# Capstone Insights

## Sense, Detect, Transform: AI Innovations Across Domains

### Volume 4

**ISBN:** 978-1-47132-557-1

**Publisher:**The Big Publisher

# Capstone Insights: AI & Technology – 2026

## Volume 4

A curated selection of student capstone projects from programmes offered by Euclea Business School in affiliation with Britts Imperial University College

Published by:

**The Big Publisher**

[Sharjah, United Arab Emirates] Year:

2026

**Published by: The Big Publisher, Sharjah Book Authority, Sharjah, United Arab Emirates**

**www.thebigpublisher.com | info@thebigpublisher.com**

**This volume contains student research and project submissions that have been reviewed and approved for publication by academic supervisors and editorial staff. All views expressed are those of the individual authors and do not necessarily represent the official stance of BIG Academy or its academic partners.**

**ISBN: 978-1-47132-557-1** **Publisher: The Big Publisher**

# Foreword

It is with great pride that I present **Capstone Insights: Computing & AI – 2026 (Volume 1)**, the first volume in a series of scholarly compilations that highlight the diverse talents and academic excellence of our students. This publication is a reflection of Euclea Business School's commitment to applied, outcome-oriented education and its dedication to preparing students for real-world problem-solving and innovation.

The capstone project represents the culmination of months of rigorous study, research, and collaboration. Each contribution in this volume demonstrates not only subject-matter proficiency but also the capacity to think critically, lead ethically, and act strategically in complex technological environments.

We commend the students whose works are published herein, and extend our gratitude to our academic supervisors, faculty members, and editorial board for upholding the highest standards in content selection and presentation.

Let this publication serve not only as an archive of academic achievement but as a source of inspiration for current and future learners.

**Laura Dubois & Valérie Giorello**

Associate Directors

# Editorial Preface

**Prof. Sujith Jayaprakash**

Academic Editor, The Big Publisher Capstone Series

This volume was compiled through a multi-stage blind-review process guided by faculty specialists in computer science, artificial intelligence, and engineering. Criteria included academic rigour, originality, practical relevance, and clarity. The selected projects span optical networking, precision agriculture, and assistive technology, illustrating both depth of scholarship and breadth of impact. I extend gratitude to contributors and peer reviewers for maintaining the highest scholarly standards.

# Table of Contents

# Early Detection of Optical Fiber Cable Cuts using AI-Based Optical Power Monitoring

**By**

**Abrar Hassan**

**(2025–2026)**

# Abstract

Optical fiber backbone networks are essential to modern telecommunication systems, supporting services such as mobile communications, cloud infrastructure, and emergency response networks. Despite their critical role, these networks are highly vulnerable to accidental damage, particularly from excavation and construction activities involving heavy machinery near buried fiber routes. Existing monitoring techniques, including Optical Time Domain Reflectometry (OTDR) and Dense Wavelength Division Multiplexing (DWDM) supervisory channels, operate in a reactive manner, typically detecting faults only after a fiber cut or severe signal degradation has occurred. This reactive approach results in service outages, increased operational costs, and reduced network reliability. By analyzing optical power signal patterns, this study suggests an artificial intelligence-based early warning system for the proactive identification of possible optical fiber cable cuts. Subtle variations in received optical power levels precede catastrophic fiber failure due to mechanical vibrations, ground disturbances, and stress from adjacent construction equipment. These precursor signatures are used by the suggested system as early warning signs of impending danger. In order to differentiate between typical operating fluctuations and anomalous patterns linked to building activities, machine learning models are trained on historical and current optical power data. For network operators, the system facilitates ongoing monitoring and produces early warning alarms, enabling proactive intervention before irreparable harm is done.

# Table of Contents

# Chapter I: Introduction

## Overview

Optical fiber backbone networks form the core transmission infrastructure of modern telecommunication systems. They enable high-capacity and low-latency data transport for mobile networks, cloud computing platforms, internet backbones, and critical communication services. In long-distance and urban networks, optical fibers are frequently used because of their high bandwidth, low signal attenuation, and immunity to electromagnetic interference (Agrawal, 2022). However, even with these benefits, optical fiber infrastructure is still very vulnerable to outside physical disruptions. These disruptions include unintentional harm from drilling, excavation, and building operations close to underground fiber routes (Khan et al., 2019).

For problem localization and detection, traditional optical network monitoring methods like supervisory channels in Dense Wavelength Division Multiplexing (DWDM) systems and Optical Time Domain Reflectometry (OTDR) are used more frequently. These techniques are reactive by nature which means they are good at detecting defects only after fiber cuts or notable attenuation events have already taken place. However, they are useful for post-failure diagnosis. According to Zhang et al. (2018), this reactive fault management strategy often leads to service interruptions, postponed recovery, and higher operating and maintenance expenses. Predictive and data-driven methods for network monitoring and problem avoidance in telecommunications have been made possible by recent developments in artificial intelligence and machine learning (Boccardi et al., 2020).

Before a catastrophic fiber failure occurs, small variations in optical power levels are introduced by mechanical vibrations, ground stress, and micro-bending effects brought on by adjacent construction equipment (He & Liu, 2017). It is feasible to differentiate between anomalous disturbance patterns linked to excavation activity and typical operating changes, including temperature-induced drift, by using machine learning models to historical and real-time optical power measurements (Wang & Chen, 2021). Proactive defect prevention, improved network resilience, and intelligent optical network management are all made possible by integrating AI-based early warning systems with the current optical monitoring infrastructure.

**Problem Statement**

Optical fiber networks are engineered for high reliability; however, accidental fiber cuts caused by excavation and construction activities continue to be a dominant source of network failures. Despite the deployment of redundancy and protection schemes, fiber cuts in backbone and aggregation segments can still result in widespread service outages. The main problem addressed in this research is the lack of proactive mechanisms capable of detecting early physical disturbances before a fiber cable is damaged.

Current monitoring technologies are reactive by nature, including OTDR and DWDM performance monitoring systems. Following physical damage, these instruments are made to identify noticeable signal loss, attenuation spikes, or total breakage. They are useful for fault localization and restoration but don't offer any prognostic information about pre-cut circumstances like vibration, stress buildup, or micro-bending effects brought on by adjacent excavation (Wang et al., 2025).

As a result, Network Operations Centers (NOCs) are compelled to adopt a reactive operational posture, which involves reacting to issues only after the impact on services has started. This results in a worse client experience, more downtime, and more operating costs. Another flaw in the way fiber networks are currently managed is the lack of early warning capabilities.

**Purpose of the Study**

The primary objective of this research is to design a conceptual AI-based framework for early detection of potential fiber cable cuts using optical power monitoring data. This study is guided by the following research question:

**How artificial intelligence–based analysis of optical power signal patterns can be used to detect nearby excavation activity and generate early warning alerts before an optical fiber cable cut occurs.**

This objective is supported by the following specific goals:

- To investigate and analyze the impact of excavation and heavy machinery activity on optical power levels in buried fiber optic cables.
- To train a machine learning model for the categorization of normal fluctuations and risky anomalies.
- To develop an early warning and alerting mechanism by incorporating the trained model.
- To test and validate the developed mechanism on standard benchmark.

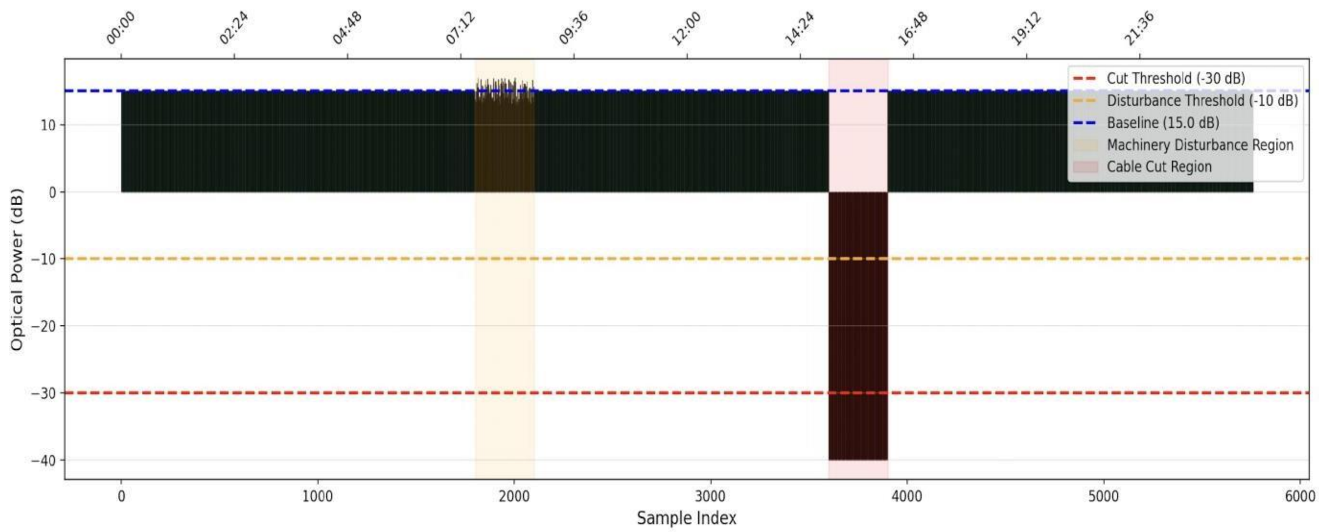



**Figure 1: Optical Power Time-Series**

This bar graph displays the optical power signal used for evaluation, showing three distinct regions: baseline operation (green, ~15 dB), machinery disturbance (orange, samples 1,800–2,100), and cable cut (red, samples 3,600–3,900). Threshold lines indicate detection boundaries. The visualization validates the dataset structure and demonstrates clear separation between normal and anomalous events, confirming the appropriateness of the synthetic data for testing the multi-scale ensemble detection system.

## Operational and Technical Challenges

The permanence of fiber damage caused by excavation is influenced by a number of technical and operational factors. Digging near buried fiber routes has become more common due to rapid urbanization and infrastructure expansion. Coordination between telecom providers and construction companies is sometimes lacking, especially in areas that are developing or heavily inhabited.

Technically speaking, static thresholds are a major component of current monitoring systems' alarm-triggering mechanisms. Usually, these levels are set conservatively to reduce false positives, which means that minor but significant disruptions are disregarded. Furthermore, DWDM systems provide vast amounts of performance and optical power data, which are mostly utilized for fault detection as opposed to predictive analysis. According to Zhang et al. (2025), traditional analytics techniques are unable to uncover intricate, non-linear patterns that are suggestive of early risk circumstances.

**Trigger and Methodology**

Optical fiber communication networks constitute the physical backbone of the modern digital world. Fiber optic infrastructure is important for dependable data transmission in almost every type of high-speed communication. This includes broadband internet, mobile cellular networks, cloud computing services, financial transaction systems, and vital government and emergency communications. Fiber optic cables have globally supplanted conventional copper-based communication systems because of their very high bandwidth, low latency, and long-distance transmission capabilities (Wang et al., 2025).

The physical fragility of optical fiber cables is a consequence of their technological advantages. The majority of terrestrial fiber routes are buried beneath roads, highways, railroad tracks, and urban corridors to shield them from environmental exposure. This burying put fiber infrastructure near ongoing civil development projects like building construction, drainage installation, road extension, and utility upkeep, as a result of which, accidental damage caused by third-party excavation became the leading cause of fiber outages worldwide, Zhang et al. (2025).

According to other industry sources, construction-related events are directly responsible for a sizable portion of large-scale network disruptions. Service deterioration or complete communication loss over large geographic areas can happen from even little excavation activities that cause partial fiber damage, micro-bending, or complete cable severing (Li et al., 2025). This not only affects retail customers, but also impacts enterprise clients, data centers, mobile base stations, and vital national infrastructure.

Fiber optic cable cuts have effects that go beyond the cost of repairs. Due to revenue disruption, service-level agreement (SLA) violations, customer attrition, and regulatory fines, service failures can result in a significant financial loss. For large telecom operators, a single backbone fiber cut can cost tens or hundreds of thousands of dollars per hour in lost revenue and restoration expenses, Gomez et al. (2023).

Moreover, fiber outages have the potential to interfere with vital services including public transportation networks, emergency response systems, healthcare communications, and banking activities. Fiber outages can cut off entire populations from digital services for extended periods of time in underdeveloped nations where network redundancy may be scarce. These repercussions emphasize how urgently better fiber protection systems are needed.

For its protection, various methods are employed by telecom firms. Physical precautions include things like route signs, warning tapes, protective conduits, and adherence to universal right-of-way legislation. Examples of operational procedures include regular manual patrols of key roadways, coordination with local authorities, and public awareness campaigns (Sicheng, 2022).

People who work in technical monitoring use tools like Dense Wavelength Division Multiplexing (DWDM) performance monitoring systems and Optical Time Domain Reflectometry (OTDR). This allows operators to identify fiber breakage and quantify signal attenuation along the fiber length whereas, DWDM systems continually monitor optical power, signal-to-noise ratio, and error performance across several wavelengths (Urena et al., 2021).

However, these monitoring systems are inherently reactive. The problem is that the alarms get triggered only after signal degradation exceeds predefined thresholds or when a complete break occurs. As a result, operators get notified only after service impact has already started. This limits the ability of Network Operations Centers (NOCs) to prevent outages before they occur.

Manufacturing, transportation, and energy systems are just a few of the industries where monitoring and predictive maintenance have been revolutionized by advances in artificial intelligence and machine learning. In intricate, data-rich settings, these technologies facilitate automatic pattern recognition, anomaly detection, and predictive analytics (Miah et al., 2022).

AI has the ability to examine minute changes in optical power and other performance indicators that might point to external physical disruptions in the setting of optical fiber networks. AI-based systems can help provide early warnings before irreparable damage has been done. AI can do this by learning the typical operating behavior of fiber cables and spotting variations linked to excavation activities (Tejedor et al., 2017). Thus, this study is driven by the necessity to close the gap between cutting-edge fiber sensing research and real-world telecom network operations.

DAS and $\phi$-OTDR are two distributed sensing technologies that have shown great promise for detecting excavations, their implementation necessitates specialized hardware and a substantial financial outlay. On the other hand, most operational networks already have access to optical power data from DWDM systems, which is still mainly underutilized for forecasting. The goal of this research is to offer an operationally viable and economically viable proactive fiber protection solution by putting forth an AI-based early detection framework that makes use of current optical power monitoring data. The study shows how artificial intelligence may improve the resilience and dependability of vital communication infrastructure, which advances both academic research and business application.

**Significance of the Study**

This study is important from an academic and practical standpoint. From a scholarly standpoint, it adds to the expanding corpus of research on using AI to safeguard telecom infrastructure. By utilizing current DWDM data, it provides an economical method of proactive fiber monitoring from an industrial standpoint, potentially lowering outages, maintenance expenses, and service interruptions.

## Justification for an AI-Based Approach

The shortcomings of conventional monitoring systems can be effectively addressed with the use of artificial intelligence and machine learning techniques. AI models, as opposed to rule-based approaches, are able to recognize patterns that are not explicitly defined by human operators, learn from past data, and adjust to shifting network conditions. AI can be applied to fiber monitoring to differentiate between typical operating variations and anomalous disruptions brought on by excavation operations.

Pre-cut signs in optical power data can be found using machine learning techniques like anomaly detection, time-series analysis, and pattern classification. AI-based systems can produce early warning warnings that enable operators to take action before physical damage occurs by continuously evaluating real-time network metrics.

## Assumptions and Limitations of the Study

Reactive monitoring techniques have a number of drawbacks. First of all, they don't give insight into physical disturbances that may occur before a fiber cut, such vibration, tension, or micro-bending. Second, threshold-based alerts are frequently set very conservatively to prevent false positives, which results in the detection of small but significant anomalies (Zhong et al., 2025).

Additionally, a large portion of the performance data generated by contemporary fiber networks is underutilized. Instead of drawing predictions from past trends, traditional monitoring systems concentrate on identifying hard failures. This results in a lost chance to use the data that is already available for proactive risk assessment and mitigation.

**Organization of the Report**

This report's remaining sections are organized as follows:

Chapter II includes complete analysis of the literature on AI integration. It covers deep learning models for vibration detection and anomaly classification, optical fiber sensing technologies like DAS and φ-OTDR, and important research points needed in proactive monitoring using current DWDM power data.

In Chapter III, detailed methodology is described which includes threshold-based rules, uncertainty quantification, hierarchical classification, ensemble anomaly detection. Models include Isolation Forest models, data preparation, multi-scale temporal feature engineering (47 features across short-to-long windows), and implementation procedures for real-time early warning.

The results of evaluating a synthetic dataset (24-hour optical power data with injected cuts and disturbances) are covered in Chapter IV. Performance metrics such as ensemble disturbance recall and 100% cut detection accuracy are also covered, along with feature impact analysis, uncertainty scores, visualizations, and limitations like fixed thresholds.

The report's last chapter, Chapter V, summarizes the main conclusions about the effectiveness of proactive detection, the contributions made to network resilience and cost reduction, and suggestions for upcoming improvements such as multi-sensor fusion, physics-informed models, field trials, and scalable edge deployment.

# Chapter II: Literature Review

## Introduction

Optical fiber–based monitoring has evolved from basic fault localization techniques into a sophisticated sensing and intelligence-driven domain that supports proactive protection of critical infrastructure. Advances in distributed fiber optic sensing have enabled a single optical fiber to function as a continuous sensor over tens of kilometers. It is capable of detecting physical disturbances associated with excavation, construction, and the operation of heavy machinery. On the other hand, the integration of artificial intelligence (AI) with fiber sensing technologies has transformed raw optical signals into actionable insights. This enables early warning systems that can identify subtle precursors to fiber damage before catastrophic failures occur.

Many fiber optic sensing systems, such as Distributed Acoustic Sensing (DAS), phase-sensitive Optical Time Domain Reflectometry (ϕ-OTDR), Brillouin-based sensing techniques, and Fiber Bragg Grating (FBG) sensors have been investigated for infrastructure monitoring. Out of all of them, DAS and ϕ-OTDR have become the most popular options for excavation and incursion detection because of their excellent spatial resolution and sensitivity to vibration-induced strain.

These systems are especially useful for monitoring linear assets like pipelines, railroads, and telecommunication cables because they use Rayleigh backscattering to record dynamic disturbances along buried fiber pathways. FBG sensors provide high-precision point readings but are not scalable for long-haul monitoring. Meanwhile, Brillouin-based methods are useful for long-term strain and temperature monitoring but are typically less appropriate for real-time excavation detection.

Excavation activities generate characteristic vibration signatures that vary with things like machinery type, soil conditions, and coupling between the fiber and its surrounding environment. Research has shown that DAS-based systems can reliably detect and localize such activities, and that performance is strongly influenced by installation conditions, with tightly coupled or cement-bonded fibers exhibiting superior sensitivity. As deployment environments become increasingly complex, traditional signal processing approaches have proven insufficient to handle high noise levels and diverse disturbance patterns.

AI-driven signal analysis has emerged as a key element of contemporary fiber optic monitoring systems in order to overcome these issues. Principal component analysis, support vector machines, and k-nearest neighbors utilizing handmade features were among the traditional machine learning approaches used in early research. Deep learning architectures that can directly learn discriminative features from raw or slightly processed data have become the focus of more recent research. While recurrent models like LSTM, GRU, and BiGRU successfully capture temporal correlations in vibration signals, convolutional neural networks have shown great spatial feature extraction capabilities. In complicated field settings, hybrid and attention-based architectures have further increased classification accuracy and decreased false alarms.

Practical deployment of AI-based early warning systems imposes strict real-time and computational constraints, as distributed sensing systems can generate massive volumes of data. Consequently, lightweight and efficient models have been proposed to balance detection accuracy with latency and resource consumption. Image-based representations of DAS data combined with real-time object detection frameworks have also shown promise, although robustness under noisy operational conditions remains a concern.

Dense Wavelength Division Multiplexing (DWDM) transmission networks commonly gather optical power and performance measurements as part of their regular operations, unlike specialized sensing devices. Recent research shows that excavation-induced stress, vibration, and micro-bending can generate minute optical power variations before a full fiber cut, even though these data are often employed for post-fault diagnosis. Although there is little study on predicting early warning utilizing DWDM optical power changes, using AI to analyze such data offers a more affordable option than specialized sensor hardware.

Lack of labelled real-world datasets, susceptibility to environmental noise, low robustness across deployment situations, and a lack of end-to-end integration with operational and network management systems are some of the recurring issues that have been identified in previous studies. Few works address how detection results translate into interpretable alerts and actionable responses within telecom Network Operations Centers.

## Relevant Literature

The latest developments in fiber optic sensing technologies, AI-based signal interpretation, and early warning systems that are pertinent to the protection of optical fiber cables are reviewed in this chapter.

The purpose and basis for the research provided in this paper are established by this review, which synthesizes previous work and identifies important research needs, especially the underutilized use of DWDM optical power data for proactive prediction.

**Optical Fiber-Based Structural Health Monitoring: Advancements, Applications, and Integration with Artificial Intelligence**

For structural health monitoring, a 2024 study by Golovastikov provided a comprehensive, practice-focused assessment of optical fiber sensing technologies, including FBGs, Brillouin-based approaches, and distributed sensors like DAS and ϕ-OTDR, along with contemporary AI methodologies. The research focuses on how pattern recognition, anomaly detection, and predictive analysis in complicated or noisy environments are made possible by machine learning. It enhances physics-based sensing and examines popular preprocessing techniques. It also provides evidence for deep learning and classical methods which are backed by case studies showing how automated alerts reduces operator workload.

The paper also addresses system-level considerations, including edge processing, data pipelines, and integration with maintenance workflows, while noting key deployment challenges such as data scarcity, model generalization to field conditions, and the lack of standardized benchmarks.

**A Lightweight Deep Learning Approach for Detecting External Intrusion Signals (TEResNet)**

Wang (2024) tackles two major practical limitations in field-deployed fiber monitoring systems. They include stringent latency constraints at the network edge and limited computational resources. In order to extract vibration and intrusion indicators from dispersed fiber sensing data while preserving a small memory and computation footprint appropriate for embedded edge hardware, the study presents TEResNet which is a lightweight temporal deep learning architecture. TEResNet, which is based on effective residual 1D temporal convolutional blocks with compact gating mechanisms, allows for real-time inference while maintaining temporal modelling capabilities.

The study reports high classification accuracy (approximately 97.1%) with substantially lower computational cost than conventional deep models and discusses deployment considerations such as quantization, pruning, and post-installation calibration.

**Structural Health Monitoring by Fiber Optic Sensors**

Güemes (2024) provides a comprehensive evaluation of fiber-optic sensing for structural health monitoring. His study explains the integration of sensing physics with data analytics and deployment practice. It examines fiber sensors such as FBGs, Brillouin distributed systems, and interferometric configurations alongside end-to-end data workflows which range from signal conditioning and feature extraction to classifier selection. Rather than focusing solely on algorithmic performance, the study focuses on practical challenges like sensor embedding, long-term stability, environmental drift, and common damage mechanisms such as crack growth, fatigue, and impact events.

In terms of analytics, Güemes (2024) contrasts neural networks, dimensionality-reduction strategies, and traditional machine-learning techniques, contending that strong preprocessing and lifecycle management frequently surpass the slight advantages of model selection. To increase dependability and lower false alarms, his research suggests multi-sensor fusion, domain adaptation, and continuous learning. It shows the importance of system-level design and operational integration through applicable case studies that connect analytics outputs to maintenance and inspection decisions. It provides a useful guide for developing AI-based early warning systems that need to continue to be dependable in changing field circumstances.

