the hyperparameters on the different axes. The colour of the lines corresponds to the validation accuracy, with darker lines indicating better performance. Unlike the initial example where some runs were pruned, all completed runs are represented, giving a comprehensive overview of the hyperparameter search space and its impact on model performance.

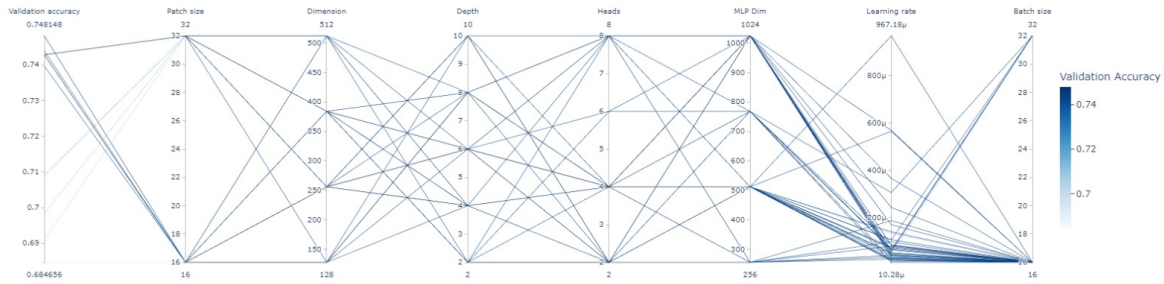

Figure 4.6: Parallel coordinate plot showing the different completed runs, the corresponding hyperparameters and the validation accuracy; with initial learning rate 10^{-3} corresponding to 1000µ

To clarify the optimal hyperparameters, an individual scatter plot was created for each parameter, as shown in Figure 4.7. In these plots, each dot corresponds to a completed run, allowing for a clearer and more focused examination of how specific hyperparameter values impact model performance.

Based on the two figures, certain hyperparameters clearly stand out by producing better validation accuracy compared to others. For the patch size, the 16x16 configuration consistently showed better performance in most runs. The number of attention heads also plays a crucial role, with 4 heads performing well and demonstrating good generalization across trials. When it comes to the model's depth, a configuration of 6 encoder layers produced the best results, providing enough capacity to capture complex patterns while avoiding overfitting or excessive computational burden associated with deeper models. Analysis of the learning rate showed that lower learning rates were beneficial for the model, confirming the established understanding that smaller learning rates contribute to smoother convergence and better optimization, particularly in transformer models.

The Mlp-dim results were more varied. While 1024 yielded some of the highest accuracy scores, a substantial number of good results were also achieved with a dimension of 512.

Evaluating individual hyperparameters in isolation, such as Mlp-dim, can provide some insights but is not the most effective approach for identifying the optimal combination of settings. Figure 4.6 clearly demonstrates that the highest-performing models result from a careful combination of multiple hyperparameters. Since these parameters interact in complex ways, tuning them together is essential for finding the best configuration. To validate this analysis, the two models with the highest validation accuracy were selected and tested on the test set to assess their performance, the hyperparameters values are represented in Table 4.6. The evolution of the loss in both the training and validation phases was tracked over the course of 60 epochs to ensure proper convergence and generalization. This thorough approach ensures that the selected hyperparameters lead to robust and high-performing models across different datasets.

Table 4.6: Values of the two best combinations of hyperparameters in a CustomViT model

Trial number	**Validation Acc**	**Batch size**	**Dim**	**Heads**	**LR**	**Mlp Dim**	**Patch size**	**Number parameters**
49	74.8	16	256	4	8×10^{-5}	1024	16	5114625
23	74.4	16	128	4	7.5×10^{-5}	512	16	17077905

The two models, each trained with their optimal hyperparameters, were assessed on the independent test dataset, with evaluation metrics including accuracy, balanced accuracy, recall, precision, and F1-score, as detailed in Table 4.7.

Table 4.7: Performance metrics of two different models for the CustomViT architecture (Trial numbers 49 and 23) models evaluated in a randomly initialized base network

Model	**Acc**	**Balanced Acc**	**Precision**	**Recall**	**F1 score**	**TP**	**TN**	**FP**	**FN**
Trial number 49	68.1	54.3	76.3	82.8	79.4	25.93	82.76	74.07	17.24
Trial number 23	63.6	62.6	82.5	64.7	72.5	60.49	60.67	39.51	35.33

4.3 Prediction across Infected Groups

Although the classification task was binary, the dataset consisted of three distinct infected groups: L1, L3, and L5. For each group, the number of true and false predictions was calculated for both the optimal pre-trained ViT and custom ViT models. It is important to note that the test set, based on the random seed used in this thesis, consistently comprised 243 samples from L1 group, 270 samples from L3 group, and 189 samples from L5 group. The results for each model are shown below:

Table 4.8: Comparison of True Negative and False Positive rates across three infected groups (L1, L3, L5) for optimal pre-trained ViT models

Model	**L1**	**L3**	**L5**	**L1**	**L3**	**L5**
	True Negative			**False Positive**		
ViT-B/16	32.1%	42.1%	25.8%	40.4%	30.0%	29.6%
ViT-B/32	35.2%	39.0%	25.9%	33.0%	37.2%	29.8%
Modify ViT-B/16	33.3%	41.3%	25.4%	38.2%	30.9%	30.9%
Modify ViT-B/32	34.1%	39.4%	26.5%	36.4%	35.2%	28.4%
Unf. First_Last ViT-B/16	28.5%	43.1%	28.3%	45.6%	30.0%	24.4%
Unf. First_Last ViT-B/32	31.0%	41.4%	27.6%	42.0%	32.5%	20.5%
Unf. First_Last Transf. ViT-B/16	29.2%	40.0%	30.8%	43.5%	36.0%	20.5%
Unf. First_Last Transf. ViT-B/16	27.0%	44.0%	29.0%	47.9%	28.8%	23.3%

Table 4.9: Comparison of True Negative and False Positive rates across models.

Model	**L1**	**L3**	**L5**	**L1**	**L3**	**L5**
	True Negative			**False Positive**		
Unf. First_Last Transf. ViT-B/16	29.2%	40.0%	30.8%	43.5%	36.0%	20.5%
Unf. First_Last Transf. ViT-B/16	27.0%	44.0%	29.0%	47.9%	28.8%	23.3%

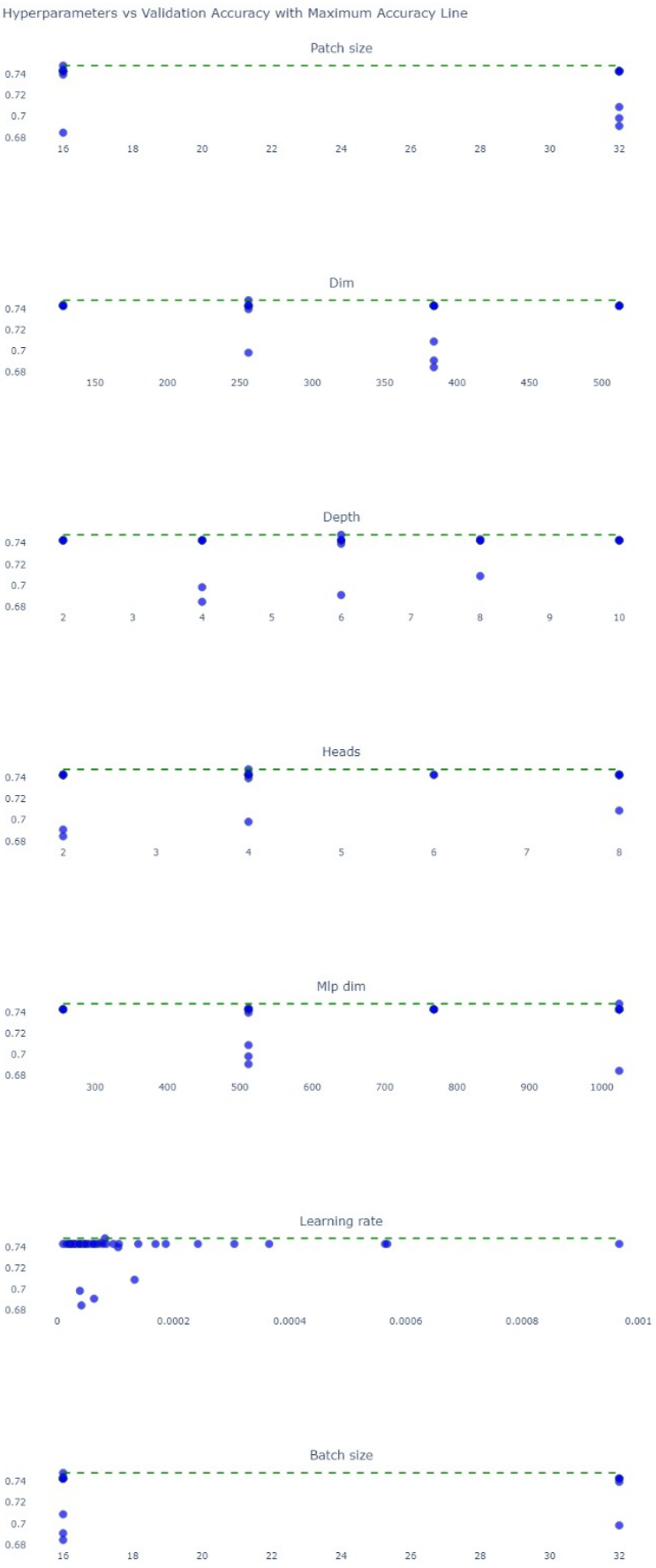

Figure 4.7: Scatter plot showing the different completed runs, the corresponding hyperparameters and the validation accuracy. Each blue dot represents the result of a complete run.

4.4 Robustness and Sensitivity metric

From the two metrics considered to evaluate the change of loss, the Loss-SENS is intended to see how much the model's for the input varies with perturbation and the LossGrad-SENS indicates the change of gradients of the loss with respect to the input. If the change in loss is small, it's assumed that optimizing an adversarial example to harm the model's performance is difficult. Similarly, small changes in loss gradients suggest a similar challenge. However, since large loss gradients might still occur even with small loss values, both metrics are compared to ensure consistent results. These metrics were applied to two Vision Transformer models: the Unfrozen First_Last Transformer ViT-B/16 with a batch size of 16, a learning rate of 10^{-6}, and 200 epochs, and the ViT-B/32 with a batch size of 16, a learning rate of 10^{-4}, and 60 epochs. The evaluation aimed to determine which model demonstrated better robustness, accuracy, and generalization in practice.

Furthermore, to calculate the Loss-SENS and LossGrad-SENS metrics in the CNN model for comparison, the process outlined by (He et al. (2024)) was replicated, applying these metrics to the training results from three random seeds using the EfficientNet6 model. By averaging the results from these three runs, a reliable estimate of the sensitivity and gradient-based loss metrics for the EfficientNet6 model, as described in (He et al. (2024)), was obtained. The plots of the loss over the epochs are shown in Appendix A.

Table 4.10: Loss-SENS and LossGrad-SENS values for the two optimal ViT models: Unfrozen First_Last Transformer ViT-B/16 with a batch size of 16, a learning rate of 10^{-6}, and 200 epochs, and the ViT-B/32 with a batch size of 16, a learning rate of 10^{-4}, and 60 epochs and the optimal model used in (He et al. (2024)

Model	**Training Loss-SENS**	**Validation Loss-SENS**	**Training LossGrad-SENS**	**Validation LossGrad-SENS**
ViT-B/32	0.0076	0.0085	0.0045	0.0061
Unfrozen First_Last ViT-B/16	0.0033	0.0017	0.0030	0.0013
EfficientNet6	0.119	0.261	0.0535	0.0815

)

Chapter 5

Discussion

The main goal of this thesis was to develop and test ViT models that automatically classify X-ray radiographs of pears. Four approaches were implemented and compared. This section discusses the outcomes of this research.

5.1 Fine-tuning ViT-B architecture

The ViT architecture processes images by breaking them into patches, allowing the model to capture both local and global features. The difference in performance between the ViT-B/16 and ViT-B/32 models primarily stems from their patch sizes and architectural configurations. Studies shown that, ViT-B/16, with smaller patches, excels at capturing fine-grained details, making it suitable for datasets where such details are crucial. This comes at the cost of higher computational intensity and potentially slower convergence. Conversely, ViT-B/32, with larger patches, offers a more global perspective of the image, leading to slightly higher accuracy and faster convergence due to its broader focus (Remondino et al. (2024)).

5.1.1 Analysis and Insights on Obtained Results

Based on the results from section 4.1, the model Unfrozen First Last Transformer ViT-B/16, with batch size 32, learning rate initialized as 10^{-6}, and epochs 200, proved to be the most effective, achieving a good balance between balanced accuracy (72.8%), precision (91.4%) and with good validation and loss curves. However, this performance came with the need for extended training (200 epochs) and careful learning rate adjustments (10^{-6}). Studies have shown that, selective fine-tuning strategies, such as those implemented in the Unfrozen architecture, often require longer training and lower learning rates (*ULMFiT Explained* (n.d.)). This is because freezing most layers while updating only specific ones forces the model to strike a careful balance between utilizing pre-trained knowledge and adapting to the new task without overfitting (Jiang et al. (2022)). A similar approach is seen in the ROSE (Robust Selective Fine-tuning) method, which follows this principle to enhance model performance (Jiang et al. (2022)).

After a deeper examination of the performance metrics across all models, the high precision values indicate strong effectiveness in accurately identifying infected cases. However, the lower recall indicates a struggle in differentiating between control and infected cases, leading to missed true positives. In imbalanced datasets, improving recall often leads to a drop in precision, as models tend to be biased toward the majority class. This bias can cause models to achieve high accuracy while still failing to correctly identify minority class instances, highlighting the importance of using balanced evaluation metrics to assess performance effectively (Werner de Vargas et al. (2023)). Additionally, the tested models exhibited a clear discrepancy between accuracy and balanced accuracy. For example, the model ViT-B/16 (BS:16 LR:0.0001 epochs 60), the accuracy was 60.8% while the balanced accuracy dropped to 42.0%. This disparity is explained by the low true negative rate for the minority class (3.29%) and a high false positive rate (96.71%) reinforcing the model's bias toward the majority class.

5.1.2 Model performance discussion

Models Depth

In the original Vision Transformer proposal by Dosovitskiy et al. (Dosovitskiy et al. (2020)), a key finding was the relationship between attention distance and network depth. As the depth of the network increases, the attention mechanism progressively focuses on wider areas of the image. This concept, termed mean attention distance, highlights that deeper layers are responsible for integrating information over larger portions of the image, transitioning from local detail extraction in early layers to global context comprehension in later layers.

Building on this, recent approaches in transfer learning, such as adapter modules and Deep Adaptation Modules (DAM) (Rosenfeld & Tsotsos (2020)), emphasize selective layer adaptation to improve parameter efficiency and performance. In this thesis, the Unfrozen First and Last Transformer ViT-B/16 model takes a similar approach by optimizing only the first and last layers. This selective fine-tuning strikes a balance between capturing fine-grained local features in the early layers and refining these with global context in the deeper layers. The smaller patch size (16x16 pixels) further enhances the initial layers' ability to capture detailed information, while the unfrozen deeper layers integrate this with broader context, aligning with both Dosovitskiy's findings and the parameter-efficient strategies seen in adapter modules and DAM. This combination likely contributed to the model's strong performance in this study (Dosovitskiy et al. (2020), Bentoumi et al. (2022)).

Patch Size

Analyzing the performance variations influenced by patch size across models with different layer depths reveals a notable pattern. Specifically, models utilizing larger patch sizes (32×32) demonstrate superior performance compared to those with smaller patch sizes (16×16) as the model depth increases. This finding contrasts with existing literature, which often highlights the benefits of smaller patches for capturing finer details (Dosovitskiy et al. (2020)). The enhanced performance

of larger patches in this context may be attributed to their ability to preserve the overall object structure more effectively, as noted by Bhojanapalli et al. (2021) (Bhojanapalli et al. (2021)), who emphasize that larger patches retain higher-level semantic information, thus benefiting certain vision tasks. Nevertheless, ViT-B/16 remains superior for tasks demanding fine-grained details, particularly in shallower configurations where fewer layers are active (Sayak (2021)).

In another study conducted to evaluate the components of ViT's in terms of their impact on model robustness, ViT-B/16 and ViT-B/32 were compared and confirmed that patch size in ViT models plays a role in their robustness and smaller patch sizes make ViT models more robust to adversarial spatial transformations, but also increase their texture bias, meaning that breaking the image input in finer, more details parts can making the model more reliant on textures rather than understanding the full structure of objects (Bhojanapalli et al. (2021)).

.

5.2 Custom_ViT

The results, presented in Figure 4.6 and Figure 4.7, indicate that the higher balance accuracy obtained in the CustomViT model was 62,6% concluded that the models trained from scratch didn't outperform the optimal pre-trained models.

Beyond the differences in initialized weights between the pre-trained models and the CustomViT model, another key distinction lies in the first layer architecture of each model. The Custom ViT uses a patch extraction layer that divides the image into fixed-size patches, which can result in a loss of local spatial detail. In contrast, pre-trained ViT models often use a 2D convolutional layer, which introduces local inductive bias, enabling them to capture spatial hierarchies more efficiently, improving performance on visual tasks (d'Ascoli et al. (2021)).

Additionally, the top two Custom ViT models were selected based on their training and validation accuracy from the Optuna search. In a comparative study of hyperparameter optimization libraries (HyperOpt, Optuna, Optunity, and SMAC), Optuna was identified as the best choice for the CASH problem benchmark, offering an optimal balance between runtime and performance (Shekhar et al. (2021*a*)). Traditional methods like grid search and random search, while commonly used, can be inefficient and computationally expensive, particularly in large-scale industrial applications (Zöller & Huber (2021)).

However, while Optuna excels in optimizing simpler search spaces, it has some limitations when it comes to handling complex conditional hyperparameter dependencies, as seen in the CustomViT model where parameters like the number of heads and dimensions are interdependent (Shekhar et al. (2021*b*)). HyperOpt, another optimization framework, may offer an advantage in such cases by providing greater flexibility in managing these dependencies (Komer et al. (2019)). Optuna's use of Tree-structured Parzen Estimators (TPE) focuses on areas of the search space that have shown promise, but this can sometimes lead to the neglect of unexplored areas that could yield better results in more intricate models. Furthermore, over-reliance on validation accuracy to

select the best models, as demonstrated in this study, can risk overfitting, meaning that the models might not generalize as well to test data as initially expected. This highlights the need for a more robust selection process that accounts for both validation and test set performance to ensure broader generalization (AI (2023)).

5.3 Prediction along Infected Groups

The results presented in Figures 4.8 and Table 4.9 indicate a clear difference in how the models perform across different infection stages (L1, L3, and L5). The intermediate infection stage (L3) consistently shows the highest accuracy, with higher true negative rates and fewer false positives. This likely occurs because the features of L3 are more distinct and balanced—neither too subtle nor too extreme—making it easier for the models to differentiate between infected and non-infected instances. In this stage, the infection is developed enough for the models to recognize clear patterns, but not so advanced that the features become overly complex or ambiguous.

In contrast, the early infection stage (L1) presents a challenge for the models, particularly for pre-trained models, as indicated by the high false positive rates. This is likely due to the subtle nature of the features in L1, where the infection has barely begun to manifest. The models struggle to distinguish between non-infected and early-stage infection because the visual or feature differences are minimal, causing them to misclassify non-infected instances as infected. Pre-trained models may especially struggle here because they may not be fine-tuned enough to capture such subtle distinctions.

On the other hand, in the most advanced infection stage (L5), both pre-trained and custom models struggle to identify true negatives, as evidenced by the low true negative rates. This suggests that the features of L5, while more pronounced, might overlap or resemble those of non-infected cases in ways that confuse the models. The advanced infection features may introduce complexities that make it harder to distinguish between severely infected and non-infected cases. However, custom models show some improvement over pre-trained models in reducing false positives for L5, possibly due to their ability to learn more specific patterns through training tailored to the dataset.

5.4 ViT vs CNN model

Despite the promising performance metrics of the ViT model in other works for image classification tasks, no guarantees were given that the model applied to a smaller dataset will work. The results obtained from the best fine-tuned ViT model show the capability to detect pests in fruits with a good level of accuracy and precision. However, the recall metric suggests that there is still a significant number of pest-affected fruits that are not being correctly identified (false negatives). The fluctuating validation loss, shown in some models, further highlights the challenge of achieving consistent performance across both training and unseen datasets.

Loss values reached

During training, the Unfrozen First_Last Transformer ViT-B/16 model achieved the lowest training loss across all models tested with a value of 0.31. However, while this is the lowest among the models, is still relatively high compared to what is typically expected in other types of deep learning architectures, such as Convolutional Neural Networks (CNNs), which often achieve lower training losses (He et al. (2024), Touvron et al. (2022)).