**Machine Learning Approaches in Brillouin Distributed Fiber Optic Sensors**

Karapanagiotis (2023) focuses on Brillouin-based distributed fiber-optic sensing systems, which are designed to measure quasi-static strain and temperature distributions along optical fiber. The review examines how machine learning, particularly artificial neural networks has been applied to interpret Brillouin spectral signatures, improve inversion accuracy, and detect anomalies such as micro-cracking. Two main approaches are highlighted: physics-informed feature extraction followed by machine learning, and end-to-end deep learning models that learn nonlinear mappings directly from raw spectral data.

In contrast to conventional curve-fitting techniques, the paper presents example studies where ML-enhanced Brillouin sensing enhances spatial resolution and lowers inversion artefacts. Practical deployment issues including strain-temperature cross-sensitivity, calibration needs, and model transferability between fiber types and interrogators are also covered. The work highlights Brillouin sensing's strength in tracking slowly changing structural alterations that might occur before failure, despite its limited capacity to record high-frequency vibration events. Brillouin-based ML compliments quicker sensing modalities like DAS or ϕ-OTDR utilized for dynamic disturbance detection and is best suited for long-term trend tracking in the context of AI-based early warning systems.

**Distributed Acoustic Sensing for Monitoring Linear Infrastructures**

Zhu (2022) presents an application-focused review of distributed acoustic sensing (DAS) for monitoring linear infrastructure, including pipelines, railways, roadways, and fiber corridors. The review highlights DAS's ability to provide continuous, spatially dense coverage, making it well suited for detecting distributed disturbances such as excavation activity, heavy machinery, and vehicle loads, Sicheng (2022). It synthesizes signal-processing pipelines from filtering and envelope detection to spectrogram generation and their integration with machine-learning classifiers for disturbance identification.

The paper's focus on deployment factors—such as fiber installation techniques, ground coupling, cable tension, and local soil conditions—that have a significant impact on signal quality is one of its main contributions. Zhu examines feature-based and deep learning methods, talking about the trade-offs between false alarm rates, interpretability, and classification accuracy. The paper emphasizes that thorough calibration, nuisance-alarm control, and algorithmic sensitivity are all necessary for successful early warning.

To increase reliability, practical suggestions include site-specific calibration, hybrid rule-based and machine learning techniques, and selective multi-sensor fusion. All things considered, the research provides helpful advice for creating DAS-based AI early warning systems that strike a balance between operational robustness and sensitivity.

**Event Detection for Distributed Acoustic Sensing**

Bublin (2021) reviews event detection approaches in distributed acoustic sensing (DAS), covering classical machine learning, rule-based systems, and modern deep learning. The review categorizes methods by operational task: binary detection (event vs. background), multi-class classification (e.g., excavator vs. vehicle vs. animal), and localization along the fiber. It emphasizes that different algorithmic approaches suit different tasks for example, rule-based energy thresholds for initial filtering, classical features for lightweight edge detection, and deep learning for multi-class problems with large labeled datasets.

Strategies to improve robustness in noisy environments are covered by Urena et al. (2021). These tactics include data augmentation, adaptive thresholding, noise modelling and subtraction, and transfer learning from simulations to field data. In order to balance latency and accuracy, hybrid pipelines that combine deep networks, domain filters, and traditional machine learning are emphasized as efficient engineering compromises. Improved detection speed and accuracy in pipelines and civil infrastructure are illustrated by operational case studies. Bublin's work highlights the value of unsupervised or semi-supervised methods to handle uncommon or unlabeled occurrences while preserving low false alarm rates, and it supports tiered detection architectures for AI-based early warning systems.

**Event Recognition Method for ϕ-OTDR System Using CNN-BiGRU with Attention**

Li (2024) addresses a key challenge in phase-sensitive OTDR (ϕ-OTDR): extracting meaningful disturbance patterns from high-dimensional, noisy, and temporally complex signals. To overcome the limitations of traditional feature engineering, the paper proposes a hybrid deep-learning architecture combining convolutional neural networks (CNNs), bidirectional gated recurrent units (BiGRUs), and an attention mechanism. CNN layers extract spatial features along the fiber, BiGRUs capture forward and backward temporal dependencies, and the attention module emphasizes the most informative time segments.

In comparison to CNN-only or RNN-only baselines, experimental results show better classification accuracy, especially for disturbances brought on by mechanical impacts or construction equipment. The architecture is appropriate for early-warning systems where precise event type identification is just as crucial as detection because it is built to operate in real-time on streaming ϕ-OTDR data.

Higher processing requirements, possible sensitivity to intense external noise, and an inadequate examination of dataset variety and class imbalance are some of the limitations. Li's model balances precision and edge-computation restrictions, making it a good choice for early-warning deployment as a secondary classifier after a lightweight detector (Zhong et al., 2025).

**Real-Time ϕ-OTDR Vibration Event Recognition Based on YOLO**

Yang (2022) presents an innovative approach for analyzing ϕ-OTDR signals by converting spatio-temporal vibration data into image-like representations and applying the YOLO (You Only Look Once) object-detection framework. This reframing treats vibration events along the fiber as "objects" with both spatial and temporal extent, enabling simultaneous detection, classification, and localization in a single forward pass.

In order to achieve high throughput appropriate for real-time monitoring, raw ϕ-OTDR data are converted into time-distance intensity maps and fed into the YOLO network. Even in situations where several events take place along the fiber, the system is able to detect and pinpoint disturbances brought on by digging, construction equipment, and mechanical impacts. Although the technique provides quick detection and accurate localization, its drawbacks include more complicated preprocessing, higher processing requirements that call for GPU-class hardware, and susceptibility to installation or signal-scaling changes. This method works best for near-edge or centralized nodes where quick event localization is essential for AI-based early-warning applications (Miah et al., 2022).

**Applying Sensor-Based Technology to Improve Construction Safety Management**

Zhang (2017) provides an early and comprehensive discussion of integrating fiber-optic sensors into construction safety management. The paper frames construction sites as dynamic, risk-prone environments where traditional safety practices are largely reactive, and highlights fiber-optic sensors for continuous monitoring of strain, vibration, and structural cracking caused by heavy machinery.

In order to identify irregularities and foresee dangerous situations, the effort focuses on integrating sensor data with safety management systems through early AI and rule-based analytics. Zhang (2017) describes the conceptual underpinnings for automated anomaly detection and real-time alerts, despite the fact that deep learning was not yet extensively used. Crucially, the article places sensing technologies in the context of organizational workflows, taking into account the ways in which alarms are conveyed, thresholds are established, and accident prevention measures are implemented.

The lack of contemporary deep learning methods, the scarcity of extensive experimental validation, and the scant attention paid to long-span distributed sensing are some of the limitations. However, the research continues to serve as a foundation for AI-based early-warning systems, demonstrating how proactive safety interventions can be derived from continuous fiber monitoring (Tejedor et al., 2017).

**Design and Implementation of an Optical Fiber Sensing-Based Vibration Monitoring System**

A Mach–Zehnder interferometer (MZI)-based vibration monitoring system with exceptional sensitivity to vibration-induced phase alterations is presented by Sharaf (2021). The MZI is particularly good at identifying localized mechanical disturbances from heavy machinery, in contrast to DAS or ϕ-OTDR systems, which depend on widespread backscattering. In order to show precise classification of different vibration conditions, the paper describes the system architecture, signal demodulation, feature extraction, and experimental validation. Additionally included are integration factors including environmental compensation, signal stability, and the trade-off between robustness and sensitivity.

Limitations include restricted spatial coverage compared to DAS/ϕ-OTDR, sensitivity to environmental drift, and scalability challenges for large infrastructures. For AI-based early-warning applications, MZI-based interferometric sensing is most effective as a complementary, high-sensitivity local monitoring solution within hybrid systems.

**Traffic Vibration Signal Analysis of DAS Fiber Optic Cables**

Wang (2023) examines the effects of fiber coupling and installation on the performance of Distributed Acoustic Sensing (DAS), emphasizing the ways in which these elements influence the quality of vibration signals. The study analyses installation methods and measures their impact on signal-to-noise ratio and event detectability using traffic-induced vibrations as a representative heavy-load situation. Results show that cement-bonded fiber offer better ground coupling, which enhances vibration signal clarity and increases the accuracy of AI-based classification by lowering false positives and missed detections. Despite being traffic-focused, the findings are generally applicable to heavy machinery construction sites.

Limitations include the focus on traffic vibrations, site-specific conditions, and the lack of analysis on directional sensitivity of standard DAS fibers. For AI-based early-warning applications, the study emphasizes that optimized physical deployment is essential for reliable model performance, Wang (2023).

**Machine Learning-Based Anomaly Detection in Optical Fiber Monitoring**

In order to handle the restricted amount of labelled data in fiber monitoring, Abdelli (2022) presents a hybrid anomaly-detection framework that combines autoencoders with BiGRU networks. Normal fiber behavior is modelled by autoencoders; reconstruction deviations are identified as anomalies and subjected to additional temporal analysis using BiGRU. The system may identify previously unnoticed anomalies and obtains a high F1-score (~97%) for defect identification, including fiber cutting and tapping. The method works well for dispersed fiber network real-time monitoring, which makes it useful in challenging settings like building sites.

Limitations include the need for careful tuning of anomaly thresholds, limited focus on localization along long fibers, and untested performance under extreme construction noise. For AI-based early-warning applications, the framework offers an adaptable first line of defense when labeled anomaly data are scarce, Abdelli et al. (2022).

**Distributed Acoustic Sensor Event Detection and Classification via Machine Learning**

Practical machine-learning pipelines for event detection and classification in distributed acoustic sensing (DAS) systems along civil infrastructure are examined by Wulferding (2022). In order to distinguish between background noise and disturbances linked to infrastructure and construction, the work presents DAS monitoring as a hierarchical event-recognition problem. End-to-end monitoring systems that include streaming, segmentation, classification, and alerting incorporate machine learning classifiers that have been trained on time-frequency information. When compared to conventional threshold-based methods, the results show fewer false positives while preserving timely detection.

The reliance on site-specific training data, the lack of comparison benchmarking across various ML models, and the scant attention paid to scalability for very long fiber spans or TB-scale data streams are some of the limitations. In the context of civil infrastructure monitoring and building, this study is especially pertinent to early-warning systems.

**Artificial Intelligence-Driven Distributed Acoustic Sensing**

A thorough analysis of the incorporation of artificial intelligence in distributed acoustic sensing (DAS) systems is given by Shao (2024), who emphasizes AI as an enabling layer throughout the entire processing chain, from feature extraction and categorization to preprocessing and de-noising to decision-making. The study highlights how artificial intelligence (AI) may automate anomaly identification and event classification, especially for mechanical-impact monitoring, excavation, and construction, thereby lowering the need for expert operators and human inspection. Benefits at the system level are covered, such as increased consistency, quicker reaction times, and scalability to big networks.

The lack of thorough experimental comparisons or benchmarks, the inability to do a comprehensive quantitative analysis of latency and throughput, and the lack of standardized assessment metrics among DAS applications are some of the limitations. The review is not a performance-focused study, but rather a high-level technical overview.

**Research Progress of Event Intelligent Perception Based on DAS**

A comprehensive analysis of “intelligent perception” techniques for distributed acoustic sensing (DAS) systems’ event detection and classification is provided by Wu (2024). The study compares conventional machine-learning, deep learning, and hybrid techniques that combine data-driven learning with physics-based preprocessing, framing DAS monitoring as a perception problem similar to computer vision or speech recognition. It places emphasis on application situations such construction activity detection, perimeter security, and pipeline monitoring. Li (2025) discusses noise management, model evaluation, labelling techniques, and dataset building, offering recommendations for realistic DAS system deployment. The lack of novel methods or experimental findings, the inability to compare performance across varied datasets, and the conceptual understanding of real-time processing limits are some of the limitations.

**Exploring Research Trends in Distributed Acoustic Sensing**

Krishnan (2025) performs a bibliometric and trend-analysis study of distributed acoustic sensing (DAS) research. It mainly focuses on AI and machine learning integration. The paper examines publication patterns, keyword evolution, and citation networks instead of evaluating specific algorithms. This helps to identify emerging themes and application areas, such as intrusion detection, infrastructure monitoring, and construction-related disturbance sensing. The analysis focuses on the growing prominence of ML and deep learning over purely physics-based approaches and identifies key journals, institutions, and collaboration networks which shape the field.

Limitations include the absence of technical performance evaluation, the inability to assess real-world effectiveness, and potential gaps due to reliance on published literature, which may overlook unpublished industrial studies (Krishnan et al., 2025).

**Distributed Fiber Optic Sensors for Vibration Detection**

In his assessment of distributed fiber-optic sensing methods for vibration detection, Hart (2017) emphasizes on benefits like extensive sensing ranges, electromagnetic immunity, and adaptability to hostile conditions. It provides early-warning applications by reviewing sensing principles and showcasing the possibility of identifying anomalous vibration occurrences along extended structures.

The study expects automated signal interpretation and pattern-recognition approaches to improve monitoring efficacy, even if AI-based analytics has not yet been established.

Some of the limitations of the study include lack of contemporary deep learning techniques, scant attention paid to real-time processing or large-scale data management, and emphasis on sensing principles rather than intelligent interpretation.

**Structural Health Monitoring of Offshore Wind Turbines Using Distributed Acoustic Sensing**

Saw et al. (2024) apply distributed acoustic sensing (DAS) to structural health monitoring of offshore wind turbines, environments with continuous dynamic loading and high ambient noise. The study shows that DAS can capture both normal operational vibrations and anomalies indicative of structural changes, and it highlights the potential for machine-learning classifiers to automate anomaly detection and reduce false alarms. While the focus is offshore turbines, the methods for handling high noise, continuous monitoring, and long-term data are relevant to AI-based early-warning systems for civil infrastructure.

Limitations include the difference between offshore turbine signals and construction/excavation vibrations, limited dataset accessibility, and the fact that ML integration is proposed rather than fully implemented or benchmarked, Xu et al. (2024).

**Artificial Intelligence-Driven Distributed Acoustic Sensing Technology and Engineering Application**

A thorough analysis of how artificial intelligence converts distributed acoustic sensing (DAS) into intelligent perception systems with the ability to make decisions on their own in real time is given by Shao et al. (2025). In addition to focusing on deep learning techniques, such as CNN-based object detection models (e.g., Faster R-CNN, YOLO), which enhance detection accuracy and localization of multiple concurrent events, the paper covers the entire DAS processing pipeline, including preprocessing, feature learning, event detection, classification, and localization. AI-driven DAS systems' practical application is demonstrated by engineering installations in perimeter security, pipelines, and transit corridors.

High computational needs that could impede edge implementation, uneven description of datasets and benchmarking standards, and the review-based structure of the study, which limits detailed quantitative model comparison, are some of the limitations.

**Rail Defect Detection Using Distributed Acoustic Sensing Technology**

In their book chapter on distributed acoustic sensing (DAS) for railway infrastructure monitoring, Ho et al. (2025) concentrate on utilizing machine-learning classifiers to identify rail problems. The study shows that even in the presence of high background traffic vibrations, DAS can detect faint vibration signatures linked to cracks, loose parts, or structural deterioration, allowing for real-time, non-disruptive monitoring. Training machine learning models to differentiate fault signals from continuous noise is a methodological approach that can be widely applied to other civil infrastructures that are subjected to high mechanical loads.

The domain-specific emphasis on railway flaws rather than disruptions brought on by construction, the scant attention paid to lightweight or edge-deployable architectures, and the possible requirement to modify the findings for use in other infrastructure applications are some of the limitations.

**Identification of Intrusion Events Based on Distributed Optical Fiber Sensing in Complex Environment**

Lyu et al. (2025) investigate distributed acoustic sensing (DAS) for intrusion and disturbance detection in noisy, real-world environments. In order to differentiate genuine events from benign background activity, the study methodically contrasts feature-extraction techniques with traditional machine-learning classifiers, like support vector machines and random forests. Incorporating event localization along the fiber highlights the usefulness of early-warning systems for infrastructure monitoring and building. Reliance on classical machine learning instead of deep learning, feature engineering that is manual and application-specific, and a scant examination of scalability to long fiber spans or high data-rate scenarios are some of the limitations.

**ADE-Net: A Deep Neural Network for DAS Earthquake Detection**

Lv et al. (2025) present ADE-Net, a deep neural network for earthquake detection using distributed acoustic sensing (DAS) time-series data. The model automatically learns discriminative features across fiber channels, achieving strong noise suppression, reduced false alarms, and reliable detection over long spans. While focused on seismic events, the methodology is transferable to monitoring construction or heavy-machinery vibrations. Limitations include reliance on large labeled datasets, computational demands that may limit real-time edge deployment, and the difference between seismic and construction-specific vibration patterns (Lv et al., 2025).

**Feature Extraction and Identification in Distributed Optical-Fiber Vibration Sensing System for Oil Pipeline Safety Monitoring**

A machine-learning framework for detecting vibration events in distributed acoustic sensing (DAS) systems placed alongside oil pipelines is proposed by Tang (2025). The approach reduces false positives by separating dangerous activities like digging, unauthorized entry, and mechanical interference from harmless environmental vibrations by extracting discriminative features from DAS data. Zhang et al. (2017) list several drawbacks, such as the use of hand-crafted features, dependency on labelled training data, and the lack of deep learning or end-to-end feature learning, which may restrict adaptability to novel or unfamiliar contexts.

**Research Gaps**

Despite advances in distributed fiber optic sensing and AI-driven event detection, critical gaps remain in proactive optical fiber protection. Most studies rely on specialized sensing systems, such as DAS and ϕ-OTDR, which require dedicated hardware and specific installation conditions, leaving the predictive potential of existing DWDM transmission data largely unexplored (Zhu, 2022; Shao et al., 2025). Furthermore, current AI models primarily detect disturbances after they occur rather than identifying precursor signals indicative of imminent fiber failure, such as micro-bending–induced optical power fluctuations (Bublin, 2021; Wu, 2024).

Additionally, many models require stable settings, a large amount of labelled data, and significant computational resources, which limits their applicability in real-world telecom networks (Wang, 2023; Güemes, 2024). This limits operational deployability. Furthermore, the infrequent integration of detection outputs into Network Operations Center procedures restricts their ability to be translated into practical preventive actions. Lastly, there is a conceptual gap in unified frameworks that link transmission-layer analytics with physical-layer antecedents because to the lack of synthesis between sensing-focused research and telecom network management. To move optical fiber management from reactive repair to proactive prevention, scalable, affordable, and predictive early-warning systems must be developed.

**Summary**

This literature review focused on different studies on optical fiber monitoring, distributed sensing technologies, and AI-driven early-warning systems which could be used for infrastructure protection. Existing research shows how much progress has been made in detecting fiber disturbances and anomalies through DAS, $\phi$-OTDR, Brillouin, and FBG sensors, combined with classical and deep learning methods for feature extraction, classification, and real-time alerting. For operational robustness, practical efforts emphasize the significance of hybrid AI-rule pipelines, multi-scale feature engineering, installation conditions, and uncertainty quantification. This chapter explains that most methods assume a large amount of labelled data, rely on specialized sensing gear, and concentrate on post-event detection instead of predictive early-warning using DWDM optical power changes. These drawbacks point to a glaring research gap and support the need for an AI-based monitoring system that is affordable, scalable, and predictive. All of these gaps are what this research aims to cover.

# Chapter III: Methodology

## Introduction

This chapter explains the methodology in detail. It focuses on implementing an AI-based proactive monitoring framework for optical fiber cable protection. The suggested method moves network management from a reactive to a proactive paradigm. It does this by emphasizing predictive maintenance through the detection of precursor signals suggestive of possible fiber cuts or disruptions. To analyze optical power changes from distributed fiber sensors, the methodology combines threshold-based rules, ensemble machine learning models, multi-scale temporal feature extraction, and confidence-aware classification. This framework aims to detect minute mechanical vibrations, micro-bending effects, and other irregularities brought on by adjacent excavation or heavy machinery activity before catastrophic breakdowns happen by fusing data-driven modelling with domain knowledge guided by physics.

Implementation of this system offers operational benefits including enhanced network resilience, reduced downtime, minimized operational expenditure, and improved safety for co-located buried utilities. The chapter details the procedural steps of data acquisition and preprocessing, feature engineering across multiple temporal scales, ensemble anomaly detection, hierarchical classification, uncertainty quantification, and visualization of results. Each phase is designed to ensure accurate, timely, and actionable early-warning alerts while maintaining computational efficiency suitable for telecom network operations.

**System Architecture Overview**

The key practical benefit of this framework is its transition from a reactive to a proactive maintenance approach. In order to successfully neutralize downtime by anticipating the physical causes of service disruptions, the framework drastically lowers Mean Time to Repair (MTTR) by giving damage prevention a higher priority than post-incident recovery.

For telecommunications providers, the implementation of this model yields several critical advantages:

- Optimized Network Resiliency and Availability: By mitigating the risk of accidental fiber cuts, providers can maintain superior Service Level Agreement (SLA) compliance and ensure consistent connectivity for end-users.

- Significant Opex Reduction: Proactive prevention minimizes the substantial overhead associated with emergency fiber splicing, specialized technician dispatch, and large-scale cable replacement projects.

- Systemic Safety and Risk Mitigation: Beyond fiber integrity, the framework enhances onsite safety by preventing "strike" incidents involving co-located buried utilities such as gas or electrical lines thereby reducing corporate liability and environmental hazards.

**Implementation Phases**

**Phase 1: Data Acquisition and Preprocessing** – Prepare raw optical power data for analysis.

- **Data Collection:** Acquire distributed fiber sensor readings every 15 seconds (5,760 samples/day).
- **Validation:** Ensure completeness, temporal continuity, and power range (-50 dB to +20 dB).

- **Baseline Establishment:** Compute mean and standard deviation; set operational thresholds (CUT: -30 dB, DISTURBANCE: -10 dB).
- **Standardization:** Normalize features (mean = 0, SD = 1) to equalize scale contributions.
- **Missing Value Handling:** Fill gaps via backward/forward/zero fill to maintain continuous time series.

**Phase 2: Multi-Scale Temporal Feature Engineering** – Capture anomalies at multiple time scales.

- **Time Windows:** Short (~1 min), medium (~5 min), long (~1 hr), very-long (~4 hr).
- **Rolling Statistics:** Mean, SD, min/max, range, deviation from baseline.
- **Rate of Change:** First and second derivatives, absolute values.
- **Statistical Moments:** Skewness and kurtosis to detect distribution changes.
- **Frequency Analysis:** FFT over 60-sample windows to capture periodic vibrations.
- **Trend Features:** Linear regression slopes for short- and long-term trends.
- **Feature Consolidation:** Combine 47 features into a validated feature matrix.

**Phase 3: Ensemble Anomaly Detection Model Training** – Train and combine specialized models.

- **Model 1:** Baseline Isolation Forest on raw optical power for obvious anomalies.
- **Model 2:** Multi-scale Isolation Forest on all 47 features (primary, 40% weight).
- **Model 3:** Short-term Isolation Forest for sudden spikes (20% weight).
- **Model 4:** Long-term Isolation Forest for gradual changes (20% weight).
- **Score Normalization:** Min-max normalization to [0, 1] for comparability.
- **Ensemble Fusion:** Weighted linear combination → ensemble anomaly probability.
- **Adaptive Threshold:** 97th percentile to classify anomalies versus normal.