This plateau can be attributed to the inherent properties of ViTs, especially the nature of multi-head self-attentions (MSAs). Research findings that MSAs flatten the loss landscape by reducing the magnitude of Hessian eigenvalues, which helps the model performance and generalization (Li et al. (2018), Park & Kim (2022)). On the other hand, MSA mechanism allows more negative Hessian eigenvalues in small data regimes compared to CNN models (Park & Kim (2022)). Since the Hessian represents local curvature, this means that the loss landscapes of MSAs are non-convex, and this non-convexity disturbs Neural Network (NN) optimization, because creating local maximos defaulting the optimization during training, especially in the start stages (Dauphin et al. (2014)). Furthermore, MSA mechanism in ViTs, processes the entire image at once and captures global relationships between all parts of the image, learning global relationships without prioritizing local spatial information (such as nearby pixels being related). This flexibility gives MSA less inductive bias what is proven to disrupt NN training (Park & Kim (2022), Zhong et al. (2020)). For example, local MSAs, which calculate self-attention only within small windows, achieve better performance than global MSAs not only on small datasets but also on large datasets (Liu et al. (2021), Park & Kim (2022)).

The Loss-SENS and LossGrad-SENS values provide additional insight into why the model's training loss did not reduce further. The low Loss-SENS (0.0033) and the low LossGrand-SENS (0.0030) suggest that the model's loss is relatively stable and does not fluctuate much with small perturbations, indicating that further training would not result in significant changes in the model's performance. This stability can be interpreted as the model has reached a point of diminishing returns, where continuing to train for more epochs does not lead to meaningful improvements in the loss (Park & Kim (2022)).

Robustness evaluation metrics

In an article titled *Exploring the differences in adversarial robustness between ViT and CNN-based models using novel metrics* (Heo et al. (2023)), Loss-SENS and LossGrad-SENS were employed to evaluate the adversarial robustness of convolutional neural networks and vision transformers in the context of random noise (epsilon values) and gradient-based adversarial perturbations (FGSM and PGD techniques). The study highlights that ViT models generally exhibit higher adversarial robustness than CNNs across a variety of datasets, though it also conclude that all models demonstrated improved robustness when pre-trained on ImageNet-21K and ImageNet-1K. The study goes furthermore, comparing ViT-B/16 and ViT-B/32 to assess the impact of the patch size on ViT sensitivity and reveals that smaller patches exhibit greater sensitivity for ViT models.

When comparing the Unfrozen First_Last Transformer ViT-B/16 model to the EfficientNet6 model from (He et al. (2024)), the Loss-SENS and LossGrad-SENS metrics reveal that the ViT model shows significantly higher robustness and stability to input perturbations. The EfficientNet6 model exhibited much higher sensitivity to changes in input data, as evidenced by its substantially higher Loss-SENS and LossGrad-SENS values during both training and validation. This higher sensitivity suggests that EfficientNet6 may be more prone to performance fluctuations in real-world scenarios where input data could be noisy or imperfect.

In contrast, the ViT-B/16 model's lower sensitivity metrics demonstrate that it maintains performance stability even with small changes in input data, making it a more reliable choice for practical applications involving noise or image distortions. These results align with findings from the literature (Heo et al. (2023)), where ViT models were shown to have higher adversarial robustness compared to CNNs.

5.5 Challenges in ViT Performance and the Path Forward

The results obtained in this thesis are not satisfactory, and even the best model is not yet ready for real-world application. The Unfrozen First Last Transformer ViT-B/16 achieved an accuracy of 72.8%, which is far from ideal. Furthermore, the model incorrectly predicted 37.61% of healthy samples, classifying them as infected. In the context of fruit infection by a pest like the codling moth, 38% of infected pears being incorrectly classified as healthy represents a substantial risk, as it means a large number of infected fruit could go undetected. This is especially critical given that the codling moth pest can easily spread from infected to healthy pears, particularly during the larval stage, the most destructive phase. During this phase, the larvae cause severe internal damage, making early detection essential to prevent widespread infestation. Additionally, 16.87% of predictions that labeled pears as infected were actually healthy, meaning that 17% of perfect fruit would be wasted, contributing to unnecessary food waste.

Performance of Vision Transformer Below Expectations

Despite the growing body of research demonstrating ViT models outperforming CNN-based models in traditional computer vision tasks, the results of this thesis suggest otherwise. In comparison to a similar study conducted with the same dataset and goal (He et al. (2024))— detecting codling moth in pears— the ViT model underperforms. The previous study reported an accuracy of 87.9%, precision of 94.2%, and an F1-score of 91.9% obtained with EfficientNetb6 architecture, significantly surpassing the best results from this thesis of 72.8%, 91.4%, and 74.2%, respectively.

These discrepancies raise important questions about why the ViT model in this thesis did not demonstrate equal or better performance, even under similar conditions. To fully understand the differences in results, it's important to examine key factors that could have influenced the performance disparity between the Vision Transformer (ViT) and Convolutional Neural Network (CNN) models and also between this thesis result and the expected ViT results from the literature. Both models used the same dataset for detecting codling moths in pears, but several differences in the

training processes, data handling, and architectural complexity played a significant role in the outcomes. First, data augmentation plays a significant role in the training process, especially for vision models (Taylor & Nitschke (2018)). The previous study used rotations (45º, 90º, 180º or 270º) as part of their data augmentation strategy, which likely helped the CNN model generalize better (Taylor & Nitschke (2018)). However, in this thesis, applying rotations during training caused instability in the validation process, leading to oscillations in validation accuracy, which ultimately prevented the model from stabilizing and improving. This could have limited the ViT's ability to learn effectively from the dataset. The ViT architecture may have been more sensitive to certain augmentations due to its reliance on global relationships in the image, making it more prone to validation issues when transformations like rotations are applied. Second, the train-validation-test split differed between the two studies. The previous study employed a 60-20-20 split, dedicating more data to validation and testing, while in thesis used an 80-10-10 split, allocating a larger proportion of the data to training. Although a larger training set can be advantageous, the smaller validation set may have led to less reliable estimates of model performance during training. This could have contributed to overfitting, where the model performs well on the training set but struggles to generalize on the test set. In contrast, the CNN model in the previous study may have benefited from having more balanced validation and test sets, leading to more robust performance.

Another critical factor is the difficulty of training Vision Transformers compared to CNNs. ViTs are known to require more data and a carefully tuned training process due to their lack of inductive bias (the ability to naturally recognize local features, such as edges and textures, which CNNs excel at). Peardataset may not have been large enough or rich enough in variability to fully exploit the strengths of the ViT architecture. As a result, the ViT model struggled to reach the same level of loss minimization during training as the CNN did. The training graphs showed that ViTs face more complexity and challenges when learning from the dataset, which may explain the suboptimal results. In literature, some studies show that ViT models outperform CNN based models (Tempelaere et al. (2023), Dosovitskiy et al. (2020), Park & Kim (2022), Nijhawan et al. (2022)), however, the used dataset was significantly different than the peardataset, in sized or complexity of the images (Regmi et al. (2023)).

Nevertheless, a 2023 study challenges this trend (Regmi et al. (2023)), here, the ViT models that pretrained on ImageNet still outperform CNNs models on two small datasets. This study applied various transformations to three different publicly available medical datasets of varying sizes: 7,135 X-ray images, 8,000 images, and 44,228 images, respectively. The study evaluated multiple CNN architectures alongside ViT models. Interestingly, for the two smaller datasets, the highest performance across all metrics was achieved with a ViT-L/16 model pre-trained on ImageNet-21k. For the third dataset, since it held unbalanced data for each class, was applied Focal Loss (FL) instead of the categorical cross-entropy loss function used in the other models. FL is an advanced form of cross-entropy, that tackles class imbalance by assigning higher weights to hard-to-classify examples and down-weighting easy ones. This encourages the model to focus on challenging cases, improving its ability to generalize across both majority and minority classes (Lin et al. (2017)). That application shows results, with the ViT-B/16 pre-trained on ImageNet-21k

outperformed CNNs in terms of weighted precision, F1-score, and MCC metrics. .

These findings suggest that both pre-training on larger datasets like ImageNet-21k and employing more sophisticated loss functions, such as Focal Loss, could be promising solutions to improve the performance of ViT models, addressing the weaker outcomes observed in this study.

From scratch may have a better future than Fine-Tuning

A recent article suggests that the advantage of fine-tuning a pre-trained model over random initialization diminishes when there is a large gap between the pre-training and target tasks (Liu et al. (2022)). In this thesis, the target task involves classifying X-ray images with grayscale textures, focusing on subtle changes in texture, internal structure, and patterns—very different from the pre-training task on ImageNet, which contains 1,000 classes of everyday objects in RGB format. As a result, fine-tuning may not offer significant benefits, and starting with a randomly initialized model could produce similar or even better results (Zoph et al. (2020)). Additionally, the study (Liu et al. (2022)) highlights that models trained from scratch can outperform pre-trained models when the dataset is large enough, though this is not the case for the pear dataset. This may explain why the CustomViT model did not show improvement over pre-trained models.

Contrary to the challenges faced by the CustomViT model, Seunghoon Lee (2022) (Lee et al. (2022)) improved the accuracy of ViTs on small datasets like Tiny-ImageNet by up to 5.7% with specific enhancements such as Shifted Patch Tokenization (SPT) and Locality Self-Attention (LSA). SPT improves the preservation of local spatial information by modifying how images are divided into patches, while LSA enhances the self-attention mechanism by focusing on local context, compensating for the lack of inductive bias that typically benefits CNNs. These improvements address the main limitations of standard ViTs on smaller datasets, suggesting that incorporating similar enhancements in the CustomViT model could potentially improve its performance on tasks like X-ray classification, where local feature understanding is crucial.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In conclusion, this thesis explored the performance of Vision Transformer (ViT) models, specifically ViT-B/16 and ViT-B/32, in classifying pear X-ray images for detecting codling moth infections. Despite the recent advances in ViTs outperforming CNN-based models in many image classification tasks, the results did not show a clear advantage over CNNs when trained on smaller, specialized datasets like the pear dataset. Several factors influenced this outcome, including differences in augmentation techniques, training splits, and the intrinsic complexity of ViT architectures, which require larger datasets and more tuning to optimize. Additionally, the instability introduced by certain augmentations like rotations, and the challenge of balancing the number of heads and dimensionality, may have contributed to sub-optimal performance.

This research highlights key challenges in pest detection, such as complex backgrounds, lighting variations, and the difficulty in collecting large, diverse pest image datasets. This study addresses these issues by evaluating how ViTs perform on small datasets compared to EfficientNet6, a well-established CNN architecture. The thesis emphasizes the need for fine-tuning ViT models, particularly for small, specialized datasets, and lays the groundwork for future advancements in crop pest detection, improving model robustness and accuracy in challenging conditions.

6.2 Future Work

Looking ahead, this work represents a significant first step in applying Vision Transformers (ViT) architectures for detecting codling moth larvae in pear X-ray radiographs. However, several areas remain open for improvement in future work. First, the Optuna hyperparameter optimization can be further explored, as only a limited range of CustomViT architectures was tested, potentially leaving out more optimal configurations. Testing the models on real radiographs and larger, independent datasets will help validate the approach. Moreover, implementing CT data directly, experimenting with hybrid models (combining CNN and ViT layers), and applying techniques discussed in the limitations could provide further insights and enhancements to the model's robustness and generalizability.

Optuna hyperparameter optimization framework

While this thesis explored a limited range of architectures, key parameters such as batch size, number of layers, and augmentation techniques could be more thoroughly investigated. Expanding the search space for these factors may uncover better-performing configurations. Additionally, testing different regularization methods could enhance the model's generalization. These improvements, coupled with more extensive hyperparameter tuning, could lead to more optimized results for the ViT models on pest detection tasks.

Real Scenario Data

To improve the robustness of the proposed model, it should be tested on real radiographs from an independent dataset of pears. Currently, the models are trained on artificially infested samples, which do not fully represent the natural infestation process. This gap between simulated and real conditions must be addressed. Future work should involve collecting pears directly from orchards, suspected of having larval infestations, to create a more realistic independent test set. This approach would help evaluate the model's effectiveness in real-world scenarios and ensure better generalization.

Larger Dataset

In the future, expanding the dataset will be crucial for improving the performance of the ViT architecture. Currently, the dataset is limited to 27 radiographs per pear, which may not be enough to fully leverage the model's capabilities. By generating more radiographs per CT scan, the dataset could be enriched with additional information, offering more variability and better insights. Alternatively, incorporating radiographs from different pear species would not only increase the dataset size but also introduce greater diversity, enhancing the model's ability to generalize across various conditions.

ViT to Learn Small-Size Dataset From Scratch

In future work, testing the implementation of a technique proposed by Lee (2022) in the paper *Improving Vision Transformers to Learn Small-Size Dataset From Scratch*, discussed in the last section, could provide valuable insights. Applying the article modifications to the CustomViT model could help overcome the limitations posed by the smaller dataset used in this study and potentially lead to performance improvements.

Using CT scans

In the literature review of X-ray techniques it became clear what advantages CT scans have over radiographs. So a future direction for this research would be to develop models trained directly on 3D CT data instead of traditional X-ray radiographs. Since CT scans inherently contain detailed spatial information, the model's predictions would be less impacted by variations in pear orientation, angle, or shape. However, finding the right balance between throughput (speed and efficiency of CT scanning) and reconstruction quality (the level of detail in the images) will remain a challenge. To address this, CNNs could be adapted for 3D data, and various CT slice counts could be tested to explore the relationship between reconstruction quality and model performance.

Hybrid model

In future work, implementing ViT layers within CNN models to create a hybrid architecture could offer promising improvements. This approach, discussed by Namuk Park (2022) in the paper *How do Vision Transformers work?*, has strong theoretical potential. By combining the local feature extraction capabilities of CNNs with the global context awareness of ViTs, a hybrid model could leverage the strengths of both architectures. This hybrid approach could also reduce the need for a large dataset, and make it more suited for smaller datasets, where both detailed and broader contexts are essential for accurate pest detection in pears, without relying heavily on extensive data collection.

References

Adedeji, A. A., Ekramirad, N., Rady, A., Hamidisepehr, A., Donohue, K. D., Villanueva, R. T., Parrish, C. A. & Li, M. (2020), 'Non-destructive technologies for detecting insect infestation in fruits and vegetables under postharvest conditions: A critical review', *Foods* **9**(7), 927.

AI, N. (2023), 'Optuna vs hyperopt: Which hyperparameter optimization library should you choose?'. Accessed: 2024-09-09.
URL: *https://neptune.ai/blog/optuna-vs-hyperopt*

Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. (2019), Optuna: A next-generation hyperparameter optimization framework, *in* 'Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining', pp. 2623–2631.

Bahdanau, D., Cho, K. & Bengio, Y. (2014), 'Neural machine translation by jointly learning to align and translate', *arXiv preprint arXiv:1409.0473* .

Barman, U., Sarma, P., Rahman, M., Deka, V., Lahkar, S., Sharma, V. & Saikia, M. J. (2024), 'Vit-smartagri: vision transformer and smartphone-based plant disease detection for smart agriculture', *Agronomy* **14**(2), 327.

Bello, I., Zoph, B., Vaswani, A., Shlens, J. & Le, Q. V. (2019), Attention augmented convolutional networks, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 3286–3295.

Bentoumi, M., Daoud, M., Benaouali, M. & Taleb Ahmed, A. (2022), 'Improvement of emotion recognition from facial images using deep learning and early stopping cross validation', *Multimedia Tools and applications* **81**(21), 29887–29917.

Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T. & Veit, A. (2021), Understanding robustness of transformers for image classification, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 10231–10241.

Blasco, J., Aleixos, N. & Molto, E. (2007), 'Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm', *Journal of Food engineering* **81**(3), 535–543.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020), 'Language models are few-shot learners', *Advances in neural information processing systems* **33**, 1877–1901.

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C. & Collins, J. J. (2018), 'Next-generation machine learning for biological networks', *Cell* **173**(7), 1581–1592.

Castiglioni, I., Rundo, L., Codari, M., Di Leo, G., Salvatore, C., Interlenghi, M., Gallivanone, F., Cozzi, A., D'Amico, N. C. & Sardanelli, F. (2021), 'Ai applications to medical images: From machine learning to deep learning', *Physica medica* **83**, 9–24.

Chai, J. & Li, A. (2019), Deep learning in natural language processing: A state-of-the-art survey, *in* '2019 International Conference on Machine Learning and Cybernetics (ICMLC)', IEEE, pp. 1–6.

Chai, J., Zeng, H., Li, A. & Ngai, E. W. (2021), 'Deep learning in computer vision: A critical review of emerging techniques and application scenarios', *Machine Learning with Applications* **6**, 100134.

Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S. & Miao, Y. (2021), 'Review of image classification algorithms based on convolutional neural networks', *Remote Sensing* **13**(22), 4712.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D. & Sutskever, I. (2020), Generative pretraining from pixels, *in* 'International conference on machine learning', PMLR, pp. 1691–1703.

Chollet, F. (2017), Xception: Deep learning with depthwise separable convolutions, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1251–1258.

Coccomini, D. A., Messina, N., Gennaro, C. & Falchi, F. (2022), Combining efficientnet and vision transformers for video deepfake detection, *in* 'International conference on image analysis and processing', Springer, pp. 219–229.

Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J. & Blasco, J. (2011), 'Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables', *Food and bioprocess technology* **4**, 487–504.

Curry, T. S., Dowdey, J. E. & Murry, R. C. (1990), *Christensen's physics of diagnostic radiology*, Lippincott Williams & Wilkins.

d'Ascoli, S., Touvron, H., Leavitt, M., Morcos, A. S., Biroli, G. & Sagun, L. (2021), 'Convit: Improving vision transformers with soft convolutional inductive biases', *Proceedings of the International Conference on Machine Learning (ICML)* .
URL: *https://proceedings.mlr.press/v139/d-ascoli21a.html*

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S. & Bengio, Y. (2014), 'Identifying and attacking the saddle point problem in high-dimensional non-convex optimization', *Advances in neural information processing systems* **27**.

De Ollas, C., Morillón, R., Fotopoulos, V., Puértolas, J., Ollitrault, P., Gómez-Cadenas, A. & Arbona, V. (2019), 'Facing climate change: biotechnology of iconic mediterranean woody crops', *Frontiers in Plant Science* **10**, 427.

Do Huh, H. & Kim, S. (2020), 'History of radiation therapy technology', *Progress in Medical Physic* **31**(3), 124–134.

Doi, K. (2007), 'Computer-aided diagnosis in medical imaging: historical review, current status and future potential', *Computerized medical imaging and graphics* **31**(4-5), 198–211.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020), 'An image is worth 16x16 words: Transformers for image recognition at scale', *arXiv preprint arXiv:2010.11929* .

Ekramirad, N., Adedeji, A. A. & Alimardani, R. (2016), 'A review of non-destructive methods for detection of insect infestation in fruits and vegetables', *Innovations in Food Research* **2**(1), 6–12.

El-Mesery, H. S., Mao, H. & Abomohra, A. E.-F. (2019), 'Applications of non-destructive technologies for agricultural and food products quality inspection', *Sensors* **19**(4), 846.

Elyan, E., Vuttipittayamongkol, P., Johnston, P., Martin, K., McPherson, K., Jayne, C., Sarker, M. K. et al. (2022), 'Computer vision and machine learning for medical image analysis: recent advances, challenges, and way forward.', *Artificial Intelligence Surgery* **2**.

Fan, J., Jehle, J. A., Rucker, A. & Nielsen, A. L. (2022), 'First evidence of cpgv resistance of codling moth in the usa', *Insects* **13**(6), 533.

Finlay, C. & Oberman, A. M. (2021), 'Scaleable input gradient regularization for adversarial robustness', *Machine Learning with Applications* **3**, 100017.

Fu, X., Ma, Q., Yang, F., Zhang, C., Zhao, X., Chang, F. & Han, L. (2024), 'Crop pest image recognition based on the improved vit method', *Information Processing in Agriculture* **11**(2), 249–259.

Galinato, S. P., Granatstein, D. & Taylor, M. R. (2011), '2010 cost estimates of establishing and producing organic apples in washington'.

Gao, X., Xiao, Z. & Deng, Z. (2024), 'High accuracy food image classification via vision transformer with data augmentation and feature augmentation', *Journal of Food Engineering* **365**, 111833.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep learning*, MIT press.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. & Lew, M. S. (2016), 'Deep learning for visual understanding: A review', *Neurocomputing* **187**, 27–48.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y. et al. (2022), 'A survey on vision transformer', *IEEE transactions on pattern analysis and machine intelligence* **45**(1), 87–110.

Hanke, R., Fuchs, T. & Uhlmann, N. (2008), 'X-ray based methods for non-destructive testing and material characterization', *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **591**(1), 14–18.

He, J., Verlinde, S., Belien, T., Alhmedi, A., Verboven, P. & Nicolai, B. (2024), 'Non-destructive larval infestation damage detection in pear fruits using deep learning and x-ray ct-generated radiographs'.

Heo, J., Seo, S. & Kang, P. (2023), 'Exploring the differences in adversarial robustness between vit- and cnn-based models using novel metrics', *Computer Vision and Image Understanding* **235**, 103800.
URL: *https://www.sciencedirect.com/science/article/pii/S1077314223001807*

Hoang, V.-D. & Pham, T.-A. (2023), Fusion of vit technique and image filtering in deep learning for plant pests and diseases recognition, *in* '2023 International Conference on System Science and Engineering (ICSSE)', IEEE, pp. 438–443.