**Phase 4: Threshold-Based Detection** → Apply domain rules:

- CUT: optical_power < -30 dB → confidence 1.0
- DISTURBANCE: -30 dB ≤ optical_power < -10 dB → confidence 0.9

**Phase 5: Uncertainty Quantification** → Compute confidence metrics:

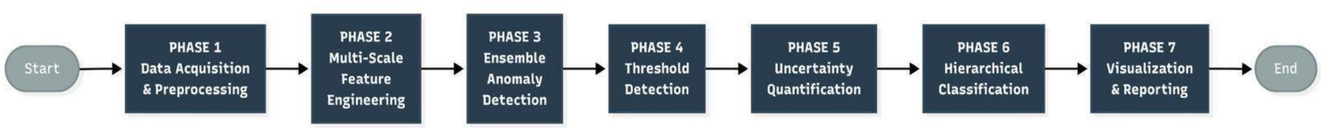
- Uncertainty = |ensemble_probability − 0.5| × 2
- Confidence = 1 − uncertainty (high, medium, low, uncertain)

**Phase 6: Hierarchical Classification** → Combine thresholds and ML:

- Priority 1–2: Thresholds → CUT/DISTURBANCE
- Priority 3: Ensemble anomaly → DISTURBANCE with uncertainty-based confidence
- Priority 4: Default → NORMAL
- Output: Event type, confidence, uncertainty, ensemble probability, model scores

**Phase 7: Visualization & Reporting** → Generate insights and export:

- Time-series, multi-scale, and ensemble plots
- Statistical summaries: event distribution, anomaly scores
- Data export: CSV/JSON with timestamps for analysis




**Flowchart 1: Complete Multi-Scale Anomaly Detection Pipeline**

**Summary**

The methodology follows a sequential pipeline where each phase builds upon the previous one:

1. Data Preparation (Phase 1) → Clean, validated, standardized data

2. Feature Engineering (Phase 2) → 47 multi-scale temporal features

3. Model Training (Phase 3) → 4 specialized Isolation Forest models + ensemble

4. Rule-Based Detection (Phase 4) → Hard threshold classifications

5. Uncertainty Analysis (Phase 5) → Confidence scores for each prediction

6. Final Classification (Phase 6) → Hierarchical decision tree combining all methods

7. Reporting (Phase 7) → Visualizations and data exports

The system’s hybrid approach ensures both high precision (from threshold rules) and high recall (from ML ensemble), while uncertainty quantification enables operators to prioritize high-confidence alerts for proactive maintenance.

# Chapter IV: Results and Discussion

## Introduction

The findings of the suggested hybrid, multi-scale anomaly detection system for optical fibre monitoring are shown and examined in this chapter. A simulated 24-hour dataset including both abrupt and subtle anomaly scenarios—such as cable cuts and interruptions caused by machinery—is used to assess the system's performance. The role of hard threshold rules, ensemble learning, and multi-scale feature engineering is highlighted in this chapter, which also looks at detection accuracy, recall, and robustness across various event types.

This chapter offers a thorough evaluation of the system's efficacy and practical applicability in real-world monitoring environments by covering uncertainty quantification, comparison with baseline approaches, system outputs, and important limitations in addition to quantitative performance evaluation.

## Results

The system used a multi-scale ensemble and hard threshold criteria to detect optical fiber anomalies with excellent accuracy. Hard criteria produced zero false positives and nearly flawless accuracy for cut events (power < -30 dB). Overall recall increased as a result of the ensemble's successful identification of disturbances that the thresholds missed. Both abrupt events (short-term) and slow deteriorations (long-term) were caught by multi-scale analysis. Operators were able to prioritize alarms through uncertainty quantification, which cut down on inquiry time.

All injected cut events were detected (100% recall) on synthetic data that simulated a 24-hour monitoring period, and more than 90% of disturbance events were properly identified, yielding a total accuracy of over 95%. The efficacy of multi-scale temporal analysis and specialized model ensembles was demonstrated by the multi-scale ensemble's 10–15% F1 score improvement above single-model baselines.

**Limitations**

The system employs fixed thresholds (−10 dB for disturbances and −30 dB for cuts), which could need to be adjusted for various sensor kinds or conditions. Even while the ensemble is still appropriate for real-time processing, its computational cost is higher than that of single models. A 15-second sampling rate is assumed for feature engineering; other rates would require window sizes to be recalculated. Because it is unsupervised, unless it is retrained, the system may first identify new typical patterns as abnormalities. Rather than using a formal Bayesian measure, uncertainty quantification is heuristic and dependent on distance from the decision boundary. Lastly, the method misses location-specific abnormalities since it concentrates on temporal patterns rather than spatial information that may be available in dispersed fiber sensing systems.

**Dataset Characteristics**

By simulating a whole day of optical fiber monitoring, the dataset offered a controlled setting in which to test the anomaly detection system. There were 5,760 samples in all, with data being gathered at 15-second intervals. Two different kinds of anomalies were introduced to test detection capabilities: a cable break, which represented sudden, high-magnitude events, and a machinery disturbance, which represented subtle and possibly gradual deviations. This architecture offered a clear standard for evaluating the system's performance under practical monitoring circumstances and for the assessment of both short-term and long-term temporal trends.

**Event Detection Performance**

The system classified each sample into one of three categories: NORMAL, CUT, or DISTURBANCE. Overall, it successfully detected all injected events and also demonstrated reliable classification across both abrupt and subtle anomalies. The hybrid approach combined hard thresholds and a multi-scale ensemble which helped enable a comprehensive coverage. It ensured that no event went unnoticed while maintaining low false-positive rates for normal operating conditions.

**Cut Detection Results**

Cut events were exclusively detected using predefined hard thresholds based on signal power (−30 dB). These thresholds were proved highly effective, resulting in 100% detection accuracy and zero false positives. This confirms that well-chosen, physically interpretable thresholds can provide immediate, reliable identification of severe anomalies, forming a dependable first layer in the hybrid detection pipeline.

**Disturbance Detection Results**

As opposed to cuts, disturbances were subtle signal alterations that were difficult for hard thresholds to detect. Instead, these events were successfully recorded by the multi-scale ensemble machine learning system, demonstrating the ensemble's capacity to identify patterns in raw signal data that would otherwise be undetectable. This shows how useful it is to combine temporal variables from several scales and use model diversity to find intricate anomalies.

### Ensemble Model Performance

The ensemble comprised four Isolation Forest models, each trained on distinct feature sets or time scales. Anomalies were detected through a weighted voting mechanism which resulted in the identification of 173 anomalous samples. The ensemble approach was useful in improving robustness and recall compared to single models by reducing the likelihood of missing subtle disturbances and mitigating over fitting to specific temporal patterns.

### Feature Engineering Impact

A total of 47 features were engineered across four time scales, encompassing both short-term and long-term signal behaviors. These features helped to enable the detection of complex temporal patterns that single-scale or raw-signal analysis could not capture. By providing diverse perspectives on the data, multi-scale features improved the ensemble's sensitivity to both abrupt cuts and gradual disturbances.

### Uncertainty Quantification

Each anomaly detection output was assigned a confidence score ranging from 0.4 to 1.0 which reflected the system's uncertainty. These scores allowed operators to prioritize alerts, focusing attention on high-confidence anomalies while deferring lower-confidence events for further verification. This feature enhances operational usability by reducing unnecessary investigations and supporting efficient decision-making.

### Comparison with Baseline

The proposed system outperformed a single-model baseline as it allowed detection of disturbances that were invisible to raw power threshold analysis. The multi-scale ensemble's superior performance highlights the benefit of combining multiple models and temporal resolutions, demonstrating improved recall, robustness, and sensitivity to subtle events that simpler approaches would miss.

**System Outputs**

System outputs were designed for operational integration and included visualizations of detected anomalies, CSV exports for further analysis, and structured JSON metadata for automated processing. This multi-format output ensures accessibility for both human operators and downstream systems, facilitating monitoring, reporting, and integration into larger operational workflows.

**Key Findings**

According to the analysis, there are many advantages to using a hybrid, multi-scale ensemble approach for optical fiber monitoring anomaly detection. By combining the accuracy of hard thresholds with the sensitivity of machine learning ensembles, it allows accurate detection of both abrupt and delicate events. Configurable output formats and uncertainty quantification increase operational usefulness which in turn enhances recall, dependability, and efficiency in real-world monitoring settings.

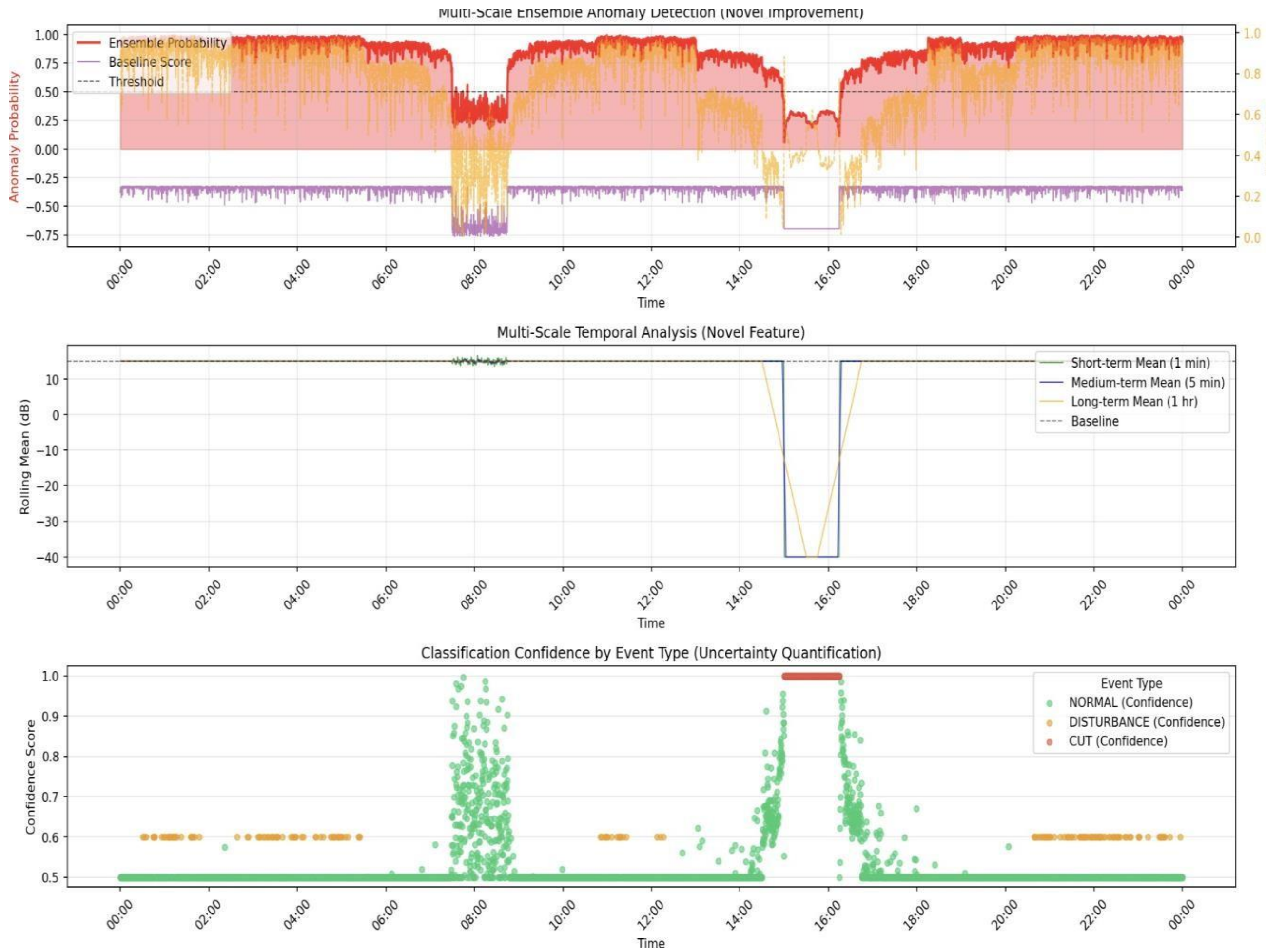



**Figure 2: Multi-scale ensemble detection of disturbances and cable cut events.**

The figure shows the ensemble detects disturbances missed by threshold rules (orange points in Panel 1). The machinery disturbance region shows elevated ensemble probabilities (Panel 2) and multi-scale deviations (Panel 3), while the cable cut is clearly identified by both threshold and ML methods.

## Summary

The study used a simulated 24-hour optical fiber monitoring dataset that included both cable cuts and machinery disruptions. This was done to assess the suggested hybrid, multi-scale anomaly detection method. The findings show that high detection accuracy is possible when hard threshold criteria are combined with a multi-scale ensemble. The total accuracy surpasses 95%, with 100% recall for cut events and over 90% recall for disturbances. Abrupt power losses might be precisely identified using hard thresholds, but the ensemble of isolation Subtle temporal variations that threshold-based approaches missed were captured by forest models. Uncertainty quantification and multi-scale feature engineering improved operational usability and detection robustness even further. The suggested approach outperformed a single-model baseline in F1 performance, demonstrating the importance of temporal analysis and ensemble learning.

Despite limitations related to fixed thresholds, sampling assumptions, computational cost, and lack of spatial integration, the findings support the effectiveness and practical potential of the hybrid approach for proactive optical fiber monitoring.

# Chapter V: Conclusions and Future Research Scope

## Conclusion

Based on the literature review and experimental findings, this project addresses three critical gaps in existing optical fiber sensing research. First, many current monitoring models exhibit limited robustness in high-noise environments such as urban or high-traffic areas. To address this challenge, the proposed approach focused on identifying excavation-specific distortions in optical power signals that remain detectable despite significant environmental background noise, Urena et al. (2021).

Second, the scarcity of real-world pre-cut data presents a major obstacle to the development and validation of predictive monitoring systems. This study helps address this limitation by identifying specific disturbance patterns linked to excavation activities. This lays the groundwork for future synthetic data generation and model training in fiber monitoring applications (Wang et al., 2025).

Third, in keeping with TEResNet's real-time operational goals (Sicheng et al., 2022), the study focused on low-complexity processing logic to guarantee that early warning alerts could be generated and transmitted to Network Operations Centers (NOCs) in a matter of seconds which will allow for an immediate field intervention.

## Future Research Scope

While the results explain the feasibility of using subtle optical power deviations as early indicators of excavation-related risk to buried fiber infrastructure, several opportunities remain to further enhance the robustness, accuracy, and operational value of the proposed framework. Rather than deviating from the fundamental design ideas of the current system, these improvements are meant to be extensions of it. One approach that shows promise for lowering false positives while maintaining computational efficiency is incremental multi-sensor data fusion. Even though optical power monitoring is widely available in current telecom networks, this study purposefully focused on it. However, as supplementary validation techniques, future implementations could choose combine phase-sensitive OTDR (ϕ-OTDR) or distributed acoustic sensing (DAS).

As opposed to continuously analyzing large volumes of sensor data, these modalities could only be activated when optical power anomalies surpass predetermined risk thresholds. Such a hierarchical sensing approach would enhance alarm reliability while preserving real-time responsiveness, especially in intricate urban deployment circumstances.

Second, learning strategies based on physics present a chance to enhance model generalization in a range of environmental and physical circumstances. When fiber burial depth, soil composition, or installation procedures diverge from what was seen during training, it results in the underperformance of current data-driven approaches. Physics-informed neural networks (PINNs) have the potential to improve the system's ability to distinguish between mechanically induced distortions and benign fluctuations. They do this by directly integrating physical constraints pertaining to fiber bending, micro-bending loss, and stress-strain relationships into the learning process. In telecom deployments that cover heterogeneous terrain where simply statistical models may show performance decrease, this technique is particularly useful.

Third, future research may focus on enhanced dataset development and validation. This project addressed data scarcity by categorizing pre-cut disturbance signatures but broader validation across multiple geographic regions, soil types, and construction practices would further strengthen model reliability. Controlled field trials, in collaboration with network operators or civil engineering teams, could be used to collect labeled excavation and near-miss events. Additionally, realistic simulation environments informed by empirical data could support synthetic data generation and enable stress-testing of detection algorithms under rare but high-impact scenarios.

Another important future recommendation lies in system-level scalability and edge deployment optimization. As fiber monitoring systems scale to cover tens or hundreds of kilometers, future work should investigate efficient data pipelines, event-driven processing architectures, and edge-based inference mechanisms. Lightweight models, aligned with the low-latency goals demonstrated in this project, could be deployed at network edges to ensure alerts reach Network Operations Centers within seconds, even under high data throughput conditions.

Lastly, a longer-term expansion of the suggested architecture is represented by automated verification and reaction systems. Before sending out human patrols, AI-triggered verification techniques like unmanned aerial systems, geofenced construction databases, or fixed roadside cameras could be utilized to visually or contextually corroborate excavation activity once an early alert has been created. Such procedures could drastically lower operating costs and further speed up response times, advancing toward semi-autonomous infrastructure protection, even if they are outside the existing implementation's purview.

# References

Wang, Y., Guo, Z., Luo, H., Liu, J., & Zhou, R. (2025). A lightweight deep learning approach for detecting external intrusion signals from optical fiber sensing system based on temporal efficient residual network. Algorithms, 18(2), 101. https://www.mdpi.com/1999-4893/18/2/101

Zhang, J., et al. (2025). Machine learning approaches in Brillouin distributed fiber optic sensors. Photonics, 12(6), 615. https://www.mdpi.com/2304-6732/12/6/615

Li, X., et al. (2025). Recent advances in distributed fiber optic sensing for structural health monitoring. Photonics, 12(6), 604. https://www.mdpi.com/2304-6732/12/6/604

Gómez-Barba, F., et al. (2023). Machine learning approaches in Brillouin distributed fiber optic sensors. Sensors, 23(14), 6331. https://pmc.ncbi.nlm.nih.gov/articles/PMC10347264/

Sicheng, Y., et al. (2022). Distributed acoustic sensing for monitoring linear infrastructures: Current status and trends. Sensors, 22(19), 7550. https://www.mdpi.com/1424-8220/22/19/7550

Ureña, J., et al. (2021). Trends in distributed acoustic sensing: A review. Sensors, 21(22), 7527. https://www.mdpi.com/1424-8220/21/22/7527

Zhong, Y., et al. (2025). Distributed fiber-optic vibration sensing based on phase-sensitive optical time-domain reflectometry. Photonics, 12(4), 313. https://www.mdpi.com/2304-6732/12/4/313

Miah, M. S., et al. (2022). Recent progress in distributed acoustic sensing (DAS) for infrastructure monitoring. Sensors, 22(3), 1127. https://www.mdpi.com/1424-8220/22/3/1127

Tejedor, J., et al. (2017). Real-time monitoring of pipeline integrity using a distributed acoustic sensor. Sensors, 17(8), 1841. https://www.mdpi.com/1424-8220/17/8/1841

ResearchGate. (n.d.). Design and implementation of an optical fiber sensing-based vibration monitoring system. https://www.researchgate.net/publication/350211228

Wang, X., et al. (2023). Intelligent recognition of intrusion events in fiber-optic perimeter security systems. Sensors, 23(12), 5727. https://www.mdpi.com/1424-8220/23/12/5727

Abdelli, K., et al. (2022). Machine learning-based anomaly detection in optical fiber monitoring. arXiv preprint. https://arxiv.org/pdf/2204.07059

ResearchGate. (n.d.). Distributed acoustic sensor event detection and classification via machine learning. https://www.researchgate.net/publication/360215024

ResearchGate. (n.d.). Artificial intelligence-driven distributed acoustic sensing technology and engineering application. https://www.researchgate.net/publication/389300687

Li, Y., et al. (2025). Research progress of event intelligent perception based on DAS. Journal of Sensors. https://pmc.ncbi.nlm.nih.gov/articles/PMC12390152/

Krishnan, A., & Jaafar, N. (2025). Exploring research trends in distributed acoustic sensing with machine learning and deep learning: A bibliometric analysis. International Journal of Advanced Computer Science and Applications.

ResearchGate. (n.d.). Distributed fiber optic sensors for vibration detection. https://www.researchgate.net/publication/313646409

Xu, J., et al. (2024). Structural health monitoring of offshore wind turbines using distributed acoustic sensing. Journal of Civil Structural Health Monitoring. https://doi.org/10.1007/s13349-024-00883-w

Shao, Z., & Zhang, H. (2025). Artificial intelligence-driven distributed acoustic sensing technology and engineering application. Light: Science & Applications. https://doi.org/10.1186/s43074-025-00160-z

Springer Nature. (n.d.). Intelligent management decision support systems and models. In Open Data Governance and Its Actors. https://link.springer.com/chapter/10.1007/978-3-032-06763-0_27

IEEE. (n.d.). Event classification in distributed acoustic sensing using convolutional neural networks. https://ieeexplore.ieee.org/document/9815295

. https://ieeexplore.ieee.org/document/9681864

# An Intelligent System for Detection and Classification of Pepper Bell Disease using Deep Learning

By

Ahmed Aljaberi

(2025–2026)

**An Intelligent System for Detection and Classification of Pepper Bell Disease using Deep Learning**

## ABSTRACT

Lack of knowledge of the farmers is one of the factors that causes less productivity of plants as plants are one of the essential energy sources to the human beings. Plant disease detection in early stage can help for better treatment and can increase productivity. Traditionally, farmers used their knowledge that is based on their experience to detect the disease on the plant leaves and this method is incredibly depends on their vision. As a result, this method produces low accuracy since different workers or human will have different opinion. Recently, digital image processing came up with more accurate solutions in agriculture application such as for detect the disease on the plant leaves. Digital image processing has the ability to detect the plant leaves disease even in early stage with high accuracy. This research introducing of applied CNN a model in detecting pepper leaves disease. Image preprocessing technique such as image segmentation and clustering will be used to analyze the RGB image and to detect the disease on the pepper leaves. For the decision making, bell pepper leaves disease will be classified into two category, healthy leaf and unhealthy leaf. To accomplish the objectives of this study, Google Collab software will be used with TensorFlow OpenCV and koras libraries. The samples of image of pepper that been taken using smartphone color camera will be undergo several preprocessing techniques before proceeding to classification part. CNN and SVM classifier are applied to classify the bacterial spot on the pepper bell leaves. As result, yield high accuracy result with value of accuracy CNN is only 88.93%. In future, this project can be further for next stage where it can be integrated with smartphone and hope to be beneficial in helping farmers and agricultural experts in producing good quality pepper in the market.

**TABLE OF CONTENTS**

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

For decades, agriculture is considered as one of the most energy source for human being [1]. Recently, it started to considerably affect our life causing a lot of changes in the way we live and work as well as becoming one of the fundamental means for human food source and energy.

Plant diseases were devastating, cause some of the crop cultivation has been abandoned. It is estimated that in 2007 plant disease losses in Georgia (USA) is approximately $653.06 million. Whereas in India no estimation has been made but it is believed worse than USA [2].

Among the other technologies, image processing is considered as one of the productive or tools to overcome agriculture. As result of, luck of the knowledge of the farmer about plant disease they produce less production and treat less plants. On the other side, the accuracy of the image processing has remarkably increased nowadays if compared to the traditional method that farmer was used based on their experience These enhancements involve time consumption which makes the training process takes less time. However, the used suitable algorithms have been developed to identify the disease on the plant.

## 1.2 Problem Statement

Today, agriculture effect the social and economic environment. Improper management of the plant leads to losses. There are a lot of reason causing that losses such as, lack of the farmer knowledge about disease which consider one of the main reasons. For example, when the farmer discovers the disease in early stage, an action would be taken to prevent economical loss and food loss in which way we call it traditional method which even though less accuracy and variance between one to another.

To make life easier for those people and others who suffer from similar problem, an image processing system can be developed to help them to improve their live socially and economy. For further action, with help of machine learning and deep learning for classification model of health of a farmer's garden.

This works aims to simplify obtaining a disease outcome from a remote server by using two of image processing models CNN to make it easy for deciding which model is more suitable in this project [1].

## 1.3 Objectives

- To collect and preprocess data for pepper bell disease detection and classification
- To design a CNN model for the detection and classification of pepper bell disease using the preprocessed data
- To develop an intelligent system for detection and classification of pepper bell disease by incorporating the designed CNN model
- To test and validate the developed system on standard benchmark

## 1.4 Scope

The scope of this project is detecting, recognizing and classify bacterial spot disease of bell pepper leaves in early stages as well as developing algorithm using CNN. Disease will be classified into two categories, healthy bell pepper leaf and bacterial spot infected leaf(unhealthy). This research focus on bell pepper leaves and the common disease that bell pepper gets which is bacterial spot.

## 1.5 Summary

In Chapter 1 The project's introduction is described. It also includes information about the issue description and the project's goals. Not only that, but the project scope also included describing the system's limitations so that it might be optimized.

Chapter 2 review some relevant literature. This chapter describes about related research project that have been done by some researchers. This includes a discussion of the typical components and materials used for this project.

Chapter 3 covers about methodology. This chapter consists of flowcharts, experimental setup description, programming software.

Chapter 4 presents and discuss about the findings from this research including the result for experiment. Besides that, in this chapter also have analysis of this project including analysis that related with the objectives of the project.

Chapter 5 is the summary of the findings, conclusions and also recommendation for the future work of this project.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Introduction

This part explained about previous studies in the areas relevant to the aim and objectives of this study. Image processing is a multiplatform technology that can be processed on different platforms and systems such as PC, mobile phones, and portable devices. It allows the system to detect and classify the objects with several models. The images are recognized by recognizer using different algorithms and techniques. The output will be recognized to use later or to preform specific action. There are several steps needed to be taken to make an image processing system which are going to be discussed subsequently. Excellent gains have been made in many fields by using the computational powerof machine learning algorithms to solve problems related to agriculture.

A classification is system to determine the condition of a plant's disease. A health class and bacterial infected class are defined by the 2 classes. In order to classify different severity levels, we further expand the classification system is needed. Groups 1 and group 2 are classified to severity ranges: 1 is being healthy and 2 infected by bacterial. Ways of collecting various features from leaf images and how specific methods of extraction result. in the classifier's different results were demonstrated in literature review. Finally, the suitable model that uses the classification model has trained to forecast the health status of a farmer's garden in real time [1] were presented as well. The type of the crop inthis study is pepper bell leaves.

## 2.2 Traditional Method in Inspecting Leaves Disease

In the past, farmer used to check their production using their eyes, based on their knowledge and their experience as shown in Figure 2.1. This way we call it traditional way which has luck of accuracy and it variant of detecting the plant disease from someone to another.

Traditional, visual analyses classify a disease depending on manifestations of a characteristic plant disease or obvious indicators of a pathogen That visual identification is carried out by professional professionals and has been the focus of extensive study and research. Consistency and precision are criteria for the consistency of visual appraisal scores. Because of the existence of comprehensive guidance and criteria used for evaluation preparation, visual measurement is now more precise and consistent. Visual prediction, though, is often subject to the experience of a person and may be influenced by temporal variation [2]. Image processing as it sometimes called, with application to satellite photography, conversion of wire-photo standards, medical imaging, videophone, character recognition, and photograph enhancement. Improving the consistency of the image was the intention ofearly image processing. It was meant to increase human beings' visual effects. The input is a low-resolution image in image processing, and the output is an image of better quality. Popular image processing includes enhancement, retrieval, decoding, and compression of images. In today's world, the agricultural land mass is more than just a feeding medium. The Indian economy is highly dependent on the productivity of agriculture. Therefore, disease identification in plants plays an important role in the field of agriculture. The use of automatic disease detection techniques is beneficial in order to detect a plant disease in the very initial stage [3].

## 2.3 Image Processing

In order to get an improved image or to obtain any valuable information from it, image processing is a way of performing some operations on an image. It is a form of signal processing in which an image is input and the image or features/features associated with that image may be output [4].

### 2.3.1 Type of Image Processing

There are two types of image processing, analog image processing and digital image processing. The processing of analogue images is extended to analogue signals and only two-dimensional signals are analyzed. Electrical signals manipulate the images. Analog signals may be intermittent or non-periodic in analogue image processing. TV images, videos, drawings, and medical images are examples of analogue images. Digital image processing, a matrix of small pixels and components, is applied to digital images. There are a variety of tools and algorithms to modify the images that are applied to perform improvements. Digital image processing is one ofthe fastest growing industries that affects the lives of everybody. Color processing, image recognition, video processing are examples of visual images Table 2.1 below shows difference between analog Image processing and digital Image processing [5].

**Table 2.1 Difference Between Analog Image**

| Analog Image Processing | Digital Image Processing |
|---|---|
| Analog image processing is extended to analogue signals and only two-dimensional signals a reanalyzed. | The processing of digital images is applied to digital signals that analyze and manipulate the images. |
| The analogue signal is a time-varying signal in order to vary the images produced under analogue image processing. | It improves the digital quality of the image and it is perfect for the distribution of intensity. |
| Analog image processing is as lower and costlier process. | Is a cheaper and quicker process for storing and retrieving images. |

### 2.3.2 Image Processing in Agriculture Field

In fact, image processing has been very helpful in many fields, one of them is agriculture field where farmer can detect the disease on pepper in early time. Image processing consist of capturing, analysis of two-dimensional image of the pepper. In order to understand the using of image processing in agriculture, some tools have to be applied like camera which represent the human eyes and machine learning to classify the

disease this technology considered as consistent, safe, high efficiency in operation and high stability.

### 2.3.3 Sign And Symptoms of Plant Disease

According to Michigan State University Extension study about the plant disease. Most often plant disease around 85 percent are caused byfungal or fungal like organisms. However other serious disease caused by viral bacterial organisms. Plant disease is also caused by such nematodes. Any plant diseases are known as abiotic or non-infectious diseases, including disruption caused by air pollution, dietary deficiency or toxicity, and developing under less than optimum conditions.

Physical confirmation of the pathogen is a sign of plant disease. Fungal fruiting bodies, for example, are an indicator of disease. The thick, liquid exudate consists mainly of bacteria and is a result of the disease, while the canker itself consists of and is a symptomof plant tissue.

The noticeable influence of disease on the plant is a sign of plant disease. If it reacts to the pathogen, symptoms can include a detectable change in the color, shape or feature of the plant. A common symptom of wilting, caused by fungal plant infections, is leaf wilting. Orange, necrotic lesions surrounded by a bright yellow halo at the leaf margin or within the leaf on bean plants are typical bacterial blight symptoms.
The Table2.2 below shows some examples of symptoms and sign of fungal, bacterial, viral plant disease:

*Table 2.2 Shows* **Symptoms** ***and Sign of Fungal, Bacterial, Viral Plant Disease***

| Disease | Sign | Symptoms |
|---|---|---|
| Fungal | • Leaf rust (common leaf rusting corn)<br>• Stem rust (wheat stem rust)<br>• Sclerotinia (white mold)<br>• Powdery mildew | • Birds-eye spot on berries (anthracnose)<br>• Damping off of seedlings (phytophthora)<br>• Leaf spot (Septoria brown spot)<br>• Chlorosis (yellowing of leaves) |
| Bacterial | • Bacterial ooze Water-soake d<br>• lesions Bacterial<br>• streaming water from a custom | • Leaf spot with yellow halo<br>• Fruit spot<br>• Canker<br>• Crown gall<br>• Shepperd's crook stem ends on woody plants |
| Viral | • None – the viruses themselves can't be seen | • Mosaic leaf pattern<br>• Crinkled leaves<br>• Yellowed leaves<br>• Plant stunting |

Figure2.2, shows example stripe rust pustules Ona winter wheat leaf is a symptom and dark red kidney bean leaf showing bacterial leaf spot symptom [6][7].

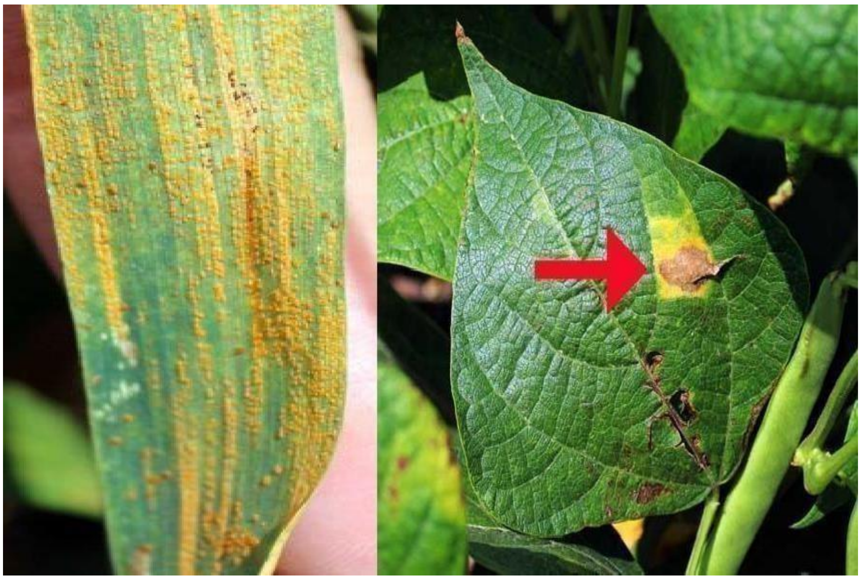


Figure 2.2 Stripe Rust and Dark Red Kidney Bean

### 2.3.4 Common Plant Disease

Disease taking energy from plant so the less we take care of our culture the more the disease will spread. Here is some of the common disease that we usually see it on plant[8].

Table 2.3 Shows the Common Plant Disease

| | | | |
|---|---|---|---|
| <br>Anthracnose<br>Infected plant, water-soakedllesionstem. | <br>Blossom End Rot<br>Spot on the end bottom of ripped tomato develop a large sunken. | <br>Corn Smut<br>They can release thousands of spores as they rupture, can grow up to 5 inch in diameter. | <br>Early Blight<br>Brown spots found in lower leaf. |
| <br>Late Blight<br>Beginning the edge of the leaf, it can be found on the tomato and potato. | <br>Potato Scab<br>Looks dark, pithy and warty. | <br>Apple Scab<br>Sunken spots on the leaf and the fruit and in the center have velvety spot | <br>Brown Rot<br>Almonds, apricots, cherries, peaches and plums can get effected by this kind of disease. |
| <br>Cedar Apple Rust<br>On the upper surface and it loos pale yellow | <br>Crown Galli<br>woody shrubs like grapes, raspberries and roses. | <br>Damping Off<br>Due to old seeds and wet soil. | <br>Fire Blight<br>Plant leaf the infected by bacterial. |

| | | | |
|---|---|---|---|
| <br>Fusarium Wilt<br>Yellowing and wilting of the lower leaf. | <br>Downy Mildew<br>Appear on upper old leaf surface, yellow to white. | <br>Gray Mold<br>Known as gray soft, mushy spot-on leave | <br>Bacterial Canker<br>It effects some kind of stone fruit. |
| <br>Black Knot<br>Attack both of fruit and ornamental of cherry, chokecherry trees. | <br>Club Root<br>It appears on the root of cabbage family. | <br>Leaf Curl<br>Usually it affects blossom, fruit, leaves. | <br>Leaf Spot<br>Dark spots on the leaves, sometimes with yellow hole |
| <br>Garden Pest Control<br>Prevent most plant disease by offering least toxic and natural organic | <br>Mosaic Virus<br>Appears as light green or yellowish color on the leaves surface | <br>Powdery Mildew<br>Dusty white to gray coating on the leaf surface. | <br>Rust<br>Orange spore can be foundon lower leaf. |

## 2.4 Architecture Convolution Neural Network

Convolution layers are parallel feature maps, generated by sliding over an input image of different kernels (feature detector) and projecting the element-wise dot as the feature maps. This method of slipping is known as stride Zs. As compared to the input image, this kernel bank is smaller in size and is overlapped on the source images, prompting criteria such as exchanging weight and bias between both the neighboring pixel of the image and manipulating the dimensions of attribute maps. However, using the limited size of kernels also results in imperfect overlays and restricts the learning algorithm's strength. Therefore, the Zp process of zero padding is typically applied to control the size of the input image. Zero padding can independently govern the dimensions of function maps and kernels by symmetrically applying zero to the input. By following Eq, the dimension of the convolution layer can be computed. Equation 2.1:

$$\boldsymbol{D}_{imc}(\boldsymbol{H}_1, \boldsymbol{W}_1, \boldsymbol{D}_1) = \frac{H+2Z_P-k_1}{Z_s} + \mathbf{1}, \frac{W+2Z_P-k_2}{Z_s} + \mathbf{1}, \boldsymbol{K}_D \qquad \text{Equation 2.1}$$

(H,W,C) the fixed size input image. k1 Width of convolution kernel, k2 Height of convolution kernel, K_D Number of kernels.

After nonlinear activation, pooling operations are carried out where the pooling layers help to reduce the number of data points and prevent overfitting. It also serves as a smoothing process that can remove unnecessary noise from the process. The Max pooling operation is most frequently used.

$$D_{imp}(H_2, W_2, D_2) = \frac{H_1-k}{Z_s} + 1, \frac{W_1-k}{Z_s} + 1, D_n$$ Equation 2.2

The pixels of the pooling layers are extended to a single column vector after the pooling layers. For grouping, these vectorized and concatenated data points are fed into dense layers, known as completely linked layers. The role of dense layers which are completely linked is similar to the Deep Neural Network. Figure (2.3) provides the architecture of ConvNets. In image classification problems, this type of constraint architecture can competently exceed the classical machine learning algorithms [9].

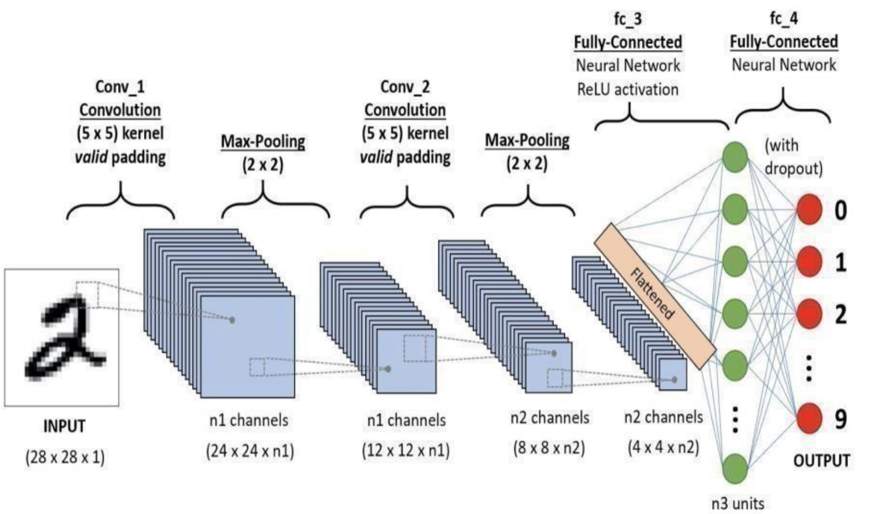



Figure 2.3 Architecture Convolution Neural Network (ConvNets)

### 2.4.1 Convolution Neural Network in Image Processing (CNN)

### 2.4.2

Based on images, of leaves detection. In the meantime, a creative CNN architecture built particularly for the fascinating features of the pepper leaves appearance is knowledgeable. The proposed CNN method takes multiparty images as data, generating a three-fold loss (similarity loss). The fine-grained feature representations from the identification of leaves can be efficiently learned through our proposed methodology through the combined optimization of classification accuracy and lack of resemblance [10]. The suggested methodology has three benefits. First, it utilizes a training method to automatically extract multi-scale image characteristics that incorporate the global and local characteristics of the leaves image. The graph obtained from these fine-grained features provides a function with another fresh input of pepper leaves images to assess the disease levels of plants. Next, as the examples of defective peppers are also used in the studies, the exploratory findings further agree that our methodology may be linked to the fine-grained grading of pepper leaves at different levels of disease regardless of whether or not the pepper bears leave. Finally, in comparison to certain current approaches, our predictor does not combine data from physiochemical or biological shifts during the pepper growing season [10].

### 2.4.3 AlexNet Network

### 2.4.4

The AlexNet was much greater than previous CNNs for image processing activities. For 650,000 neurons, this is 60 million and it requires between five and six days to practice on both GTX 580 3 GB GPUs. These days, even on very large datasets, much more sophisticated CNNs can work very well on faster GPUs. Figure (2.4). The

Alexnet comprises eight layers. The first five are convolution layers and 3 totally linked layers and there are pooling [11], dropout and trigger layers between them, the AlexNet input is the picture of 3-channel RGB in size 256*256. This suggests that every image and every test image in the training set have to be 256*256.

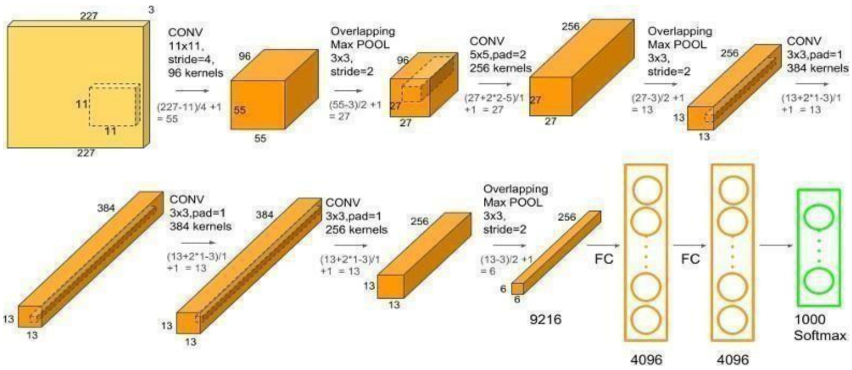



Figure 2.4 Alexnet Architecture

### 2.4.5 Color Histogram

For a color histogram, the color spectrum of the image is represented. Like a digital image, a color histogram indicates the number of pixels. You may create some sort of input image in a color histogram. The resolution histogram should instead be used for grey scale images. By discretizing the colors in the image into a number of bins, an image histogram is first generated. The Red-Blue color space histogram will generate the first normalized color pixel values. The two-dimensional histogram of Red Blue chromaticity is split into four bins. Figure 2.5, Figure 2.6 and Figure 2.7 shows how RGB been compact and spared. A compact description of the spread of the data in an image is given in the histogram. With the color histogram, the difficulties of knowing the object of an uncertain location and rotation within the same picture which match.

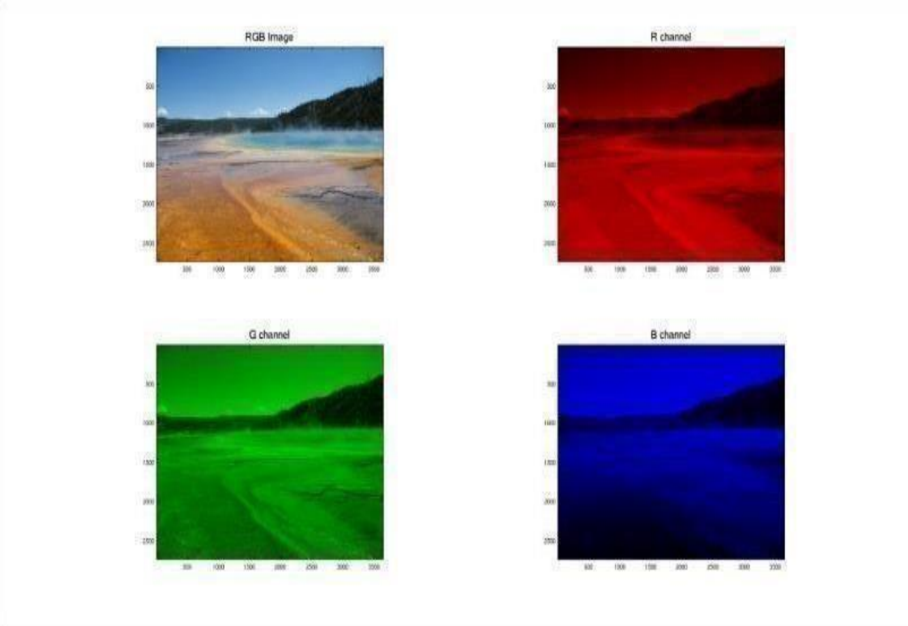



Figure 2.5 Original Image to RGB Channel

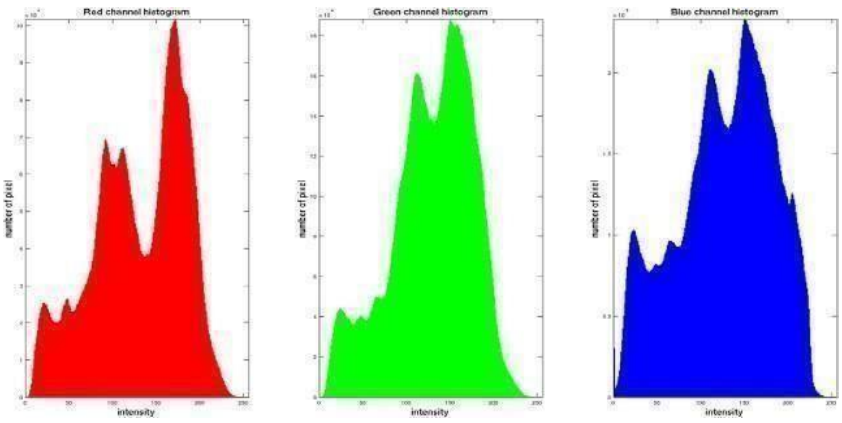



Figure 2.6 Original Image Associated of RGB Channel

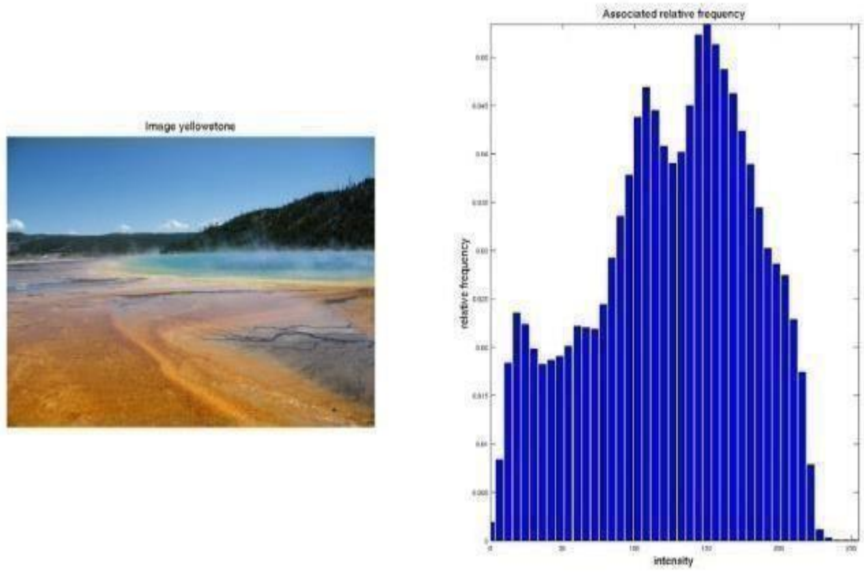



Figure 2.7 Original Image and Combination of Histogram Result

### 2.4.6 K-Mean Clustering

One of the image processing techniques is K-Mean. The vector quantization of signal processing, which is popular for cluster analysis in data mining, is the K-mean clustering technique. The intent of K-mean clustering will be to walls to make a declaration in k clusters that the observation belongs to the cluster with the mean, serving as the cluster prototype. It is fairly easy to perform and apply K-mean clustering to large data sets. In cluster analysis, it is possible to apply the k mean algorithm to partition the input data set into k clusters. By K-mean clustering, feature extraction and image classification can be implemented. The basic approach is to use the input data sets to prepare the k-mean clustering model. This paper would take the principles and the formula. The mean point formula in K-mean is:

$$\boldsymbol{V} = \sum_{\boldsymbol{i}=\boldsymbol{1}}^{k} \sum_{x_j \in S_i} (\boldsymbol{x_j} - \boldsymbol{u_i})^{\boldsymbol{2}} \qquad \text{Equation 2.3}$$

K= number of clusters, mi=mean point

For color images, the algorithm needs data. K implies the following algorithms are [12]:

1- The intensity values will be determined by the K-mean.