Hounsfield, G. N. (1973), 'Computerized transverse axial scanning (tomography): Part 1. description of system', *The British journal of radiology* **46**(552), 1016–1022.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017), Densely connected convolutional networks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4700–4708.

Husain, M., Rathore, J. P., Sharma, A., Raja, A., Qadri, I. & Wani, A. (2018), 'Description and management strategies of important pests of pear: A review', *J. Entomol. Zool. Stud* **6**, 677–683.

James-Martin, G., Williams, G., Stonehouse, W., O'Callaghan, N. & Noakes, M. (2015), 'Health and nutritional properties of pears (pyrus)'.

Jamil, S., Jalil Piran, M. & Kwon, O.-J. (2023), 'A comprehensive survey of transformers for computer vision', *Drones* **7**(5), 287.

Janiesch, C., Zschech, P. & Heinrich, K. (2021), 'Machine learning and deep learning', *Electronic Markets* **31**(3), 685–695.

Jiang, D., Chen, S., Hao, M., Fu, J. & Ding, F. (2018), 'Mapping the potential global codling moth (cydia pomonella l.) distribution based on a machine learning method', *Scientific reports* **8**(1), 13093.

Jiang, L., Zhou, H., Lin, Y., Li, P., Zhou, J. & Jiang, R. (2022), 'Rose: Robust selective fine-tuning for pre-trained language models', *arXiv preprint arXiv:2210.09658* .

Jin, H. (2022), 'Hyperparameter importance for machine learning algorithms', *arXiv preprint arXiv:2201.05132* .

Joshi, N. K., Hull, L. A. & Krawczyk, G. (2020), 'Insecticide baseline sensitivity in codling moth (lepidoptera: Tortricidae) populations from orchards under different management practices', *Journal of entomological science* **55**(1), 105–116.

Kadoić Balaško, M., Bažok, R., Mikac, K. M., Lemic, D. & Pajač Živković, I. (2020), 'Pest management challenges and control practices in codling moth: A review', *Insects* **11**(1), 38.

Kastner, J. & Heinzl, C. (2015), 'X-ray computed tomography for non-destructive testing and materials characterization', *Integrated Imaging and Vision Techniques for Industrial Inspection: Advances and Applications* pp. 227–250.

Ketcham, R. A. & Carlson, W. D. (2001), 'Acquisition, optimization and interpretation of x-ray computed tomographic imagery: applications to the geosciences', *Computers & Geosciences* **27**(4), 381–400.

Khaled, A. Y., Ekramirad, N., Parrish, C. A., Eberhart, P. S., Doyle, L. E., Donohue, K. D., Villanueva, R. T. & Adedeji, A. A. (2022), 'Non-destructive detection of codling moth infestation in apples using acoustic impulse response signals', *Biosystems Engineering* **224**, 68–79.

Khan, M. A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H. & Saba, T. (2018), 'Ccdf: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep cnn features', *Computers and electronics in agriculture* **155**, 220–236.

Kim, M., Kang, D., Lee, D., Kim, H., Cho, G. & Jae, M. (2014), 'Fast signal transfer in a large-area x-ray cmos image sensor', *Journal of Instrumentation* **9**(08), P08011.

Komer, B., Bergstra, J. & Eliasmith, C. (2019), 'Hyperopt-sklearn', *Automated Machine Learning: Methods, Systems, Challenges* pp. 97–111.

Kotwaliwale, N., Singh, K., Kalne, A., Jha, S. N., Seth, N. & Kar, A. (2014), 'X-ray imaging methods for internal quality evaluation of agricultural produce', *Journal of food science and technology* **51**, 1–15.

Kramer, P. (2012), *Physiology of woody plants*, Elsevier.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), 'Imagenet classification with deep convolutional neural networks', *Advances in neural information processing systems* **25**.

Kugunavar, S. & Prabhakar, C. (2021), 'Convolutional neural networks for the diagnosis and prognosis of the coronavirus disease pandemic', *Visual computing for industry, biomedicine, and art* **4**(1), 12.

Lee, S., Lee, S. & Song, B. C. (2022), 'Improving vision transformers to learn small-size dataset from scratch', *IEEE Access* **10**, 123212–123224.

Li, H., Li, S., Yu, J., Han, Y. & Dong, A. (2022), Plant disease and insect pest identification based on vision transformer, *in* 'International conference on internet of things and machine learning (IoTML 2021)', Vol. 12174, SPIE, pp. 194–201.

Li, H., Xu, Z., Taylor, G., Studer, C. & Goldstein, T. (2018), 'Visualizing the loss landscape of neural nets', *Advances in neural information processing systems* **31**.

Li, L.-H. & Tanone, R. (2022), Vision transformer approach for vegetables recognition, *in* '2022 International Seminar on Application for Technology of Information and Communication (iSemantic)', IEEE, pp. 113–118.

Liang, C.-P., Sack, C., McGrath, S., Cao, Y., Thompson, C. J. & Robin, L. P. (2021), 'Us food and drug administration regulatory pesticide residue monitoring of human foods: 2009-2017', *Food Additives & Contaminants: Part A* **38**(9), 1520–1538.

Lim, K. & Barigou, M. (2004), 'X-ray micro-computed tomography of cellular food products', *Food research international* **37**(10), 1001–1012.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollar, P. (2017), Focal loss for dense object detection, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'.

Liu, H., Lee, S.-H. & Chahl, J. S. (2017), 'A review of recent sensing technologies to detect invertebrates on crops', *Precision Agriculture* **18**, 635–666.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021), Swin transformer: Hierarchical vision transformer using shifted windows, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 10012–10022.

Liu, Z., Xu, Y., Xu, Y., Qian, Q., Li, H., Ji, X., Chan, A. & Jin, R. (2022), Improved fine-tuning by better leveraging pre-training data, *in* S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho & A. Oh, eds, 'Advances in Neural Information Processing Systems', Vol. 35, Curran Associates, Inc., pp. 32568–32581.

URL: *https://proceedings.neurips.cc/paper_files/paper/2022/file/d1c88f9790765146ec8fb5d02e5653a0-Paper-Conference.pdf*

Loshchilov, I. (2017), 'Decoupled weight decay regularization', *arXiv preprint arXiv:1711.05101* .

Lu, Y., Huang, Y. & Lu, R. (2017), 'Innovative hyperspectral imaging-based techniques for quality evaluation of fruits and vegetables: A review', *Applied Sciences* **7**(2), 189.

Luo, R., Song, Y., Zhao, H., Zhang, Y., Zhang, Y., Zhao, N., Huang, L. & Su, R. (2022), 'Densetnt: Efficient vehicle type classification neural network using satellite imagery', *arXiv preprint arXiv:2209.13500* .

Maggi, C. & Chreil, R. (2023), 'Codling moth (cydia pomonella) biology, and integrated pest management', *Tree Fruit Insects,(1)* pp. 1–12.

Mahendiran, G., Lal, S. & Sharma, O. (2022), 'Pests and their management on temperate fruits: (apple, pear, peach, apricot, cherry, persimmon, walnut, olive, kiwifruit and strawberry)', *Trends in Horticultural Entomology* pp. 891–941.

Mankin, R., Osbrink, W., Oi, F. & Anderson, J. (2002), 'Acoustic detection of termite infestations in urban trees', *Journal of Economic Entomology* **95**(5), 981–988.

Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y. & Xue, H. (2022), Towards robust vision transformer, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 12042–12051.

Mathanker, S. K., Weckler, P. R. & Bowser, T. J. (2013), 'X-ray applications in food and agriculture: a review', *Transactions of the ASABE* **56**(3), 1227–1239.

Moscetti, R., Haff, R. P., Saranwong, S., Monarca, D., Cecchini, M. & Massantini, R. (2014), 'Nondestructive detection of insect infested chestnuts based on nir spectroscopy', *Postharvest Biology and Technology* **87**, 88–94.

Naseer, M. M., Ranasinghe, K., Khan, S. H., Hayat, M., Shahbaz Khan, F. & Yang, M.-H. (2021), 'Intriguing properties of vision transformers', *Advances in Neural Information Processing Systems* **34**, 23296–23308.

Neethirajan, S., Jayas, D. S. & White, N. (2007), 'Detection of sprouted wheat kernels using soft x-ray image analysis', *Journal of Food Engineering* **81**(3), 509–513.

Nijhawan, R., Batra, A., Kumar, M., Jain, D. K. et al. (2022), 'Food classification of indian cuisines using handcrafted features and vision transformer network', *Available at SSRN 4014907* .

Olakanmi, S., Karunakaran, C. & Jayas, D. (2023), 'Applications of x-ray micro-computed tomography and small-angle x-ray scattering techniques in food systems: A concise review', *Journal of Food Engineering* **342**, 111355.

Omar, A., Andreo, P. & Poludniowski, G. (2020), 'A model for the energy and angular distribution of x rays emitted from an x-ray tube. part i. bremsstrahlung production', *Medical Physics* **47**(10), 4763–4774.

Organization, W. H. et al. (2020), *Pesticide residues in food 2019-Report 2019: Joint FAO/WHO Meeting on Pesticide Residues 2019*, Food & Agriculture Org.

Otsu, N. et al. (1975), 'A threshold selection method from gray-level histograms', *Automatica* **11**(285-296), 23–27.

Pang, S., Ding, T., Qiao, S., Meng, F., Wang, S., Li, P. & Wang, X. (2019), 'A novel yolov3-arch model for identifying cholelithiasis and classifying gallstones on ct images', *PloS one* **14**(6), e0217647.

Parikh, A. P., Täckström, O., Das, D. & Uszkoreit, J. (2016), 'A decomposable attention model for natural language inference', *arXiv preprint arXiv:1606.01933* .

Park, N. & Kim, S. (2022), 'How do vision transformers work?', *arXiv preprint arXiv:2202.06709* .

Parvaiz, A., Khalid, M. A., Zafar, R., Ameer, H., Ali, M. & Fraz, M. M. (2023), 'Vision transformers in medical computer vision—a contemplative retrospection', *Engineering Applications of Artificial Intelligence* **122**, 106126.

Pathmanaban, P., Gnanavel, B. & Anandan, S. S. (2019), 'Recent application of imaging techniques for fruit quality assessment', *Trends in Food Science & Technology* **94**, 32–42.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2023), *sklearn.metrics.balanced_accuracy_score*. Accessed: 2024-08-26.
URL: *https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced$_a$ccuracy$_s$core.html*

Pinhas, J., Soroker, V., Hetzroni, A., Mizrach, A., Teicher, M. & Goldberger, J. (2008), 'Automatic acoustic detection of the red palm weevil', *computers and electronics in agriculture* **63**(2), 131–139.