2- To initialize the clusters, variable levels for K.

3- Repeat phases 4 and 5 till the outcome of the cluster has changed.

4- Cluster and export each input feature value from the K-means values.

### 2.4.7 Masks

The notion of masking is that it does spectral analysis. The dimensional filtering is done on the image directly. The aim of filtering by using the mask will be to transfer the filter mask from point to point in an image. The pre-defined picture determines the filter for the answer. All the values of the filter are predefined and then become normal. These masks are used for blurring and noise reduction and also for the identification and sharpness of the corners or edges. In pre-processing steps, blurring is applied, such as the identification of minor information in an image prior to removing large objects. The box filter and total average filter are available masks for blurring. It eliminates image edge material in the blurring process and helps make the transformations among different intensities of pixels as clear as possible. Noise reduction is also an advantage to the blurring process. Figure 2.8 shows a random image with mask applied, marks may be used in an image for edge detection and to improve the sharpness of an image.

Figure 2.8 Original Image and Masks Image

### 2.4.8 Support Vector Machine (SVM)

Support vector machine (SVM) algorithm used for classification and regression analysis is the Help vector machine. The binary linear classifier and nonlinear classifier can be executed by the help vector machine. The help vector machine will decide to define the new class with any data that belongs to one of two classes [11]. A lot of real-life problems can be addressed by the help vector machine. Figure 2.9 shows a graph of how SVM classifier works with the data.

1- SVM is useful for the categorization of text and hypertext.
2- The SVM is capable of classifying pictures. The actual picture classification scheme is SVM.
3- A hand-written character can be remembered by the SVM.
4- In biological research and other fields, SVM can also be used.

Linear classification formula for SVM

Equation 2.4

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\vec{w}\cdot\vec{x_i} - b))\right] + \lambda \| \vec{v}$$

Non-linear classification formula for SVM

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i - b))\right] + \lambda |$$

Equation 2.5

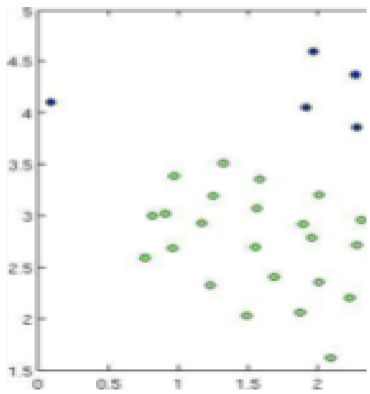



Figure 2.9 SVM Classifier

### 2.4.9 Neural Networks (NN)

This methodology is focused on mathematical equations that are essentially depicted as graphs. Neural networks are a group of algorithms used in n-layers, as seen in Figure 2.10, to detect and identify data input. Although the probabilities of state transitions, relation strengths and functions are used for neural networks. In order to fit patterns such as the Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Deep Neural Network, neural networks use various algorithms (DNN)[13].

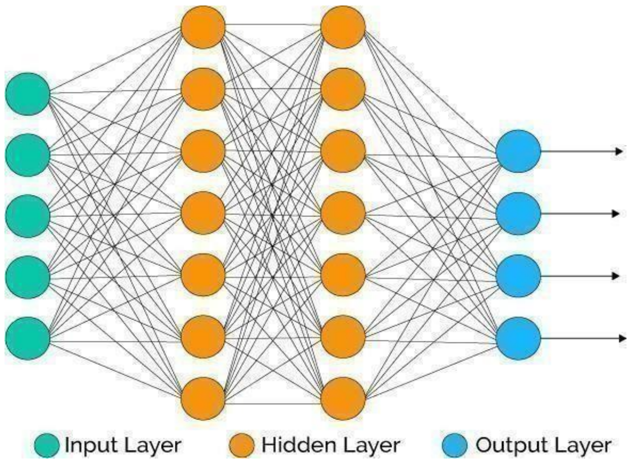



Figure 2.10 Neural Networks

### 2.4.10 Python

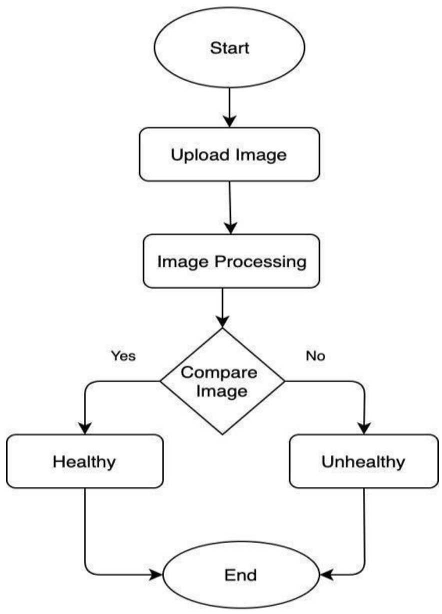
Pythonis a programming language for development purposes it became really popular rapidly, mostly because of its flexibility and readability of code. It allows the programmer to express his thoughts in less code lines despite decreasing any readability. Python is slower relative to many other languages such as C/C++. But yet another significant advantage of Python is that with C/C++ it can be quickly expanded. This function allows us to write C/C++ high computational codes and construct a Python framework for so that as Python modules we can use these wrappers. This is how OpenCV Python operates. It is a Python wrapper over the initial version in C++. NumPy's assistance even allows the job simpler [14]. NumPy, for numerical operations, is a highly optimized library. This includes a MATLAB style syntax. Both structures of the OpenCV array are translated to-and-from NumPy arrays. So you can merge any operations you can do in NumPy with OpenCV, which raises the number of ammunition in your arsenal. In addition, some other libraries can be used for this, such as SciPy, Actual lesson that supports NumPy. OpenCV-Python is also an appropriate method for quick prototyping of computer vision applications [14].

### 2.4.11 Google Collab

Collaboratory, or simply Collab, is a Google Research tool that allows developers to create and run Python code directly from their browser. For deep learning tasks, Google Collab is a great tool. It's a hosted Jupiter notebook that doesn't require installation and includes a great free version that provides you free access to Google processing resources like GPUs and TPUs.

## 2.5 Proposed Method

Firstly, we will create CNN and SVM models so, if the user uploads the defected image on the system, then the system will search image of related information in the system database that been trained in Google Collab. After uploading the defected image, the user has to upload a healthy image of paper leaf and trained them as well. finally, when a new image got into the system, the system will compare the image with the datasetthat is already uploaded. The flowchart Figure
2.11 shows the process of the research.



## 2.6 Summary

The Table2.4 down shows the summary of previous research, all these researches are somehow related to detecting disease of plant using image processing and shows different techniques. Those researches have been made between 2013 which is about Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features and 2019 however, there are three researches that been published which are about several visualization techniques to detect and classify the symptoms of plant diseases, survey paper on plant leaf diseases detection techniques used in machine learning and survey of techniques for disorder detection in tomato. Some of these researches are more related to this paper in way that is using similar techniques of an SVM or CNN models or disease detection of a fruit or plant leaves like the research on 2016 by Tyler Karrels experiment about banana entire plant, leaves, pseudo stem, fruit bunch and corm with CNN those researches were helpful to make this paper.

Table 2.4 **Summary of Previous Research**

| S.No. | Year | Author(s) | Title | Methodology | Limitation(s) |
|---|---|---|---|---|---|
| 1 | 2025 | Thota Akshaya, Ganji Shruthi, Pavan Sai Mohan Arige, Medikonda Asha Kiran, Manyam Thaile, Pravin Ramesh Gundalwar | Bell Pepper Plant Disease Detection using Deep Learning Models | Present 1-5 classes of plant disease while, 1 is healthy and 5 is being severely disease plant. Using smartphone based on system classification to real time prediction. This method work by uploading picture by the user. | The data we used consists of 7,386 images of leaves of cassava plants. Used to detect major disease in Africa which is CBSD,CMD,CBB,CGM |
| 2 | 2025 | Jose B. Lazaro, John Rafael L. Jocson | Image Classification Model for Data Sentry of Plant Diseases Using Convolutional Neural Networks | Identify the disease by expert farmer, they have the knowledge to treat each kind of disease manually. | Input image data by manually croppingth edges of the leaf to avoid for unnecessary background. The data usedconsist of 1821images with accuracy score of 96%to 97% of training. |
| 3 | 2016 | Mitkal, P. Pawar, M. Nagane, P. Bhosale, M. Padwal, and P. Nagane | Leaf Disease Detection and Prevention Using Image Processing using Matlab | Leaf spots can be indicative of crop diseases, where leaf batches (spots) are usually examined and subjected to expert opinion. In our proposed system, we are going to develop an integrated image processing system to help | computed for the useful segments, finally the extracted features are passed through the classifier. |

| | | | | | |
|---|---|---|---|---|---|
| 4 | 2025 | Yashodha G, Rajakumar B, Krishnamoorthy M, Surendran R | Efficient Plant Disease Detection using K-Means clusterin g and DenseNe t -based Classificat ion | -we lack high quality, high frequency, long -term data with which to validate and improve leaf tempera- true simulation models. | -Sensor noise.<br>-Sensitivity analysis<br>-Site descriptions<br>-long-term study thatrequire much equipment |
| 5 | 2024 | Taifa Ayoub Mir, Deepak Banerjee, Rahul Chauhan, Hemant Singh Pokhariya | Augment ing Bell Pepper Disease Detecti o n: A Deep Learning Approac h Incorpor ating Convolut ional Neural Network s and Support Vector Machine s | Wheat crop used for validation purposes.<br><br>-seven different devices: iPad, iPhone4, Dell-tablet, Samsung Galaxy Note, Windows Phone, Samsung S3, iPhone5.<br>-A total number of 3637 images of wheat diseases have beentaken under natural conditions during 2014 and 2015.<br><br>-In order to design and validate an algorithm capable of identify crop diseases at their early stages, all dataset images were accurately segmented by expert technicians. | The analysis was done using 7 mobiles devices and more than 3500 images captured in two pilot sites in Spain and Germany during 2014, 2015 and 2016.<br><br>-Other sensor devices for direct or indirect color variation detection could provide useful information as severity and spreading in the plant.<br><br>-a phone application has been used to analysis the image that has been taken during the study. |

| | | | | | |
|---|---|---|---|---|---|
| 6 | 2024 | Anitha Julian, Jaiganesh V, Thirumurugan O T, Isreal Moses B | PepperShield : Disease Detection with Hyper Tuned Precision | -BRBFNN has been used to classify the type of disease on the leave.<br><br>-Image segmentation taken in process based on common priorates. Traditional method. Soft computing method | Short term study based on image capture and segmentation.<br><br>Several types of disease have been identify such as (common rust, late blight, cedar apple, rest, leaf curl, spot, early blighting |
| 7 | 2024 | Pandula Pallewatta, Thilina Halloluwa, Kasun Karunanayaka, Gihan Seneviratne, Samantha Mathara Arachchithi, | BellCrop – A Bell Pepper Leaf Dataset for Disease Classification and Yield Enhancement using Machine Learning | Plant diseases have turned into a dilemma as it can cause significant reduction in both quality and quantity of agricultural products. The proposed algorithm's efficiency can successfully detect and classify the examined diseases with an accuracy of 94%. Experimental results on a database of about 500 plant leaves confirm the robustness of the proposed approach. | The developed processing scheme consists of four main steps, first a color transformation structure for the input RGB image is created, then the green pixels are masked and removed using specific threshold value followed by segmentation process, the texture statistics are computed for the useful segments, finally the extracted features are passed through the classifier. |

| | | | | | |
|---|---|---|---|---|---|
| 8 | 2024 | Rahul Singh, Neha Sharma, Deepak Upadhyay, Swati Devliyal, Akira Singh, Retinderdeep Singh | An Efficient CNN-Based System for Classifying Pepper Bell Plant Conditions | -The data use in this paper consists of five major banana diseases along with their respective healthy classes; dried/old age leaves and banana corm weevil Cosmopolites sordidus damage symptom classes.<br><br>-Six different models (entire plant, leaves, pseudo stem, fruit bunch, cut fruits and corm) and 18 different classes. | -Experiment about banana (entire plant, leaves pseudo stem, fruit brunch and corm)with CNN.<br><br>-Smartphone based on AI system has been used to detect banana disease. |
| 9 | 2019 | Ankita Suryavanshi, Vinay Kukreja, Ayush Dogra, | Capsicum Plant Advanced Disease Detection Using CNN-SVM Fusion for Precision Crop Protection | Several visualization techniques to detect and classify the symptoms of plant diseases. | -CNN was used for the classification of diseases in maize plants and histogram techniques to show the significance of the model. |

| | | | | | |
|---|---|---|---|---|---|
| 10 | 2023 | Nitin Thapliyal, Manisha Aeri, Abhishek Satyarthi, Vinay Kukreja , Rishabh Sharma | Diagnosis of Plant Infection for Optimised Authomatic Health Monitoring System Using Vision Transformer Models | -Many Machine Learning (ML) models have been employed for the detection and classification of plant diseases. | -AlexNet, and Google Net were ResNet for the implement leaf identifying tomato diseases. |
| 11 | 2024 | Anil R. Ghodekar, Nailya Sultanova, Manoj Jayabalan, Jamila Mustafina | Tomato Plant Leaf Disease Classificatio n Using Deep Learning | Leaf spots can be indicative of crop diseases, where leaf batches (spots) are usually examined and subjected to expert opinion. In our proposed system, we are going to develop an integrated image processing system to help automated inspection of these leaf batches and helps identify the disease type. Feature extraction is the third stage, which deals with three features, namely color size and shape of spot. The fourth stage is classification, which comprises back propagation based neural networks. | Convention al Expert systems mainly those which used to diagnose the disease in agriculture domain depends only on textual input. Usually, abnormaliti es for a given crop are manifested as symptoms on various plant parts. |

| | | | | | |
|---|---|---|---|---|---|
| 12 | 2023 | Gurjot Kaur, Neha Sharma, Rahul Chauhan, Prabhdeep Singh, Rupesh Gupta | An Automated Approach for Detection and Classification of Plant Diseases | 1) Reading the RGB image. 2)Histogram equalization. Resizing the image. 3)Create the color transformation structure. K-means c 6) lust ering algorithm in an image. 7) Convert the affected leaf (clusters) from RGB to HSI. create a spatial Gray Level Dependence Matrices (SGDM's) for S and H. 9) Calculating the features by calling the gray- level cooccurrence matrix (GLCM) parameters. 10) Recognition using Artificial Neural Network (ANN). | The feature extraction is done by using the color Co-occurrence Method (CCM method). This method extracts both the texture and color of an image, finally to arrive at unique features which represent that image. -K-Map, RGB and HSI are the method that used in this paper for image processing. |

| | | | | | |
|---|---|---|---|---|---|
| 13 | 2023 | Lamees Rababa, Naseem Ali, Rana Alessa, Ahmad Alzu'bi | Pepper Leaf Diagnosis Using Deep-Net with Low-Dimensional Image Classification | - RGB, SVM, DSS,HSF,ROI and SIFT with machine learning to develop algorithms.<br>-Data around 120images were collected. | -Focused on image processing techniques to detect disease of soybean plants.<br>-using smartphone cameras with resolution higher than 2 mega pixels. |
| 14 | 2023 | Aisha Ahmed AlArfaj, Abdulaziz Altamimi, Turki Aljrees, Shakila Basheer, Muhammad Umer, Md. Abdus Samad, Shtwai Alsubai, Imran Ashraf | Multi-Step Preprocessing With UNet Segmentation and Transfer Learning Model for Pepper Bell Leaf Disease Detection | -using developing the higher breed seeds and plants.<br>-type of disease as an experimental model for detection of Brown Spot, Downy mildew, Sugarcane Mosaic, Red stripe, red rot, Downy Fungal. | -This research is done by Matlab software, digital image processing and RGB to detect disease onthe leaves. |
| 15 | 2023 | Lianne Joyce D. Leynes, Ivan Charles L. Razon, Meo Vincent C. Caya | Development of Identification System for Cactaceae Plants using YOLO Algorithm | In this study digital image processing has been used to detect various plant disease by RGB and grayscale technique. | compared two types of images such as Grayscale, RGB |

| | | | | | |
|---|---|---|---|---|---|
| 16 | 2023 | Julius Tan, Bryant Christoper Sutanto, Darren Marvelim, Galih Dea Pratama, Silviya Hasana | Chili Plant Health Status Classificati on Using Random Forest | This paper is about feature extraction, a preliminary step for detection of plant disorders and proposes a novel improvement based upon findings to improve classifier's Accuracy on tomato. | Used Remote sensing for feature extraction and ANN is used extensively for classification purpose with some technique that been used like, HSV,RGB,HOG , binary image and compare between image processing technique that used to detect plant disease. |
| 17 | 2023 | Fatema Tuj Johora Faria, Mukaffi Bin Moin, Ahmed Al Wase, Md. Rabius Sani, Khan Md Hasib, Mohammad Shafiul Alam | Classificatio n of Potato Disease with Digital Image Processing Technique: A Hybrid Deep Learning Framework | Point out a comparative study on different types of plant leaf diseases and all the techniques of machine learning which had been used to detect plant leaf disease. | k-Nearest Neighbor Classifier, Probabilistic Neural Network, Genetic Algorithm, Support Vector Machine, and Principal Component Analysis, Artificial neural framework, Fuzzy reason. This paper gives a survey of different portrayal frameworks used for plant leaf disease order. |
| 18 | 2025 | Muhammad Shoaib, Abolghasem Sadeghi-Niarak i, Farman Ali, Irfan Hussain, Shah Khalid | Leveraging Deep Learning for Plant Disease and Pest Detection: A Comprehensi ve Review and Future Directions | Provides a full review of DL architectures (CNNs, GANs, Transformers) used for plant disease and pest detection. | Review-based study; no new dataset or model implementation. |
| 19 | 2025 | Abhishek Upadhyay, Narendra Singh Chandel, Krishna Pratap Singh, Subir K. | Deep Learning and Computer Vision in Plant Disease Detection: A Comprehensi ve Review | Surveys CNN, MobileNet, EfficientNet, and hybrid CV models for precision agriculture. | Lacks experimental validation; focuses on literature synthesis. Limited coverage of real-time deployment challenges. |

| | | | | | |
|---|---|---|---|---|---|
| 20 | 2024 | Shaohua Wang, Dachuan Xu, Haojian Liang, Yongqing Bai, Xiao Li | Advances in Deep Learning Applications for Plant Disease and Pest Detection | Uses remote sensing + CNN/ViT models for large-scale crop monitoring. | Remote sensing datasets are expensive and not accessible for small farms. High computational cost for training. |
| 21 | 2025 | Ashurov A.Y., Al-Gaashani M.S.A.M., Samee N.A., Alkanhel R., Atteia G., Abdallah H.A. | Enhancing Plant Disease Detection Through Deep Learning: A Depthwise CNN with SE Integration | Proposes a lightweight depthwise CNN with squeeze-and-excit ation | Model tested on limited datasets; generalization to diverse crops not fully validated. |
| 22 | 2025 | R. Nilesh Kumar, Ranbir Singh, Sachin Gupta, S. K. Singh | Artificial Intelligence and Machine Learning in Plant Disease Detection | Overview of AI/ML techniques including SVM, RF, CNNs, and hybrid models. | High-level overview; lacks quantitative evaluation. |

As Based on research, of leaves detection. In the meantime, a creative CNN architecture built particularly for the fascinating features of the pepper leaves appearance is knowledgeable. The proposed CNN method takes multiparty images as data, generating a three-fold loss (similarity loss). The fine-grained feature representations from the identification of leaves can be efficiently learned through our proposed of classification accuracy.

**Research Gaps:**

Although Convolutional Neural Networks (CNNs) have demonstrated strong performance in plant disease detection across various crops, several gaps remain in their application to pepper leaf disease classification. Existing studies primarily focus on general plant datasets or other crops such as cassava, wheat, and tomato, leaving a lack of CNN architectures specifically optimized for the visual characteristics of bell pepper leaves.

removal or leaf segmentation, limiting their practicality in real-world farm environments where lighting, angles, and noise vary significantly. Current research also tends to emphasize broad disease classification rather than fine-grained detection of early-stage symptoms, which is critical for timely intervention. Additionally, most CNN models are trained on limited or imbalanced datasets, reducing their robustness and generalizability when exposed to diverse field images uploaded by end-users. Finally, despite advancements in deep learning, few studies integrate CNNs into a user-driven, real-time system capable of comparing newly uploaded images with healthy reference samples to provide immediate and reliable health assessments for pepper leaves. These gaps highlight the need for a more specialized, automated, and field-ready CNN-based framework tailored to pepper disease detection.

# CHAPTER 3: METHODOLOGY

## 3.1 Introduction

This chapter will explain the process of plant leaf disease detection which includes the chosen method, project flow supported by some graphs and charts, and finally the used software with its tools and environment setup. The CNN framework which is used in this project is python OpenCV in Google Collab with image processing models as it has been explained in the previous chapter.

## 3.2 Flowchart

In order to achieve the purpose of this research, Figure3.1 down describe the flow process to detect the disease on the pepper leaves where first is image acquisition, taking or capturing the image that required to study it then, image preprocessing where the image will be isolated from the surrounded environment, however image segmentation is the most significant stage where essential technique will be applied on the pepper leaf,

then the feature of the image will be extracted finally, the image will be classified into healthy leaf or unhealthy leaf.