Qian, X., Jianrong, C., Can, D., Li, S. & Junwen, B. (2021), 'Detection of peel puffing and granulation in citrus based on soft x-ray imaging technology', *Smart Agriculture* **3**(4), 53.

Rady, A., Ekramirad, N., Adedeji, A., Li, M. & Alimardani, R. (2017), 'Hyperspectral imaging for detection of codling moth infestation in goldrush apples', *Postharvest Biology and Technology* **129**, 37–44.

Ranjbarzadeh, R., Bagherian Kasgari, A., Jafarzadeh Ghoushchi, S., Anari, S., Naseri, M. & Bendechache, M. (2021), 'Brain tumor segmentation based on deep learning and an attention mechanism using mri multi-modalities brain images', *Scientific Reports* **11**(1), 1–17.

Raza, S. M., Raza, A., Babeker, M. I. A., Haq, Z.-U., Islam, M. A. & Li, S. (2024), 'Improving citrus fruit classification accuracy in xray images using features enhanced vision transformer architecture'.

Regmi, S., Subedi, A., Bagci, U. & Jha, D. (2023), 'Vision transformer for efficient chest x-ray and gastrointestinal image classification', *arXiv preprint arXiv:2304.11529* .

Remondino, F., Menna, F., Nex, F., Spera, M. G. & Nocerino, E. (2024), 'Deep learning for 3d reconstruction and object detection in cultural heritage: A review of algorithms and applications', *Remote Sensing* **16**(17).
URL: *https://www.mdpi.com/2072-4292/16/17/3135*

Rosenfeld, A. & Tsotsos, J. K. (2020), 'Incremental learning through deep adaptation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(3), 651–663.

Salama, A.-M., Ezzat, A., El-Ramady, H., Alam-Eldein, S. M., Okba, S. K., Elmenofy, H. M., Hassan, I. F., Illés, A. & Holb, I. J. (2021), 'Temperate fruit trees under climate change: Challenges for dormancy and chilling requirements in warm winter regions', *Horticulturae* **7**(4), 86.

Sanchez, O. R., Repetto, M., Carrega, A. & Bolla, R. (2021), Evaluating ml-based ddos detection with grid search hyperparameter optimization, *in* '2021 IEEE 7th International Conference on Network Softwarization (NetSoft)', IEEE, pp. 402–408.

Saranwong, S., Haff, R. P., Thanapase, W., Janhiran, A., Kasemsumran, S. & Kawano, S. (2011), 'A feasibility study using simplified near infrared imaging to detect fruit fly larvae in intact fruit', *Journal of near infrared spectroscopy* **19**(1), 55–60.

Sarvamangala, D. & Kulkarni, R. V. (2022), 'Convolutional neural networks in medical image understanding: a survey', *Evolutionary intelligence* **15**(1), 1–22.

Sayak, P. (2021), 'Vision transformers are robust learners', *Retrieved from https://arXiv: 2105.07581* .

Schwab, E., Gooßen, A., Deshpande, H. & Saalbach, A. (2020), Localization of critical findings in chest x-ray without local annotations using multi-instance learning, *in* '2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)', IEEE, pp. 1879–1882.

Shamsudin, S. S. (2013), 'The development of neural network based system identification and adaptive flight control for an autonomoushelicopter system'.

Shekhar, S., Bansode, A. & Salim, A. (2021*a*), A comparative study of hyper-parameter optimization tools, *in* '2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)', pp. 1–6.

Shekhar, S., Bansode, A. & Salim, A. (2021*b*), A comparative study of hyper-parameter optimization tools, *in* '2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)', IEEE, pp. 1–6.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

Song, Y., Fan, L., Chen, H., Zhang, M., Ma, Q., Zhang, S. & Wu, J. (2014), 'Identifying genetic diversity and a preliminary core collection of pyrus pyrifolia cultivars by a genome-wide set of ssr markers', *Scientia Horticulturae* **167**, 5–16.

Stoeckli, S., Hirschi, M., Spirig, C., Calanca, P., Rotach, M. W. & Samietz, J. (2012), 'Impact of climate change on voltinism and prospective diapause induction of a global pest insect–cydia pomonella (l.)', *PloS one* **7**(4), e35723.

Sumedrea, M., Marin, F.-C., Calinescu, M., Sumedrea, D. & Iorgu, A. (2015), 'Researches regarding the use of mating disruption pheromones in control of apple codling moth-cydia pomonella l', *Agriculture and Agricultural Science Procedia* **6**, 171–178.

Sutin, A., Yakubovskiy, A., Salloum, H. R., Flynn, T. J., Sedunov, N. & Nadel, H. (2019), 'Towards an automated acoustic detection algorithm for wood-boring beetle larvae (coleoptera: Cerambycidae and buprestidae)', *Journal of economic entomology* **112**(3), 1327–1336.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015), Going deeper with convolutions, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1–9.

Tan, M. & Le, Q. (2019), Efficientnet: Rethinking model scaling for convolutional neural networks, *in* 'International conference on machine learning', PMLR, pp. 6105–6114.

Taylor, L. & Nitschke, G. (2018), Improving deep learning with generic data augmentation, *in* '2018 IEEE symposium series on computational intelligence (SSCI)', IEEE, pp. 1542–1547.

Taylor, P. M. (2023), Principles of x-ray production and radiation protection, *in* 'Imaging and Technology in Urology', Springer, pp. 3–8.

Tempelaere, A., Van Doorselaer, L., He, J., Verboven, P., Tuytelaars, T. & Nicolai, B. (2023), Deep learning for apple fruit quality inspection using x-ray imaging, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision', pp. 552–560.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A. & Jégou, H. (2021), Training data-efficient image transformers & distillation through attention, *in* 'International conference on machine learning', PMLR, pp. 10347–10357.

Touvron, H., Cord, M., El-Nouby, A., Verbeek, J. & Jégou, H. (2022), Three things everyone should know about vision transformers, *in* 'European Conference on Computer Vision', Springer, pp. 497–515.

ULMFiT Explained (n.d.). Accessed: 2024-09-12.

Ünal, H. T. & Başçiftçi, F. (2022), 'Evolutionary design of neural network architectures: a review of three decades of research', *Artificial Intelligence Review* **55**(3), 1723–1802.

USDA, F. (2013), 'Usda foreign agricultural service'.

USDA Foreign Agricultural Service (2024), 'Pears: Production, supply, and distribution'. Accessed: 2024-02-16.
URL: *https://www.fas.usda.gov/data/pears-production-supply-and-distribution*

Van Aarle, W., Palenstijn, W. J., Cant, J., Janssens, E., Bleichrodt, F., Dabravolski, A., De Beenhouwer, J., Batenburg, K. J. & Sijbers, J. (2016), 'Fast and flexible x-ray tomography using the astra toolbox', *Optics express* **24**(22), 25129–25147.

Van Dael, M. (2017), 'Online quality control of fruit and vegetables using x-ray imaging'.

Van De Looverbosch, T., Bhuiyan, M. H. R., Verboven, P., Dierick, M., Van Loo, D., De Beenbouwer, J., Sijbers, J. & Nicolaï, B. (2020), 'Nondestructive internal quality inspection of pear fruit by x-ray ct using machine learning', *Food Control* **113**, 107170.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), 'Attention is all you need', *Advances in neural information processing systems* **30**.

Wang, S., Tang, J. & Cavalieri, R. (2001), 'Modeling fruit internal heating rates for hot air and hot water treatments', *Postharvest Biology and Technology* **22**(3), 257–270.

Wang, Z., Herremans, E., Janssen, S., Cantre, D., Verboven, P. & Nicolaï, B. (2018), 'Visualizing 3d food microstructure using tomographic methods: Advantages and disadvantages', *Annual review of food science and technology* **9**(1), 323–343.

Werner de Vargas, V., Schneider Aranda, J. A., dos Santos Costa, R., da Silva Pereira, P. R. & Victória Barbosa, J. L. (2023), 'Imbalanced data preprocessing techniques for machine learning: a systematic mapping study', *Knowledge and Information Systems* **65**(1), 31–57.

Westigard, P., Allen, R. & Gut, L. (1981), 'Pear psylla: relationship of early-season nymph densities to honeydew-induced fruit damage on two pear cultivars', *Journal of Economic Entomology* **74**(5), 532–534.

Wightman, R. (2024), *Pytorch image models*.
URL: *https://github.com/huggingface/pytorch-image-models*

Wilkinson, C. T. (2012), 'Principles of x-ray production and radiation protection', *Imaging and Technology in Urology: Principles and Clinical Applications* pp. 3–5.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N. & Recht, B. (2017), 'The marginal value of adaptive gradient methods in machine learning', *Advances in neural information processing systems* **30**.

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K. & Vajda, P. (2020), 'Visual transformers: Token-based image representation and processing for computer vision', *arXiv preprint arXiv:2006.03677* .

Wu, J., Wang, Y., Xu, J., Korban, S. S., Fei, Z., Tao, S., Ming, R., Tai, S., Khan, A. M. & Postman, J. D. (2018), 'Diversification and independent domestication of asian and european pears', *Genome Biology* **19**, 1–16.

Yong, H., Huang, J., Hua, X. & Zhang, L. (2020), Gradient centralization: A new optimization technique for deep neural networks, *in* 'Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16', Springer, pp. 635–652.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014), 'How transferable are features in deep neural networks?', *Advances in neural information processing systems* **27**.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J. & Yan, S. (2021), Tokens-to-token vit: Training vision transformers from scratch on imagenet, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 558–567.

Yulianti, I., Addawiyah, A., Setiawan, R. et al. (2018), Optimization of exposure factors for x-ray radiography non-destructive testing of pearl oyster, *in* 'Journal of Physics: Conference Series', Vol. 983, IOP Publishing, p. 012004.

Zhong, Z., Zheng, L., Kang, G., Li, S. & Yang, Y. (2020), Random erasing data augmentation, *in* 'Proceedings of the AAAI conference on artificial intelligence', Vol. 34, pp. 13001–13008.

Zöller, M.-A. & Huber, M. F. (2021), 'Benchmark and survey of automated machine learning frameworks', *Journal of artificial intelligence research* **70**, 409–472.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D. & Le, Q. (2020), 'Rethinking pre-training and self-training', *Advances in neural information processing systems* **33**, 3833–3845.