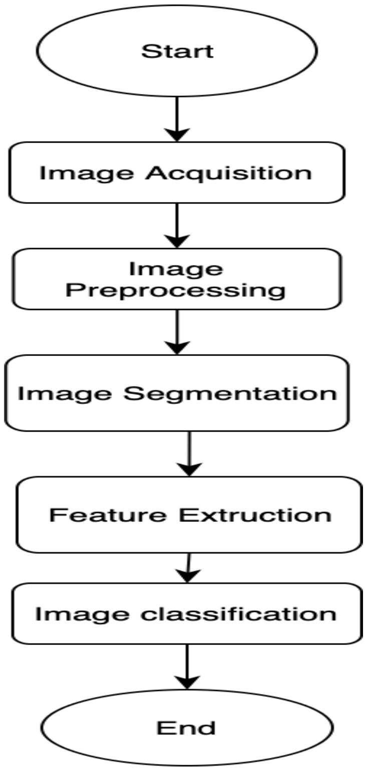



Figure 3.1 Flowchart

### 3.2.1 Image Acquisition

This is the first stage, where we use smartphone camera to capture the image of pepper leaf through sensor called image acquisition. the image that has been taken is in form of RGB color model (Red, Green, Blue) with size of (256 x 256 pixels ) .Figure 3.2 shows two samples of bacterial infected pepper bell leaf and healthy pepper bell leaf:

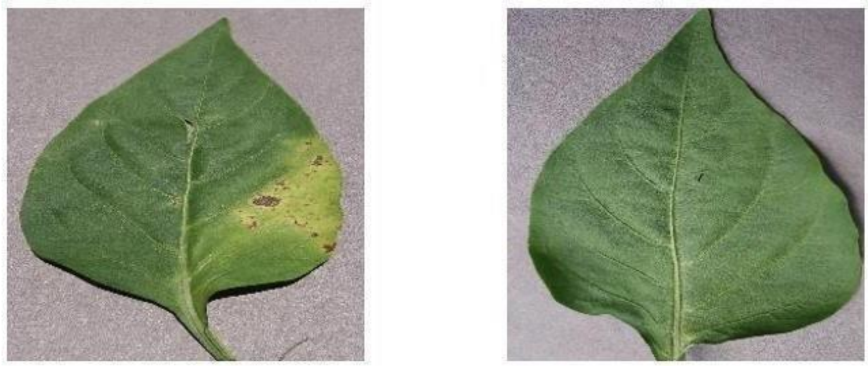


Figure 3.2 Samples of Bacterial Infected and Healthy Pepper Bell Leaves.

### 3.2.2 Image Preprocessing

In this stage we need to apply some technique on the image that has been taken. To improve and extract the data that we need from the picture and clear the unwanted data. We have some technique for image preprocessing intensity adjustment, histogram equalization, all these method needs to be applied to extract the leaf out of the soil and surrounded environment . The Figure3.3 below shows the image acquisition which is capturing the object and second stage which extracting the plant leaf and the data needed.

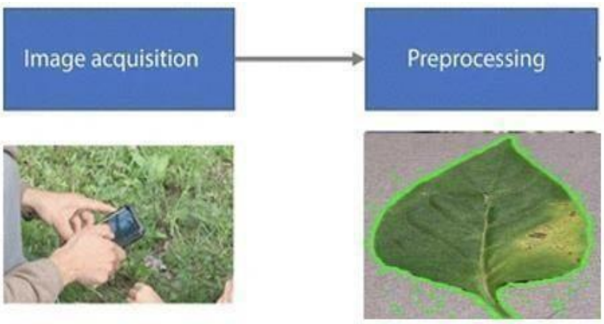



Figure 3.3 Image Acquisition and Image Preprocessing.

### 3.2.3 Image Segmentation

This is the most significant stage of the image proceeding like, image retrieval and small surveillance. In this part where we can identify the pepper bell leaves from thedata that has been extracted on the second stage. And it is working based on process of isolating an image in to more significant representations by understanding and identify the visual entity recognition as we can see on Figure3.4 , where it shows pepper bell leaves with segmentation technique of SI, adaptive mean and Otsu's methods been applied on pepper bell leaf that infected by bacterial however, these technique are poorly preformed where slight illumination changes have occurred .OTSU method worked nicely and contained many data, but it was also unable to handle the improvements in pepper bell leaf effect and illumination.

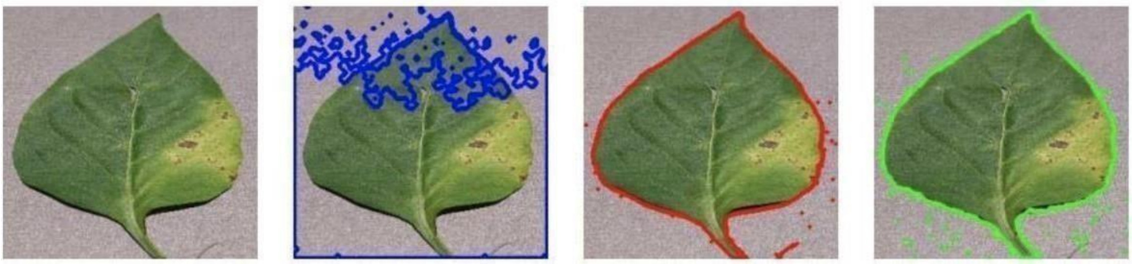


Figure 3.4 Segmentations Using Of SI, Adaptive Mean and Otsu's Methods

Figure 3.5: shows another technique of segmentation applied to pepper leaf infected by bacterial where it illuminate the part that infected on the pepper bell leaf.

Figure 3.5 Algorithm Implementation on Bacterial Infected of Pepper Bell Leaf.

Segmentation has many techniques that we need to use to recognize the leaf like:

- Clustering
- Watershed
- Region Based
- Edge Based
- Threshold

**Clustering Segmentation** it clusters some pixel in the image where the image first converted to histogram and then cluster it one of the cluster phases is called k-mean where the image being cluster in texture image using some algorithm.

**Watershed based Segmentation,** in this technique the pixel which has more gradient is represented as boundary which is broken so, it thinks about the gradient of image as topographic surface.

**Region Based** the primary purpose of using region-based technique is to split the image into different type of regions then, pixels which from the same type identify and group together into same type of regions. Region Growing, Region Splitting and Region Merging are the main type in this technique.

**Edge Detection** indeed for detecting the edge of the leaf based on the leaf's shape Sobel, Canny, Laplacian, fuzzy and different intensity at the prodder of the leaves help us identify our object.

**Threshold** this technique considers as the simplest method among the others where it based on threshold the value that been calculated by convert the image to binary. Thresholding methods Globally and Locally have methods like Adaptive local thresholding be global thresholding.

Table3.1 shows some of these techniques needs to be applied and identify the leaf from the picture that been taken and connecting the previous stage to the second [15].

Table 3.1 Shows Comparison Between Different Techniques

| **Segmentation Technique** | **Advantages** | **Disadvantages** |
|---|---|---|
| Clustering | Eliminate spots and noise. | Difficult to predict k value |

| | | |
|---|---|---|
| Watershed | Continuously detecting boundaries and with suitable result. | Complex to calculate the gradients. |
| Regio n Based | Immunity to noise, good when we have similarity criteria. | Expensive method regarding of time and memory required. |
| Edge Based | Good for better contrast among images. | Not suitable if the image has a lot of edges. |
| Threshold | Fast and easy to use. | Special details are not considered, has high sensitivity and noise. |

Connected to the python software which will be illustrated later in this project. It will show proper accuracy using some of the technique that listed above. Nevertheless, with help of other stages pepper leaves can be detected.

### 3.2.4 Feature Extraction

In this stage we consider the features that happen to the pepper bell leaves, when the pepper bell leaves get effected by bacterial, batches (spot) will appear on the leaf. Batches (spots) on the crop leaf consider important units indicating the existence disease and regarded as indicator of crops disease. In order to classify the samples of disease classification we need to extract the features from our data. Figure (3.6) shows citrus greening on a pepper bell leaves which obviously looks infected:

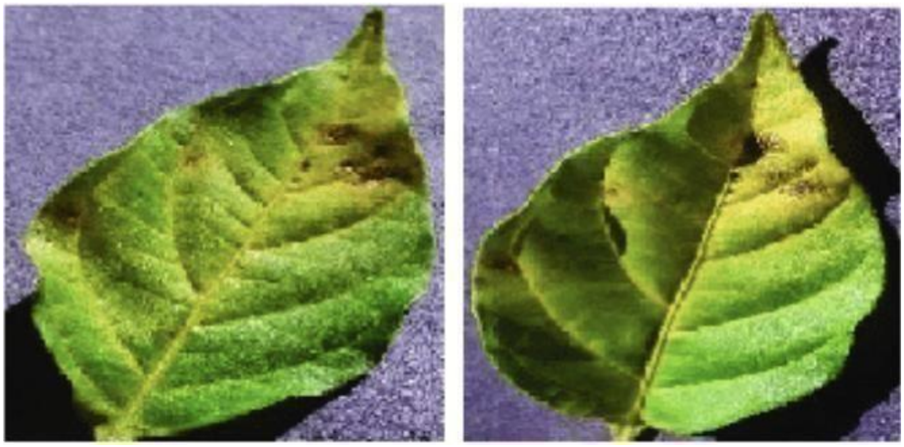


Figure 3.6 Samples of Citrus Greening of Pepper Bell Leaves

Figure 3.7, Also shows a pepper bell leaves that infected with spot disease:

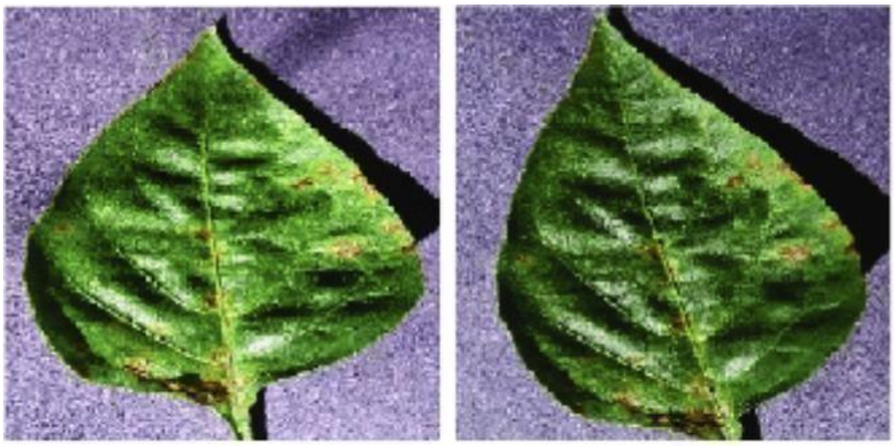


Figure 3.7 Samples of a Pepper Bell Leaves Infected By Spot Disease.

A variety of spot characteristics are studied for the recognition and identification of the multiple leaves of the disease. Spot features are extracted using the required form of image processing from the image. For the color and morphology of the leaf spots, these characteristics are very important and offer crucial details about their visual representation. The mean and variation of the grey stage of the red, green and blue channels of the spots correspond to the characteristics of color, and other characteristics correspond to the morphological and geometrical characteristics of the spots. It is easy for us to remove the characteristics of the disease leaf from the picture by using the segmentation technique. The image analysis here focuses on the extraction of shape characteristics and segmentation dependent on color. Contrast, energy, similarity, homogeneity are the characteristics usually used for analysis. Correlation points out the correlation between a pixel and their image Neighbor. The Help Vector Machine classifier used for this function is (CNN). This classifier belongs to a category of supervised methods of learning commonly used for classification and identification of patterns. Supervised learning is a machine learning algorithm that allows predictions about a new dataset. the research dataset, utilizing a proven dataset. the training dataset. CNN classifies the image by constructing hyper planes in a high-dimensional space, which is then used to identify data points belonging to various data points. Oh, lessons. If the number of samples increases in the training dataset, the accuracy of the CNN classifier gets higher. By choosing the best hyper plane that distinguishes between the data points of two groups, the classification using CNN is completed. The hyper plane with the biggest distance between the two groups is the strongest CNN hyper plane. A basic binary CNN classifier is used in the proposed solution.

The purpose of the proposed algorithm is to classify the images of pepper bell leaves into two classes using the CNN binary classifier as healthy leaves and bacterial leaves. To differentiate between irregular and regular leaves, the classifier uses the key points in the photos. However, the training picture that matches the healthy set that already storge in the dataset the classifier will classify it as healthy otherwise infected by bacterial [16].

### 3.2.5 Image Classification

In this stage we classify and divide the pepper bell leaf disease and take the feature derived from the leaf images representing the different diseases and train a suitable classifier that can offer good performance. In case to achieve our goals machine learning toolbox to train suitable classifier, CNN classifier was trained and used.

## 3.3 Get Started With OpenCV

In this project the software that we aiming to use is python, OpenCV is a library in python that allow you to have use many features for image processing as well as keras library with a Python interface.

### 3.3.1 Import Keras Library

Keras is an open-source software library which supports artificial neural networks with a Python interface. Keras serves as a TensorFlow library gui, Figure3.8 shows how to import keras layers and models library.

```python
# train_model.py
import os, json, numpy as np, tensorflow as tf
from tensorflow.keras import layers, Model

```

Figure 3.8 Import Keras Library On Python OpenCV.

### 3.3.2 Initializing Classifier

Figure(3.9,3.10) shows the first step would be initializing classifiers and the second step of Adding Maxpooling2d for data that been collected:

```
# نموذج CNN بسيط
model = tf.keras.Sequential([
    layers.Conv2D(32, (3,3), activation='relu', input_shape=(224,224,3)),
    layers.MaxPooling2D(2),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D(2),
    layers.Conv2D(128, (3,3), activation='relu'),
    layers.MaxPooling2D(2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])
```

Figure 3.9 Initializing CNN Classifiers

```
train_data = train_datagen.flow_from_directory(
    TRAIN_DIR, target_size=IMG_SIZE, batch_size=BATCH_SIZE,
    class_mode='categorical', shuffle=True, seed=SEED
)
val_data = val_datagen.flow_from_directory(
    VAL_DIR, target_size=IMG_SIZE, batch_size=BATCH_SIZE,
    class_mode='categorical', shuffle=False
)
```

Figure 3.10 Step Two of train the dataset.

## 3.4 Summary

In this study, we have divided the methodology into five stages to detect the bacterial spot on a pepper bell leaf, with techniques that been mentioned previously. Image acquisition is the first stage where dataset have been collected using smartphone color camera with resolution of 4128(h) * 3096(v) pixels. Image preprocessing where the image of the pepper bell leaves will be illuminated and isolated from the surrounding environment, this stage considered as preparation for the next stages. Image segmentation is the most significant stage where some techniques applied to identify and extract the leaf

from the image and select the bacterial spots in the infected leaves. Moreover, Features extraction, when the pepper bell leaves get effected by bacterial, batches spot will appear on the leaf. Batches spots on the crop leaf consider important units indicating the existence disease and regarded as indicator of crops disease. The final stage, which is image classification, in this stage CNN classifiers has proved itself on classifying the bacterial spots on the pepper bell leaves and applying layers on the dataset

. In addition, the dataset will be divided into two categories, healthy pepper bell leaves and bacterial spot infected pepper bell leaves using python OpenCV software and keras library which these libraries used in image processing.

# CHAPTER 4: RESULTS AND DISCUSSION

## 4.1 Introduction

This chapter explain about the output and the results of our project. Python software has been used to accomplish the objects in this paper, with importing OpenCV, keras and tensorflow libraries,other toolbox of deep learning and machine vision . 300 images of pepper bell leaves have been collected. The dataset of pepper bell leaves divided into two categories, 150 images of healthy pepper bell leaves and 150 images areinfected by bacterial spot of pepper bell leaves. A smartphone color camera used to collect the samples.

## 4.2 Capturing Image

The dataset collected using smartphone color camera with 4128(h) * 3096(v). collected two set of healthy and bacterial of pepper bell leaves Figures (4.1,4.2) shows samples of pepper bell leaves that been collected and saved in JPG format.


Figure 4.1 Samples of Healthy Pepper Bell Leaves


Figure 4.2 Samples of Infected Leaves of Pepper Bell .

### 4.2.1 Resizing The Image

Throughout collecting the image, the dimension was too big with resolution of 352x288 so, it was important to do the resizing image of pepper bell. The resizing was done in two dimensions to compare the observation of the result, by 227x227x3.

## 4.3 Experimental Results

In this experiment keras library has been applied to both CNN and contained convolution2D layers, Maxpooling layers, flatten layer and classificationlayer. The convolution2D with 64 filter and 3 represent RGB colors, this layer is the firstlayer that been adjust and applied however, maxpooling set to (2,2) of the pool size both of these layers can be added more than one time, with activation of relu

```python
def build_cnn(input_shape=(IMG_SIZE, IMG_SIZE, 3), num_classes=2):
    model = Sequential([
        Conv2D(32, (3, 3), activation="relu", input_shape=input_shape),
        MaxPooling2D(2, 2),

        Conv2D(64, (3, 3), activation="relu"),
        MaxPooling2D(2, 2),

        Conv2D(128, (3, 3), activation="relu"),
        MaxPooling2D(2, 2),

        Flatten(),
        Dense(256, activation="relu"),
        Dropout(0.4),
        Dense(num_classes, activation="softmax")
    ])

    model.compile(
        optimizer="adam",
        loss="categorical_crossentropy",
        metrics=["accuracy"]
    )

    return model
```

Figure 4.3 Setting the Layers

'Rectified linear unit'. Full connection layer with 128 filter is the last layer, from this layer the image would be classify as healthy or bacterial spot. Figure4.3 shows the set of the layers that been used in this project.

### 4.3.1 CNN Training

```python
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATA_DIR = os.path.join(BASE_DIR, "data")
OUTPUT_DIR = os.path.join(BASE_DIR, "output")
os.makedirs(OUTPUT_DIR, exist_ok=True)

MODEL_PATH = os.path.join(OUTPUT_DIR, "best_cnn_pepper.h5")
CLASS_MAP_PATH = os.path.join(OUTPUT_DIR, "class_mapping.json")

IMG_SIZE = 224
BATCH_SIZE = 16
EPOCHS = 20

# ============================================================
# Data generators
# ============================================================

train_dir = os.path.join(DATA_DIR, "train")
val_dir = os.path.join(DATA_DIR, "val")

train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    rotation_range=20,
    zoom_range=0.2,
    horizontal_flip=True
)

val_datagen = ImageDataGenerator(rescale=1.0 / 255)
```

Figure 4.4 Set the CNN Training

We initialize our optimizer with the training data and decline parameters we described above before we begin the training of our algorithm. As it almost always achieves faster and better global minimum integration relative to the other optimization techniques, we prefer the Adam optimization strategy Figiure4.4 shows the code.

### 4.3.2 CNN Testing

We only have to include the full route to the image and shows the image together with the predictor class or the pepper bell disease. Figure4.5 we randomly pick images from the dataset for data analysis and attempt to predict the plant image class or disease.

```python
def train_model():
    print("Starting CNN training for pepper leaf disease classification...")

    model = build_cnn(num_classes=num_classes)

    history = model.fit(
        train_generator,
        epochs=EPOCHS,
        validation_data=val_generator
    )

    model.save(MODEL_PATH)
    print(f"Model saved at: {MODEL_PATH}")

    generate_analytics_and_plots(model, history)

    return MODEL_PATH
```

Figure 4.5 CNN Testing Set

### 4.3.3 CNN Evaluation

By observing Figure4.6, we find that as the accuracy of the CNN training improves, the accuracy of validity increases. Likewise, the validation loss reduces as the instruction loss decreases, too. By adjusting the CNN training set or practicing on more images or training the model for more iterations, we mayachieve better outcomes.

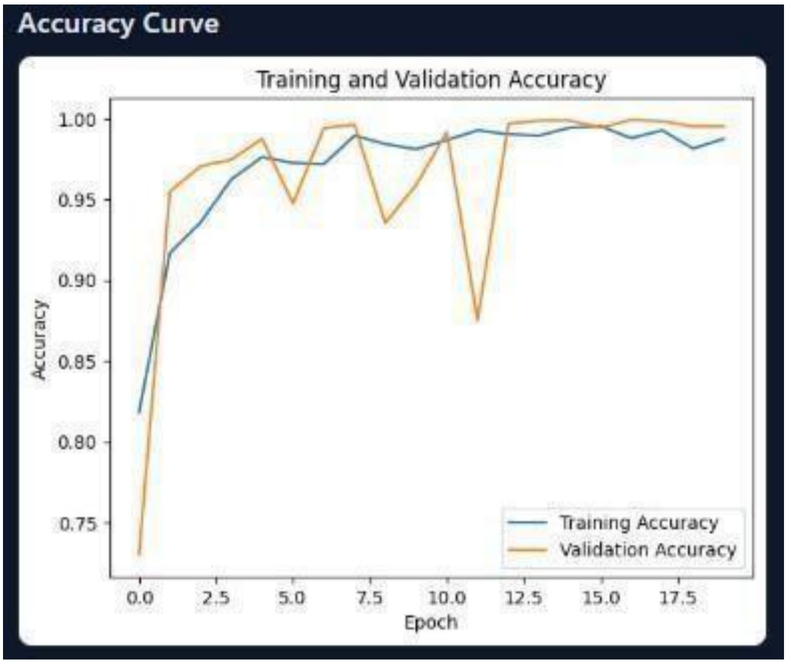



Figure 4.6 Training Loss and Validation Loss.

to achieve 99.93% accuracy.

The F1 score was also computed to measure the balance between precision and recall, especially important when dealing with datasets that may not be perfectly balanced. A high F1 score
reflects the model's ability to correctly identify diseased leaves while minimizing false alarms and missed detections.

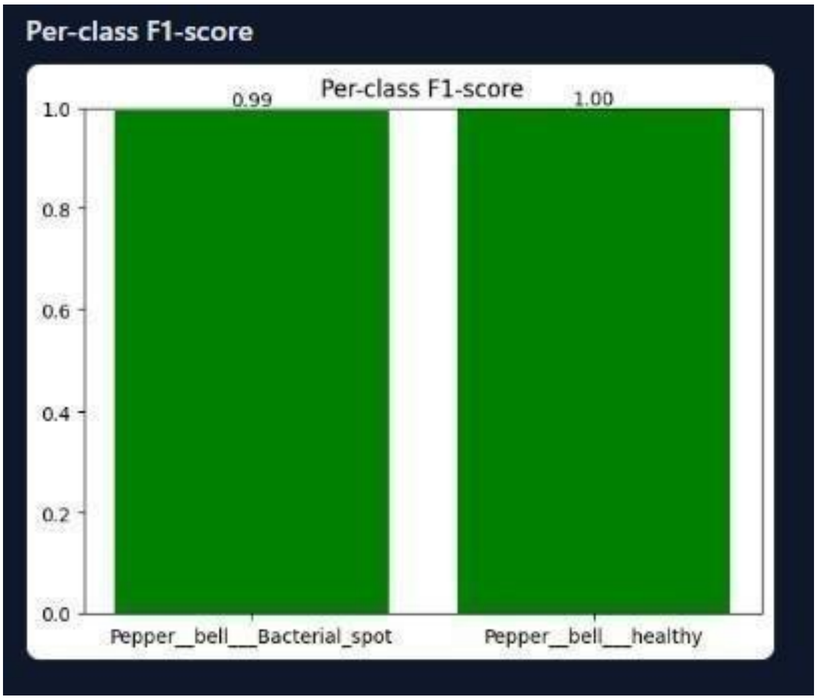



Figure 4.7 F1-Score

The confusion matrix provides a clear visualization of how well the model distinguishes between healthy and diseased pepper leaves by showing the number of correct and incorrect predictions for each class. A high number of true positives and true negatives in the matrix indicates strong classification capability and minimal misclassification.

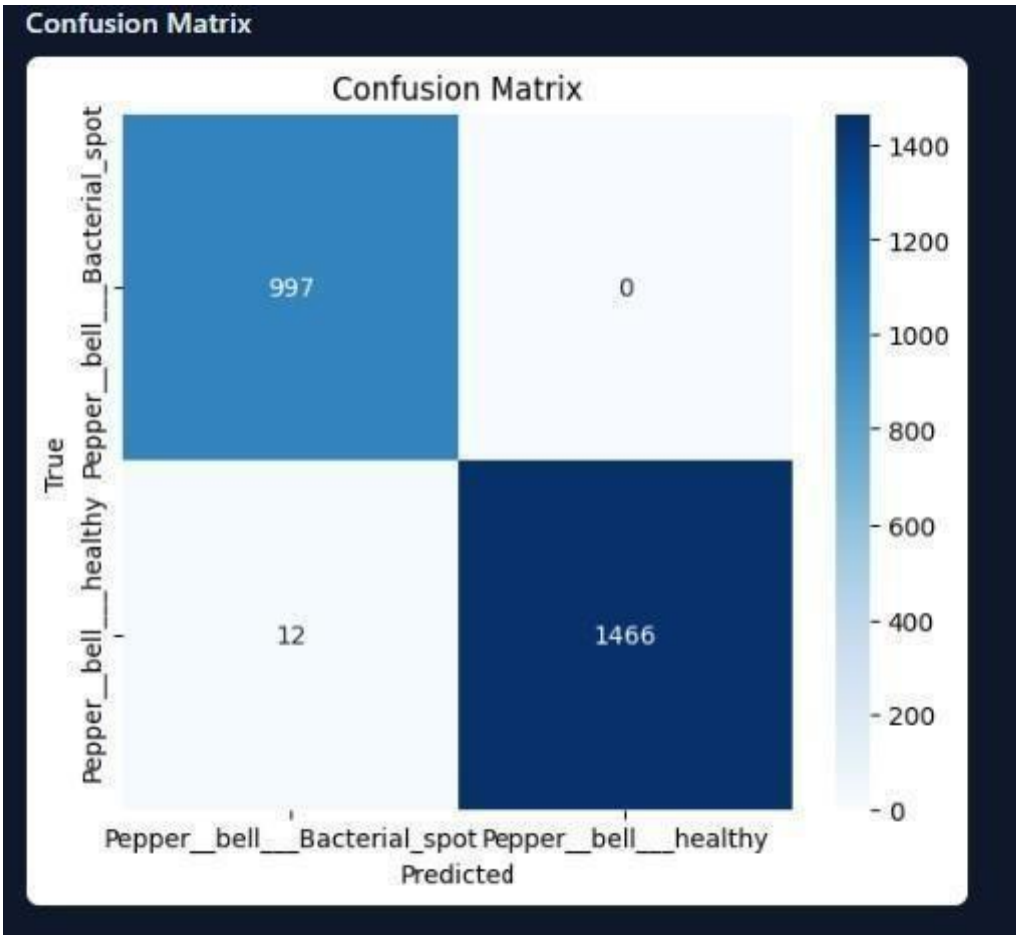



Figure 4.8 Confusion Matrix

Overall, the CNN model achieved reliable classification results, supported by strong confusion matrix performance, high F1-scores, and smooth convergence in the loss curves.

# CHAPTER 5: CONCLUSION

## 5.1 Conclusion

This study presented a complete approach for detecting pepper bell leaf diseases using image processing and a Convolutional Neural Network (CNN) model.
Pepper bell leaves were selected as the target crop due to their agricultural importance and susceptibility to bacterial infections. Images were collected using a standard smartphone camera, demonstrating that high-cost equipment is not required for reliable disease detection. The dataset was divided into two main categories: healthy pepper leaves and leaves infected by bacterial disease.

Image preprocessing played a crucial role in enhancing the quality of the input images by improving clarity, reducing noise, and highlighting relevant visual features. These steps ensured that the CNN model received clean and consistent data, enabling it to learn meaningful patterns. The extraction of color and texture features from the leaves proved essential for distinguishing between healthy and diseased samples. Overall, the CNN model successfully learned to classify pepper leaf conditions with strong accuracy, demonstrating the effectiveness of deep learning in agricultural disease detection.

## 5.2 Summary

The developed system contributes to the agricultural sector by providing an automated, efficient, and user-friendly method for detecting pepper bell leaf diseases. Manual inspection of leaves is often time-consuming, inconsistent, and dependent on human expertise. By contrast, the CNN-based system offers a reliable alternative that can analyze images and determine the health status of a leaf within seconds.

Throughout the project, the CNN model was trained, tested, and evaluated using the prepared dataset. The model achieved high accuracy and demonstrated stable performance across training and validation phases. The system processes each uploaded image through a sequence of automated steps—preprocessing, feature extraction, and classification—and finally outputs whether the leaf is healthy or infected.

The project successfully met all its objectives, including dataset preparation, CNN model development, system implementation, and performance evaluation. Challenges encountered during the project, such as image noise, inconsistent lighting, and dataset imbalance, were addressed through preprocessing techniques and careful model tuning. The final system provides a dependable tool that can support farmers, buyers, and agricultural workers by reducing manual effort, improving disease detection accuracy, and saving time.

## 5.3 Recommendation

Although the CNN model performed effectively, several improvements can enhance future versions of the system. The current approach relies primarily on color and texture features extracted from images. Future work may incorporate additional sensing modalities—such as leaf shape analysis, multispectral imaging, or thermal imaging—to improve classification accuracy, especially for early-stage infections.

Integrating the system into a mobile application or drone-based platform would enable real-time field monitoring and large-scale data collection. Furthermore, expanding the dataset to include more disease types, varying lighting conditions, and different growth stages would strengthen the model's robustness.
Combining image processing with other sensor technologies, such as humidity or nutrient sensors, could also provide a more comprehensive assessment of plant health. These enhancements would support the development of a more advanced and intelligent agricultural monitoring system.

## 5.4 Commercialization Potential

The proposed CNN-based disease detection system has strong potential for commercialization, particularly in regions where agriculture remains a major economic sector. Many countries, including Malaysia, Thailand, and the Philippines, face challenges such as labor shortages, reduced interest in farming among younger generations, and the need for more efficient crop monitoring tools. An automated leaf disease detection system can help address these issues by reducing reliance on manual inspection and enabling faster decision-making.

Agricultural companies, cooperatives, and individual farmers can adopt this technology to improve crop quality, reduce losses, and optimize farm management. The system can be integrated into mobile applications, smart farming platforms, or agricultural drones, making it accessible and scalable. With further refinement and expansion, the CNN-based detection system can become a valuable commercial product that supports modern, technology-driven agriculture.

# REFERENCES


T. Akshaya, G. Shruthi, P. S. M. Arige, M. A. Kiran, M. Thaile, and P. R. Gundalwar, "Bell pepper plant disease detection using deep learning models," in 2025 5th Int. Conf. on Intelligent Technologies (CONIT), 2025, pp. 1–6.

S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. Vishnu Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," Agric. Eng. Int. CIGR J., vol. 15, no. 1, pp. 211–217, 2013.

J. B. Lazaro and J. R. L. Jocson, "Image classification model for data sentry of plant diseases using convolutional neural networks," in 2025 17th Int. Conf. on Computer and Automation Engineering (ICCAE), 2025, pp. 171–175.

P. Mitkal, P. Pawar, M. Nagane, P. Bhosale, M. Padwal, and P. Nagane, "Leaf disease detection and prevention using image processing using MATLAB," Int. J. Recent Trends Eng. Res., vol. 2, no. 2, pp. 26–30, 2016.

G. Yashodha, B. Rajakumar, M. Krishnamoorthy, and R. Surendran, "Efficient plant disease detection using K-means clustering and DenseNet-based classification," in 2025 Int. Conf. on Electronics and Renewable Systems (ICEARS), 2025, pp. 1197–1204.

A. Julian, V. Jaiganesh, O. T. Thirumurugan, and B. I. Moses, "PepperShield: Disease detection with hyper-tuned precision," in 2024 IEEE 16th Int. Conf. on Computational Intelligence and Communication Networks (CICN), 2024, pp. 1242–1248.

P. Pallewatta et al., "BellCrop – A bell pepper leaf dataset for disease classification and yield enhancement using machine learning," in IECON 2024 – 50th Annual Conf. of the IEEE Industrial Electronics Society, 2024, pp. 1–7.

T. A. Mir, D. Banerjee, R. Chauhan, and H. S. Pokhariya, "Augmenting bell pepper disease detection using CNN and SVM," in 2024 4th Asian Conf. on Innovation in Technology (ASIANCON), 2024, pp. 1–5.

R. Singh et al., "An efficient CNN-based system for classifying pepper bell plant conditions," in 2024 5th Int. Conf. on Electronics and Sustainable Communication Systems (ICESC), 2024, pp. 1133–1139.

A. Suryavanshi, V. Kukreja, and A. Dogra, "Capsicum plant advanced disease detection using CNN-SVM fusion," in 2024 Intelligent Systems and Machine Learning Conf. (ISML), 2024, pp. 589–594.

N. Thapliyal et al., "Diagnosis of plant infection using vision transformer models," in 2024 IEEE Int. Conf. on Computing, Power and Communication Technologies (IC2PCT), 2024, vol. 5, pp. 44–48.

A. R. Ghodekar et al., "Tomato plant leaf disease classification using deep learning," in 2023 16th Int. Conf. on Developments in eSystems Engineering (DeSE), 2023, pp. 685–690.

G. Kaur et al., "An automated approach for detection and classification of plant diseases," in 2023 2nd Int. Conf. on Futuristic Technologies (INCOFT), 2023, pp. 1–6.

L. Rababa et al., "Pepper leaf diagnosis using deep-net with low-dimensional image classification," in 2023 14th Int. Conf. on Information and Communication Systems (ICICS), 2023, pp. 1–5.

A. A. AlArfaj et al., "Multi-step preprocessing with UNet segmentation and transfer learning model for pepper bell leaf disease detection," IEEE Access, vol. 11, pp. 132254–132267, 2023.

L. J. D. Leynes et al., "Development of identification system for plants using YOLO algorithm," in 2023 IEEE Eurasia Conf. on IoT, Communication and Engineering (ECICE), 2023, pp. 469–473.

J. Tan et al., "Chili plant health status classification using random forest," in 2023 Int. Conf. on AI and Data Sciences (AiDAS), 2023, pp. 212–217.

F. T. J. Faria et al., "Classification of potato disease using hybrid deep learning framework," in 2023 IEEE CCWC, 2023, pp. 820–826.

M. Shoaib et al., "Leveraging deep learning for plant disease and pest detection: A comprehensive review," Frontiers in Plant Science, vol. 16, pp. 1–20, 2025.

A. Upadhyay et al., "Deep learning and computer vision in plant disease detection: A comprehensive review," Artificial Intelligence Review, vol. 58, 2025.

S. Wang et al., "Advances in deep learning applications for plant disease detection," Remote Sensing, vol. 17, no. 4, pp. 1–25, 2024.

A. Y. Ashurov et al., “Enhancing plant disease detection through deep learning,” Frontiers in Plant Science, 2025.
N. Kumar et al., “Artificial intelligence and machine learning in plant disease detection,” Agrospheres e-Newsletter, vol. 6, no. 8, pp. 1–2, 2025.

# Transforming Medical Imaging with CNN-Based Detection Systems

**By Asif Syed**

**(2025–2026)**

## Abstract

This research project aims to develop an end-to-end deep learning system designed to detect and classify cases of COVID-19 and pneumonia using chest X-ray images through Convolutional Neural Networks (CNNs) . The system leverages both raw DICOM format medical imaging data and publicly available datasets from platforms such as Kaggle to train models capable of distinguishing between normal, pneumonia, and confirmed COVID-19 cases with high accuracy. The developed model was integrated into a Flask-based web application, enabling real-time image classification and diagnosis support for healthcare professionals.

In addition to traditional deep learning techniques, this study explores the use of Google Teachable Machine, a no-code AI training platform, to democratize access to machine learning capabilities for non-technical users. Emphasis was placed on preprocessing steps such as Extraction of DICOM images from PACS server and DICOM-to-PNG conversion, dataset balancing, and hyperparameter tuning to enhance model performance and generalization.
The findings indicate that while initial models showed signs of overfitting, retraining with regularization and early stopping significantly improved robustness. The hosting and training model on Google Teachable Machine demonstrates the potential for quickly deploying AI-based diagnostic tools in real-world clinical environments, especially in resource-constrained settings where rapid diagnosis is critical during the pandemics and epidemics.

**Table of Contents**

**Figures**



## Keywords

CNN, COVID-19 Detection, DICOM, Deep Learning Pneumonia, Flask Application, Google Teachable Machines, Medical Imaging,

# 1. Introduction

## 1.1 Problem Statement

The outbreak and subsequent worldwide expansion of the novel pneumonia-causing virus coronavirus 2 (SARS-CoV-2) have caused an unprecedented world wide public health emergency associated with important morbidity, mortality and socio-economic disruptions. By early 2025, the pandemic has killed more than 7 million people worldwide and infected hundreds of millions more, with a series of resurgences continuing to stress health systems in both advanced and developing nations. The current emergency has highlighted some major shortfalls in our global health architecture, especially in the kind of diagnostic capability which is necessary for a well-functioning pandemic readiness.

Classic diagnostic methods for COVID-19, such as Reverse Transcription Polymerase Chain Reaction (RT-PCR) adequately identified viral genetic materials. But these methods possess the following problems: they come with a number of severe limitations which limit their applicability, particularly in resource-limited or heavily-loaded environments. (c) Materials RT-PCR testing generally requires;

1. Specialized laboratory infrastructure with controlled environments
2. Highly trained technical personnel for sample processing and analysis
3. Expensive analytical instrumentation and reagents
4. Considerable time for sample collection, transportation, processing, and result reporting (often 24-48 hours)
5. Complex supply chains that are vulnerable to disruption during global crises

These limitations make traditional testing methods especially challenging in remote, resource poor, or economically disadvantaged areas, where health system infrastructure is generally poor. Moreover, in the setting of surge demand, not even well-resourced health systems can keep pace with demand, and there is a marked backlog in diagnosis, propagating ongoing community transmission.

To address these challenges, medical imaging methods such as chest X-rays (CXRs) and computed tomography (CT) have become important adjunct methods for diagnostic workup. The advantages of these imaging methods over molecular testing are:

1. Wider availability in healthcare settings, including in resource-limited areas
2. Rapid acquisition and processing (results potentially available within minutes)
3. Ability to visualize pathological changes in lung tissue that may indicate viral pneumonia
4. Potential for detecting COVID-19-related abnormalities in patients with false-negative RT-PCR results

However, the interpretation of medical images presents its own set of challenges. Traditional radiological assessment relies on human expertise, which introduces several limitations:

1. Global shortage of qualified radiologists, particularly in low and middle-income countries
2. Potential for inter-observer variability and human error
3. Cognitive fatigue during high-volume periods, potentially compromising diagnostic accuracy
4. Time-intensive nature of manual interpretation, creating bottlenecks during surge periods

These limitations underscore the urgent demand for automated, reliable, and readily available diagnostic support systems, which can complement human skills and facilitate fast diagnostics. Recently, AI, especially deep learning techniques, for example, CNN, has shown great promise in this field which can realize fast, standardized and large-scale image analysis.

CNNs are expert in image classification, feature extraction and segmentation process, that could be very useful in identifying the subtle patterns and abnormalities present in medical images that indicates the certain pathologies. Their capacity to learn hierarchical features from large training databases allows them to identify visual patterns that are too subtle to be recognized by human experts or that have variabilities among specialists.

Despite the great progress achieved in the AI for medical image analysis, even the transformative models need further works before being available to the clinical practitioners as:

1. Limited availability of comprehensive, diverse, and well-annotated training datasets
2. Technical barriers to implementation, particularly for healthcare professionals without specialized computing expertise
3. Integration challenges with existing clinical workflows and Picture Archiving and Communication Systems (PACS)
4. Concerns regarding explainability, transparency, and clinical validation
5. Regulatory and ethical considerations related to automated diagnostic systems

This research addresses these challenges by developing an end-to-end deep learning system for the detection and classification of COVID-19 and pneumonia from chest X-ray images. By combining traditional CNN-based approaches with accessible no-code AI platforms like Google Teachable Machine, this study aims to democratize access to advanced diagnostic tools while maintaining high standards of accuracy and reliability.

### 1.2 Objectives

- End to End workflow of training machine learning model including using no-code tool
- Extraction of medical images from PACS
- Develop a preprocessing tool to convert DICOM images to Lossless PNG
- Develop a CNN-based model for COVID-19 detection.
- Train and evaluate the model using X-ray image datasets.
- Deploy the model using based web application for quick availability
- Analyze model accuracy, biases, and ethical implications.
- Quick deployment for limited resources areas for first opinion.

## 2. Literature Review

The application of artificial intelligence to medical imaging has evolved significantly over the past several decades, transitioning from rudimentary pattern recognition systems to sophisticated deep learning architectures. This evolution can be broadly categorized into four distinct phases, each characterized by specific technological advancements and clinical applications.

## 3. Early Automated Image Analysis (1960s-1980s)

The first decade of the efforts on automatic analysis of medical images was conducted at the end of 1960s, along with the growing number of computerized tomography (CT) and other digital imaging modalities. Rule-based strategy and elementary statistical techniques used by these early systems to identify simple patterns in medical images. Lodwick et al. (1963) introduced some of the first computer-aided diagnosis systems for chest X-rays, utilizing statistical pattern recognition methods to detect lung nodules. These early systems suffered from heavy computational constraints and depended largely on handcrafted features.

In the 1970s and 1980s, scientists in general started investigating more advanced techniques for image segmentation, feature extraction and classification. Meyers et al. (1976) proposed the automated analysis of mammograms, while Chan et al. (1987) developed the early PCAD systems for pulmonary nodule detection in chest radiographs. Such systems often based on classical image processing methods like edge detection, thresholding, and region growing, in association with the use of statistical classifiers such as discriminant analysis and decision trees.

However, despite these technological breakthroughs, the clinical impact of these early instruments was. to a great extent limited, ascribable to several reasons:

1) Lack of computing capacity for analyzing high quality medical images
2) Restricted accessibility to digital imaging data for algorithm research and testing
3) Dependence on hand-engineered features that capture the full complexity of medical images rarely.
4) No integration with HIS or EMR tools

### 3.1 Machine Learning Era (1990s-2000s)

In the 1990s, the direction of medical imaging analysis was completely changed when machine learning techniques were shown to be capable of learning relevant features from training data which could be used to classify an image. SVMs, Random Forests and other statistical learning methods started to replace the rule-based models with better performance and flexibility.

Giger et al. (1994) showed the value of machine learning in mammo- graphic lesion classi cation, Armato et al. (2001) used similar methods for lung nodule detection in CT images. Such systems usually included classical image processing for feature extraction and machine learning methods for classification, with the performances that were close to clinical use for certain and well-defined tasks.

These approaches continued to evolve in the early 2000s, as investigators began to a apply more elaborate feature extraction features, and ensemble learning methods. Computer-Aided Detection (CAD) software What developed: Over this time, the CAD industry matured, with this era representing the first widespread clinical use of AI in radiology, especially for mammographic screening. Nevertheless, these systems were still domain-expertise-dependent in feature engineering, and often lack generalization ability for diverse patient populations and imaging protocols.

As a consequence of the COVID-19 pandemic the application of artificial intelligence to medical diagnostics had gained further momentum. With millions of cases confirmed around the world, the need for fast, accurate and affordable tests has never been higher. Traditional diagnostic procedures, e.g. RT-PCR testing, are dependable, but are slow (with long turn-around times), have limited availability of the test kits and they face logistical barriers. Therefore, ML and CV algorithms became competing candidate solutions in detecting and classifying respiratory diseases using medical imaging data.

Chest X-ray (CXR) and computed tomography (CT) are widely used imaging modalities for diagnosing pneumonia as well as lung related diseases including those caused by the SARS-CoV-2 virus. These imaging approaches enable the clinicians to visualize the lung structures as well as to identify abnormalities, such as ground-glass opacities, consolidations, and interstitial thickening that are frequently seen in viral pneumonia patients and those with severe COVID-19.

However, manual interpretation of such images is time-consuming and needs expertise wherever the same is not available at all time, particularly timely or in remote or underserved area. This development has resulted in a blossoming of automated image analysis systems based on artificial intelligence, especially deep learning approaches like Convolutional Neural Networks (CNNs).

### 3.2 Deep Learning Revolution (2010s)

The age of AI in medical imaging AI is moving along the lines of greater maturity, clinical penetration and regulations. Contemporary methods prioritize attributes beyond performance, including interpretability, fairness, robustness, and clinical utility. A few trends are driving today's landscape:

**Multimodal Integration:** The latest work addresses the integration of information from multiple imaging modalities (e.g., CT, MRI, PET) and non-imaging data sources (e.g., electronic health records, genomics) for more comprehensive diagnostic support. Yao et al. (2021) have showed that multimodal approaches can increase the ACC of the prognosis drastically in oncology applications.

**Federated Learning:** In order to combat the issues regarding privacy and data silos, federated learning approaches allow models to be trained between multiple institutions while preventing raw patient data from being shared. Sheller et al. (2020) have also shown that federated learning can be used for brain tumor segmentation in a multi-institutional scale, and it could produce equivalent results to the centralized training.

**Explainable AI:** Due to regulatory demands and clinical deployment concerns that underline interpretability, different visualisation and explanation methodologies have been proposed by researchers for deep learning decisions. The use of Gradient-weighted Class Activation Map (Grad-CAM) type methods to generate heat maps of those regions a model looks at when making a prediction has become standard (Selvaraju et al., 2017).

**Regulatory Pathways:** The formation of regulatory pathways for AI-based medical devices, including the FDA's proposed regulatory framework for AI/ML-based Software as a Medical Device (SaMD), has also contributed increased clarity in principles governing clinical translation. Numerous AI-models have been approved due to this approach, for clinical utility.

**Democratization of AI:** The development of no-code and low-code platforms drastically diminishes technical barriers to utilization of AI, and allows healthcare workers without programming skills to create and deploy custom models for specific clinical applications.

AI for COVID-19 diagnosis is an expression of such a convergence, which rests on decades long methodological advances, and however tailored to meet the specific problems of a global pandemic. The fast tracking of AI-based COVID-19 detection systems are indicative of the maturity of the field and its ability to respond quickly to new healthcare challenges.

## 3.3 Application of CNN in Medical Imaging

Convolutional Neural Networks have emerged as the dominant architectural paradigm for medical image analysis due to their ability to automatically learn hierarchical features from raw image data. The evolution of CNN architectures for medical imaging has been characterized by increasing depth, specialized components, and task-specific optimizations.