Appendix A

Appendix Results from robustness evaluation metrics

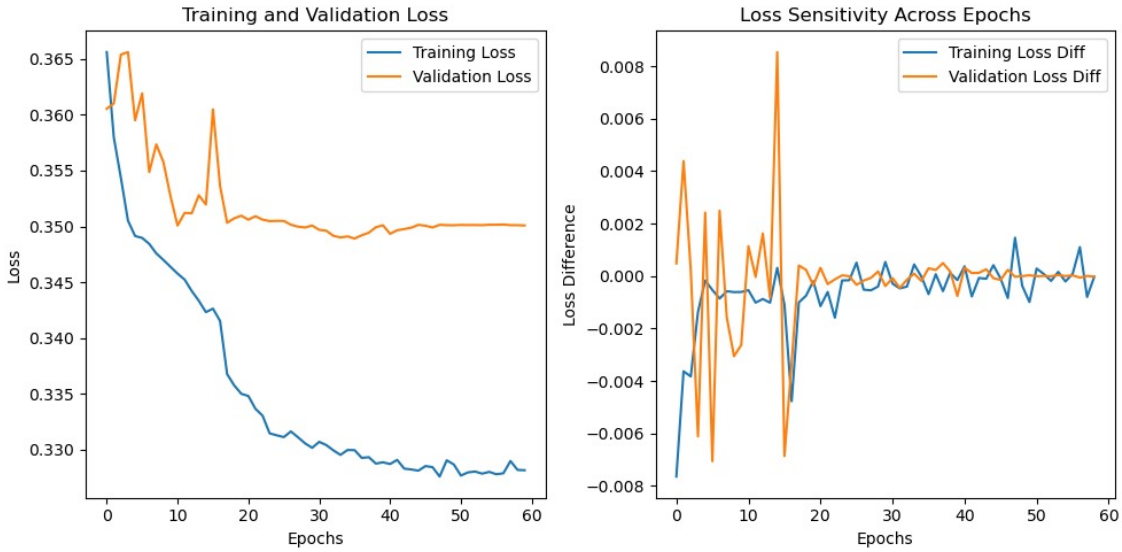

Figure A.1: Representation of training and validation loss curves (left) and the graph of **Loss-SENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **ViT-B/32** optimal model

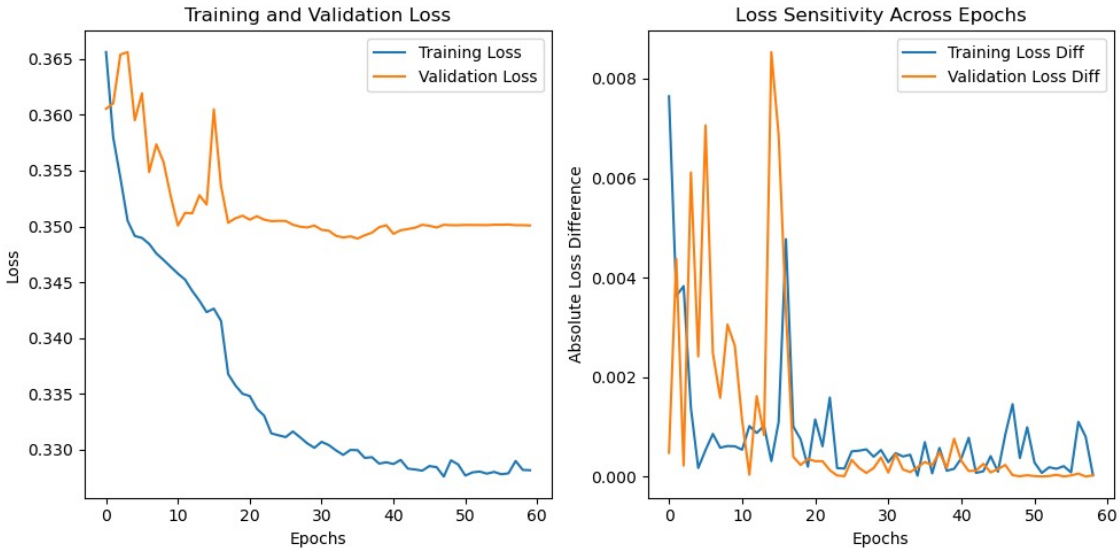

Figure A.2: Representation of training and validation loss curves (left) and the graph of **LossGradSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **ViT-B/32** optimal model

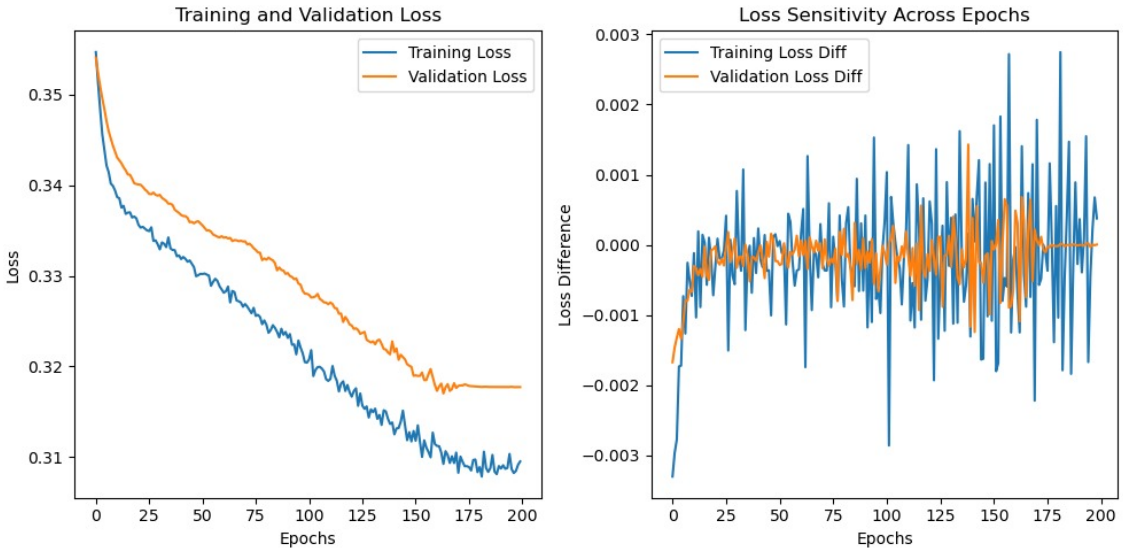

Figure A.3: Representation of training and validation loss curves (left) and the graph of **LossSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **Unfrozen First_Last Transformer ViT-B/16** optimal model

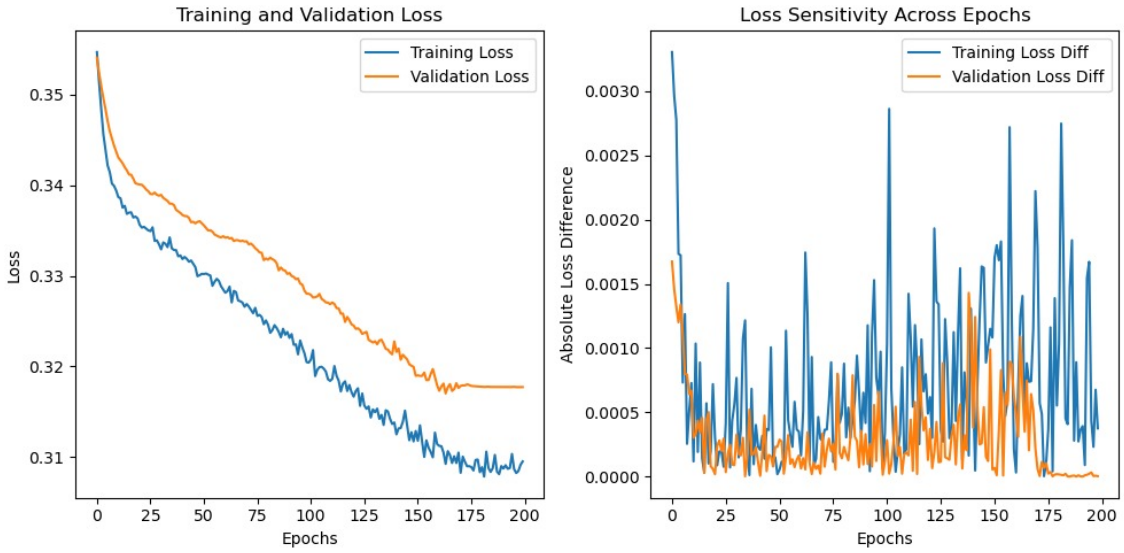

Figure A.4: Representation of training and validation loss curves (left) and the graph of **LossGradSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **Unfrozen First_Last Transformer ViT-B/16** optimal model

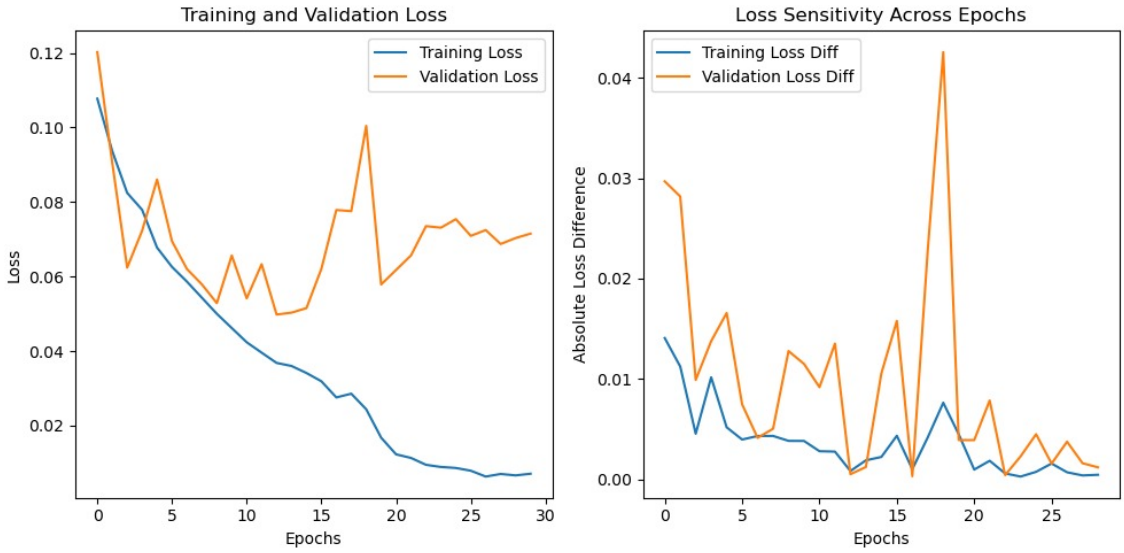

Figure A.5: Representation of training and validation loss curves (left) and the graph of **LossSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **EfficientNet6 random seed 1** based from He et al. (2024)

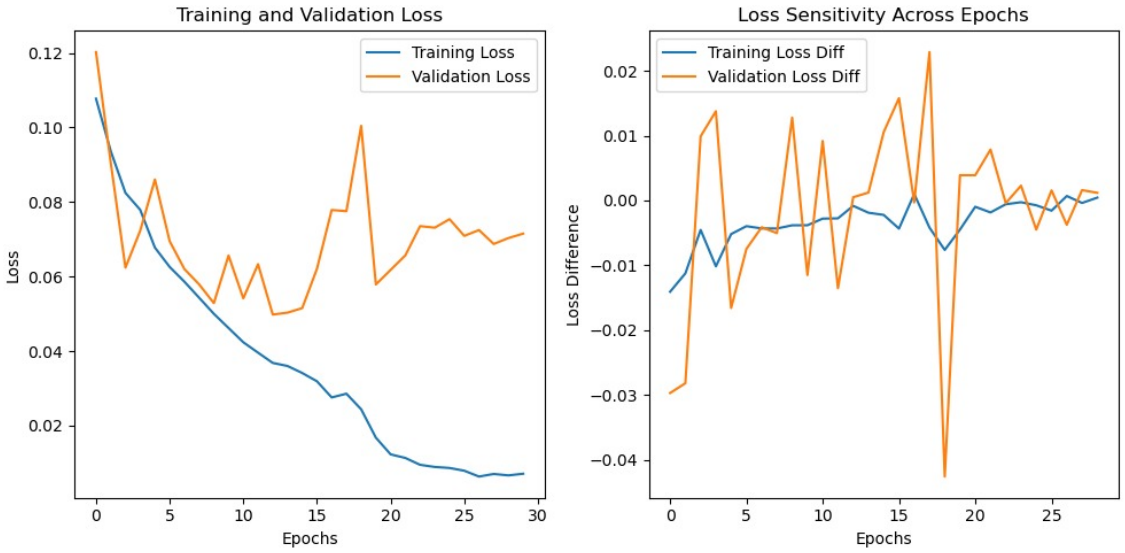

Figure A.6: Representation of training and validation loss curves (left) and the graph of **LossGradSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **EfficientNet6 random seed 1** based from He et al. (2024)

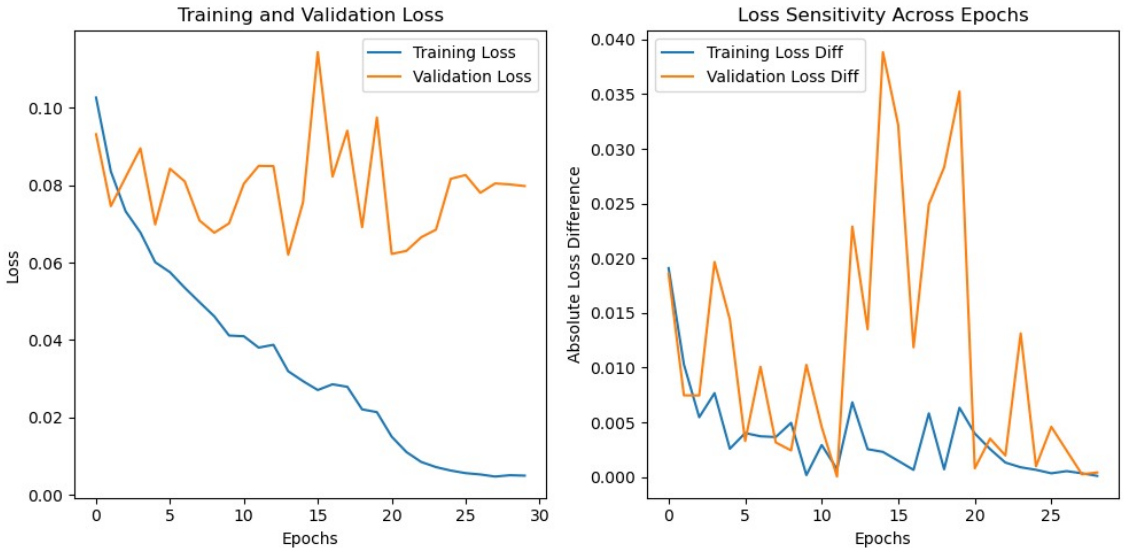

Figure A.7: Representation of training and validation loss curves (left) and the graph of **LossSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **EfficientNet6 random seed 2** based from He et al. (2024)

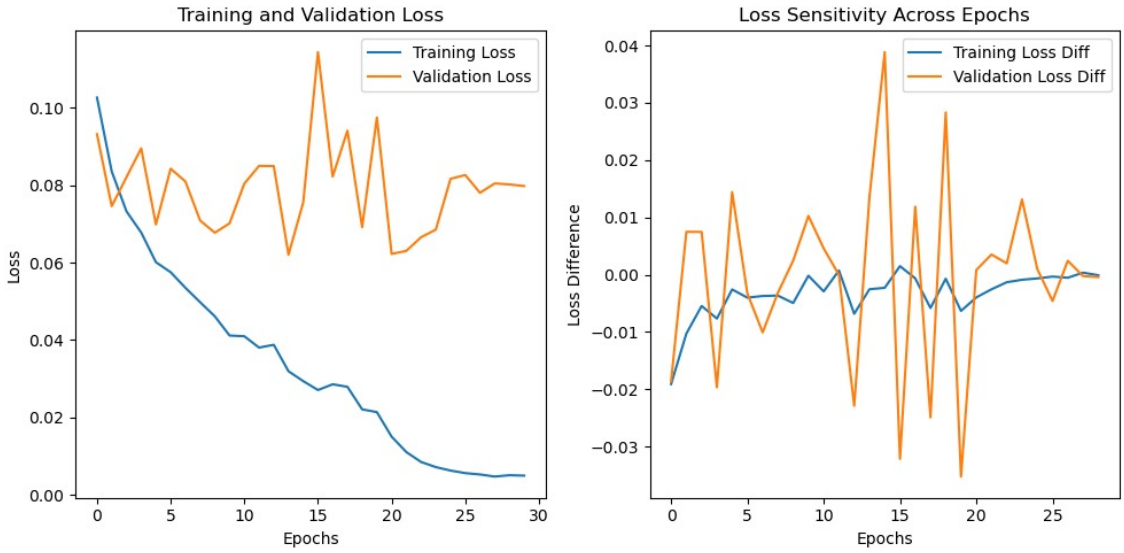

Figure A.8: Representation of training and validation loss curves (left) and the graph of **LossGradSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **EfficientNet6 random seed 2** based from He et al. (2024)

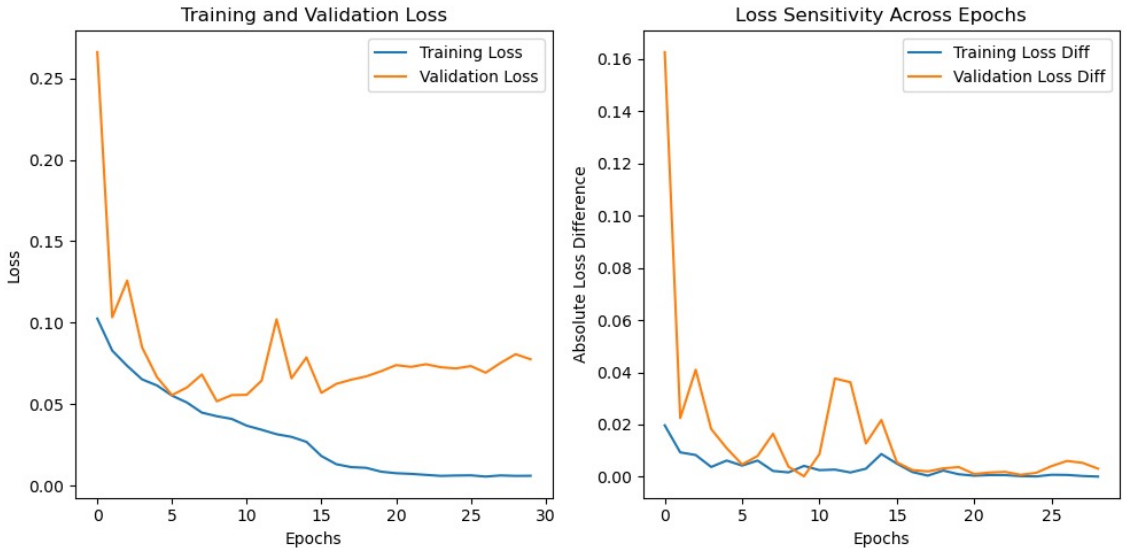

Figure A.9: Representation of training and validation loss curves (left) and the graph of **LossSENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **EfficientNet6 random seed 3** based from He et al. (2024)

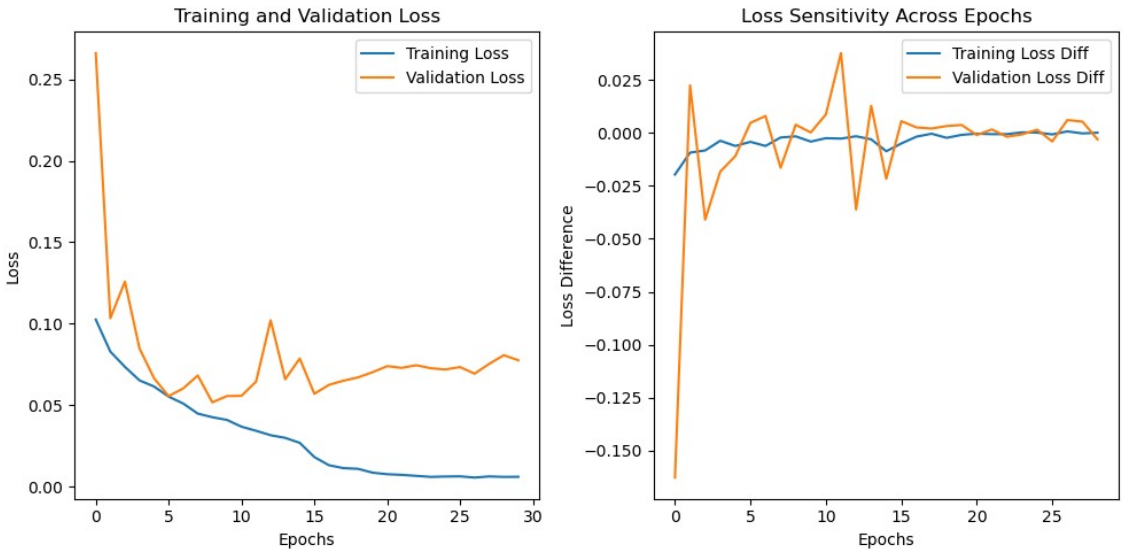

Figure A.10: Representation of training and validation loss curves (left) and the graph of **LossGrad-SENS** (right), measuring how much the model's loss changes when noise is added to the input, for the **EfficientNet6 random seed 3** based from He et al. (2024)