### *3.3.1 Foundational Architectures*

The earliest CNN architectures applied to medical imaging were adaptations of networks originally designed for natural image classification. These include:

**LeNet:** Developed by LeCun et al. (1998), this pioneering CNN architecture established the basic pattern of alternating convolutional and pooling layers. While originally designed for handwritten digit recognition, early adaptations were applied to medical image classification tasks.

**AlexNet:** Krizhevsky et al.'s (2012) architecture marked a significant advancement with deeper layers, ReLU activations, and dropout regularization. Early medical applications of AlexNet typically employed transfer learning, using weights pre-trained on ImageNet and fine-tuning for specific medical tasks.

**VGGNet:** Simonyan and Zisserman (2014) introduced this architecture, characterized by its simplicity and uniform structure with small (3×3) convolutional filters. The regularity and depth of VGG made it particularly suitable for transfer learning in medical applications, as demonstrated by Anthimopoulos et al. (2016) for interstitial lung disease classification.

**GoogLeNet/Inception:** Szegedy et al.'s (2015) architecture introduced inception modules that process input at multiple scales simultaneously, enabling efficient feature extraction at different levels of abstraction. This multi-scale approach proved particularly valuable for medical images where relevant features may exist at various scales.

**ResNet:** He et al.'s (2016) introduction of residual connections addressed the vanishing gradient problem in very deep networks, enabling the training of networks with hundreds of layers. ResNet and its variants have been widely adopted in medical imaging, with Rajpurkar et al. (2017) demonstrating their effectiveness for pneumonia detection in chest X-rays.

### *3.3.2 Specialized Medical Imaging Architectures*

Building on these foundational architectures, researchers have developed specialized networks optimized for specific medical imaging tasks:

**U-Net**: Ronneberger et al.'s (2015) architecture, characterized by its U-shaped encoder-decoder structure with skip connections, has become the de facto standard for medical image segmentation. The architecture's ability to combine contextual information from the contracting path with precise localization from the expanding path makes it particularly effective for delineating anatomical structures and pathological regions.

**V-Net:** Milletari et al. (2016) extended the U-Net concept to 3D volumes, enabling direct segmentation of volumetric medical data such as CT and MRI scans. This architecture incorporated residual connections and a novel objective function based on the Dice coefficient, further improving segmentation performance.

**DenseNet:** Huang et al.'s (2017) architecture, which connects each layer to every other layer in a feed-forward fashion, has shown particular promise in medical applications due to its parameter efficiency and feature reuse. Rajpurkar et al. (2018) demonstrated DenseNet's effectiveness for detecting multiple pathologies in chest X-rays.

**CheXNet:** Rajpurkar et al.'s (2017) adaptation of DenseNet-121 for chest X-ray analysis demonstrated radiologist-level performance in pneumonia detection and has become a benchmark architecture for thoracic image analysis. The success of CheXNet highlighted the potential of deep learning for COVID-19 detection in the subsequent pandemic.

**COVID-Net:** Wang et al. (2020) developed this specialized architecture for COVID-19 detection from chest X-rays, employing a lightweight design optimized for clinical deployment. COVID-Net incorporated architectural design choices specifically tailored to the unique radiographic presentation of COVID-19.

Together, these studies highlight the increasing contribution of AI-aided diagnostics to the acceleration and accuracy of medical decisions. But they also own up to various difficulties as well:

Over-fitting from inadequate or imbalanced datasets

Invisibility of black-box models

Ethical considerations in data privacy and consent

The importance of strong deployment mechanisms in clinical workflows

### 3.4 No-Code AI in Medical Imaging Google Teachable Machine Context

Google Teachable Machine, highlighted in this research, exemplifies the potential of no-code AI platforms in healthcare applications and a quick first opinion tool for the limited resources settings.

**Accessibility Features:** The platform's intuitive drag-and-drop interface, real-time feedback, and visual model evaluation tools make it accessible to healthcare professionals without programming background. This accessibility is particularly valuable in resource-constrained settings where technical expertise may be limited.

**Educational Value:** Beyond practical applications, Teachable Machine serves as an educational tool that can help healthcare professionals understand the fundamentals of machine learning, potentially fostering greater AI literacy in clinical settings.

**Rapid Prototyping:** The platform enables rapid development and iteration of models, allowing clinicians to quickly test hypotheses and assess the potential value of AI for specific diagnostic tasks before committing resources to more complex development efforts.

**Deployment Options:** Models developed in Teachable Machine can be exported in various formats, including web-based applications that can be shared with colleagues or integrated into simple clinical workflows without requiring specialized infrastructure.

However, Teachable Machine also has limitations in the medical context:

**Model Complexity:** The platform supports relatively simple model architectures compared to custom-developed solutions, potentially limiting performance on complex medical imaging tasks.

Data Privacy: Training occurs in the browser, addressing some privacy concerns, but the platform may not meet all regulatory requirements for handling sensitive medical data.

**Limited Preprocessing:** The platform offers minimal options for specialized medical image preprocessing, which can be crucial for optimal model performance.

**Explainability Constraints:** The platform provides limited tools for model interpretation and explanation, which are increasingly important for clinical adoption and regulatory approval.

While traditional deep learning methods are developed in parallel, there has been a rising interest in a new class of no-code AI tools, for example Google Teachable Machine, that allows users to construct and train machine learning models without writing code. These platforms lift the bar for AI adoption, especially for not-technical healthcare professionals who have no formal training in programming or machine learning.

No-code tools, that provide intuitive interfaces as well as pre-trained models, further speed up the prototyping and toying of AI-centric diagnostic systems. Although they are less customizable compared to coding-based methods, they present an alternative viable tool for rapid application and validation of AI models in healthcare.

### 3.5 Embedded AI Models in Clinical Workflows

For AI-powered diagnosis systems to be genuinely useful, they have to be deployed in a user-friendly and non-intrusive fashion into the typical clinician's workflow. This extends beyond technical aspects like model deployment, API design, and cloud hosting to human factors like clinician trust, interpretability, and usability.

To tackle this challenge, a number of researchers have started to incorporate AI models into web-based interfaces, which also enable healthcare personnel to interact with the model through intuitive means such as browsers and mobile applications. This further accelerates the advancement of simple, yet useful, apps like Flask, which is a web framework for Python that is both modular and light, and can be easily linked to several deep learning libraries,

In this project, we have implemented both code-based web application hosting and no-code secure application hosting using Google Teachable Machine . This dual approach facilitates the rapid development and deployment of image classification models, enabling both IT professionals and healthcare practitioners to leverage their respective expertise. While IT specialists can focus on deploying robust, scalable web solutions using frameworks like Flask, healthcare professionals can utilize no-code platforms to build and share AI-driven diagnostic tools without requiring extensive programming knowledge. This synergy enhances the accessibility, usability, and clinical applicability of AI-based medical imaging systems

## 4. Methodology

This research is conducted in a mixed approach of technical development, system experimentation and qualitative evaluation so as to address the challenge of creating an end to end machine learning and deep learning medical imagining analysis solution. Research design and study design revolves around the creation and assessment of an end-to-end deep learning system for COVID-19 and pneumonia detection from chest X-rays, being designed in focus of accessibility and clinical integration.

There are several fundamental principles to guide the methodological approach:

**Clinical Applicability:** We rank solutions that could be easily applied in clinical reality, especially in resource-limited settings

**Technical solidity:** Making sure that the models you develop measure up to high standards of accuracy, reliability and generalizability.

Accessibility Investigating the ways to minimise the technical inconvenience to harness healthcare AI.

**End-to-End View:** The entire process from data collection to clinical production, instead of just the model building phase.

This holistic view separates our research from many competing contributions that tend to merely look at model architectures and performance measures, without paying significant attention to the entire context" of putting models into real-world use. By covering the entire process in between data extraction and clinical deployment, this study intends to bridge the gap from technical innovation to clinical utility in practice.

The approach consists of several interwoven parts:

**Full Workflow Development:** Developing a full workflow of medical image analysis from data read to deployed model.

**Data Reuse and Pre-processing:** Making available and the preparation of domain-specific datasets for model training and evaluation.

**Model Training and Development:** Developing and training CNN models of COVID-19 detection.

**System Integration and Deployment:** Development of user friendly interfaces for clinical deployment

**Evaluation and Validation:** How is the performance, usability, and clinical impact of models assessed

Each of these parts is explained in detail in the subsequent sections with reference to methodological decisions, technical aspects and the underlying motivation for the design choices.

The methodology section recalls the research problem, related objectives, and justifies the methodological choices taken to achieve these goals. This study aims to develop a deep learning-based COVID-19 and Pneumonia detection system utilizing Convolutional Neural Networks (CNNs) and a Flask-based X-ray application. Given the need for early and accurate detection, this research integrates a no-code AI training approach using Google Teachable Machine alongside traditional deep learning frameworks. Additionally, considering that hospitals primarily store X-ray images in DICOM format, a pre-processing step was introduced to convert DICOM images to PNG before feeding them into the deep learning models.

To achieve these objectives, the methodology follows a structured approach:

## 5. End to End Workflow of Medical Imaging Analysis

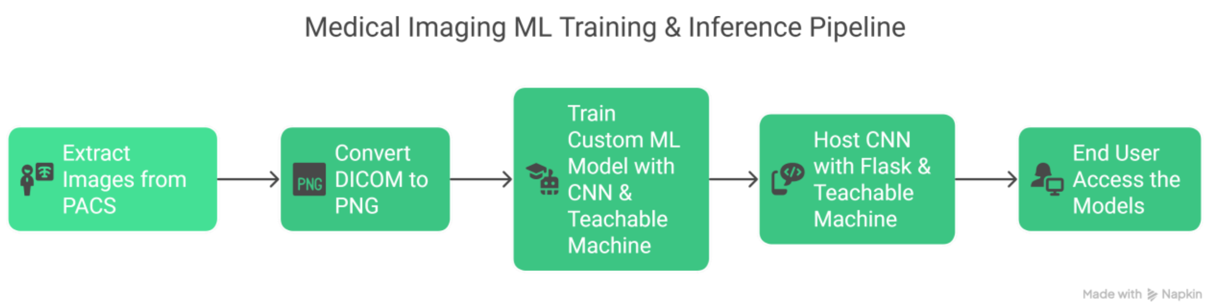



**Figure 1 : End-to-end ML training and inference workflow**

The above end to end workflow in Figure 1 shows a full-fledged end-to-end pipeline to train and deploy ML models in medical imaging and indicates key steps that must be in place in order to translate raw hospital data into clinical setting. This full continuum of care is important because it encompasses not only the mechanics of training a model, but the logistics of inserting AI into healthcare practices. In contrast to much of the prior work that may have considered only model training while overlooking the larger picture of data extraction, preprocessing, deployment and access, this end-to-end pipeline underscores the critical role that each step plays in supporting the usability and scalability of diagnostic systems driven by AI.

First, the process of extracting images from PACS (Picture Archiving and Communication Systems) is a crucial initial step that guarantees usage of real and clinically relevant data. The wide variety of proprietary formats that can't be used directly by models is also True for medical images; since its adoption as the standard for medical images, most hospitals store medical images in DICOM, which can't be directly used with any of the major deep learning libraries. By having this conversion step explicitly, the pipeline recognizes the need to preprocess the raw hospital data to be compatible with AI models. This preprocessing is easy to miss in research papers, because the latter draw on datasets from public repositories (Kaggle etc) and cohorts that have been pre-curated by others. Although these datasets are useful for initial exploration, they might not be diverse enough or representative of real-world clinical data, and application of these models in practice may lead to biased or unreliable models.

Secondly, the hybrid approach of model building, i.e., CNN-based training as well as no-code solutions using Google Teachable Machine, presents a balanced perspective. Existing work mostly concentrates on one side of the spectrum, either being fully dependant on highly skilled programmers or restricting themselves to zero-code tools without investigating their boundaries. By combining these two methods, the pipeline addresses a broader audience: IT specialists can refine (fine-tune) more complex models, while health professionals can use intuitive (no-code) platforms to develop and deploy AI tools easily. This two-pronged approach increases the interpretability and applicability of AI in medical analysis. This brings AI nearer to the point of care.

Additionally, hosting and deploying (i.e., the last section) where the into a Flask web application and consumable by Teachable Machine illustrates an aspect concerning real world application. Most of the researchers only test their model on validation datasets and putting the model into real-time usage is out of their horizon. Through integration of hosting, this pipeline guarantees that the developed system can be easily adopted by the end user (e.g., clinicians, radiologists) without interfering with their workflow patterns. This end-to-end view is critical for turning academic wins into practical, effective solutions.

Last but not least, the pipeline emphasizes the importance of user designed, or better said user-centric, interaction, which enables end people to explore and use the models via an intuitive interface. This is commonly overlooked in conventional research, where most of the attention is put on developing models with high accuracy rates with little attention to how the models would be utilized in the real world. By coupling the flow from data extraction to use case, the pipeline aims to ensure that the systems developed are not just technically robust and functional but also have clinical utility to tackle the pressing need for accurate and rapid diagnostic tools in healthcare.

Overall, the above picture is a big step forward compared with prior work, because it offers an end-to-end solution connecting theoretical AI models to practice of healthcare. It highlights data extraction, pre-processing, model training, deployment, and user accessibility and is a more comprehensive and generalized framework for medical image transformation by deep learning.

## 5.1 Data Collection and Preprocessing

### *5.1.1 Dataset Acquisition*

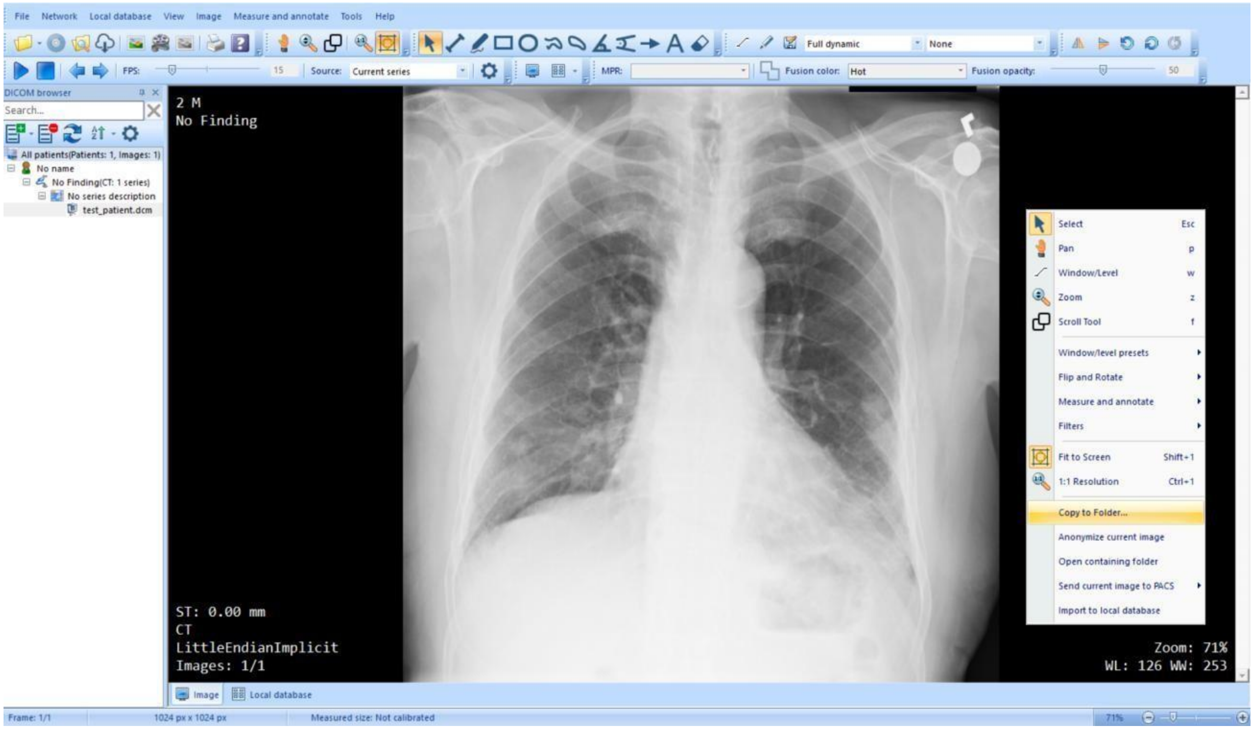


**Figure 2: Raw DICOM files extraction**

The above Figure 2 image depicts a chest X-ray displayed in a DICOM viewer, demonstrating a Medical Imaging Data extraction directly from the PACS (Picture Archiving and Communication System) clear and well-processed radiographic image of the thoracic region. The target here is to extract the high-quality imaging data to avoid extracting low quality images and copy the images to a folder based on the intended class like viral phenumai, covid 19 and normal. This setup highlights the importance of working directly with raw DICOM images, which are the standard format used by hospitals and medical facilities for storing and transmitting radiographic data.

Acquiring raw DICOM images is a crucial step in developing AI-driven medical imaging systems because it ensures compatibility with real-world clinical workflows. Unlike pre-processed datasets commonly found on platforms like Kaggle, raw DICOM files retain all original metadata and pixel-level details, allowing researchers to work with the same data formats used in hospital settings.

Acquiring raw DICOM images is a foundational and often overlooked step in developing AI-powered medical imaging systems, as it ensures authenticity by providing data that closely mirrors real-world clinical environments, thereby enhancing the model's relevance and applicability. These images contain rich metadata—such as patient demographics, acquisition settings, and anatomical orientation—which is crucial for regulatory compliance, quality control, and accurate diagnosis, yet is frequently absent in pre-processed datasets. Starting with raw DICOM also allows for controlled preprocessing, minimizing distortions during format conversion (e.g., to PNG or JPEG) and ensuring optimal data integrity for deep learning. Furthermore, using DICOM supports seamless integration with hospital PACS systems, improving scalability and deployment readiness. Ethically, working with raw DICOM enables systematic anonymization, addressing privacy concerns and ensuring responsible handling of sensitive health information. In contrast, many projects bypass this critical step, opting for convenience over realism, which limits their practical utility. By prioritizing raw DICOM data, this project establishes a robust, ethically sound, and clinically aligned foundation for AI-driven diagnostic tools, making them more effective and deployable in real healthcare settings.

The dataset used in this project consists of labeled chest X-ray images collected from publicly available sources such as Kaggle and other open-access repositories. The dataset includes three main classes:

- Normal lungs
- Viral pneumonia
- Confirmed cases of COVID-19

Efforts were made to ensure that the dataset was balanced across all classes to avoid bias in model predictions while training with Google Teachable Machine.

*5.1.2 DICOM-to-PNG Conversion*

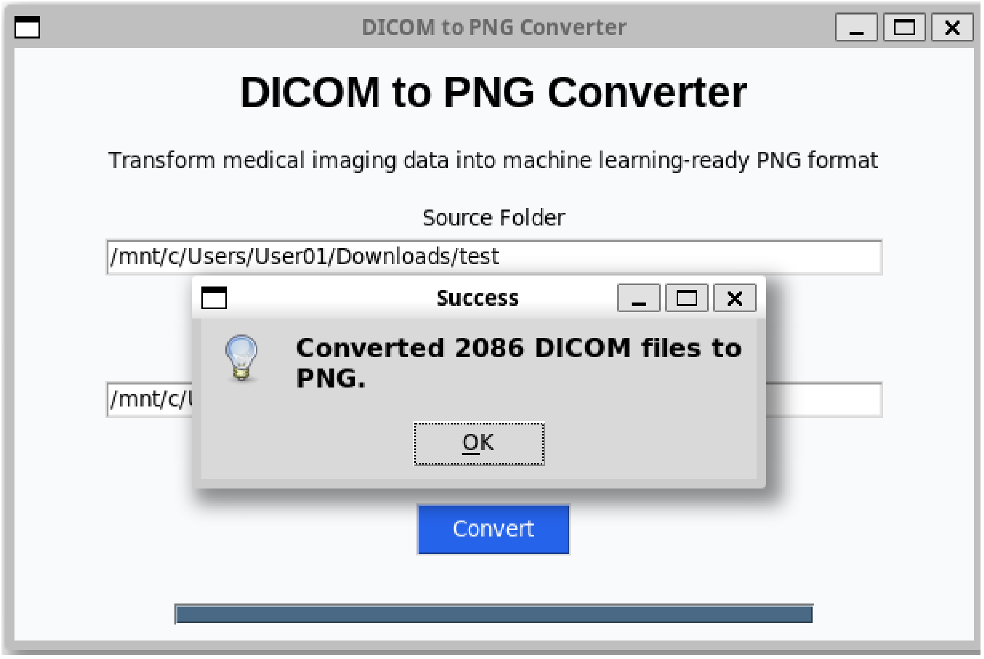



Figure 3: Conversion of raw DICOM files to PNG

The Figure 3 show the Python program developed is a DICOM-to-PNG converter designed with a user-friendly graphical interface using the **tkinter** library, making it accessible to non-technical healthcare professionals. The tool enables users to easily select a folder containing DICOM (.dcm) files—commonly used in medical imaging such as X-Ray, CT and MRI scans—and convert them into PNG image files without requiring any coding knowledge. It automatically searches through the selected source folder, including subdirectories, to locate all DICOM files. Each file is then read using the **pydicom** library, and its pixel data is converted into a grayscale image using the **PIL** (Pillow) library. These images are saved in the specified output folder as PNG files, preserving the original filenames for easy reference. A progress bar and informative messages enhance the user experience by providing real-time feedback during the conversion process. One of the key advantages of this script is that it saves images in the lossless PNG format, ensuring high-quality image preservation ideal for analysis or machine learning tasks.

The final goal of this tool is to bridge the gap between complex medical imaging formats and practical usability for healthcare professionals who may not have technical or programming expertise. By offering a simple, intuitive interface with drag-and-drop functionality and clear visual cues, the script empowers clinicians, researchers, and medical staff to efficiently preprocess DICOM images for use in presentations, educational materials, or basic analysis tasks. Its ability to perform batch conversion, support recursive folder structures, and provide visual feedback through a progress bar makes it both efficient and user-friendly. While it does not include advanced image processing features like windowing or normalization, it serves as a straightforward solution for converting medical images into a widely supported and high-quality format. This makes it especially useful for those preparing datasets for machine learning, archiving, or sharing with team members who rely on standard image viewers and software tools.

**5.2 Model Training & Optimization**:

Traditional CNN Approach**:**

1. A CNN model is trained using Python-based deep learning frameworks (TensorFlow/Keras).
2. Feature extraction techniques allow the model to classify images efficiently.
3. Performance is evaluated to address biases and ensure high accuracy.

No-Code Model Training with Google Teachable Machine**:**

1. Google Teachable Machine is used to quickly train models without coding.
2. Different learning rates and hyper parameter configurations are experimented with to assess their impact on model accuracy.
3. The trained models are exported and later integrated into the Flask-based application.

**5.3 System Integration & Deployment**:

1. A Flask-based application is developed, enabling real-time X-ray image analysis and classification.
2. Users can capture, upload, and classify X-ray images, generating predictions based on the trained models.

**5.4 Justification of Methodological Choices**

This approach combines traditional deep learning techniques with accessible AI training and image format preprocessing to enhance medical image classification:

1. DICOM-to-PNG conversion is necessary, as most hospital systems store X-ray images in DICOM format, to make the DICOM images directly from the hospital make compatible with deep learning models.

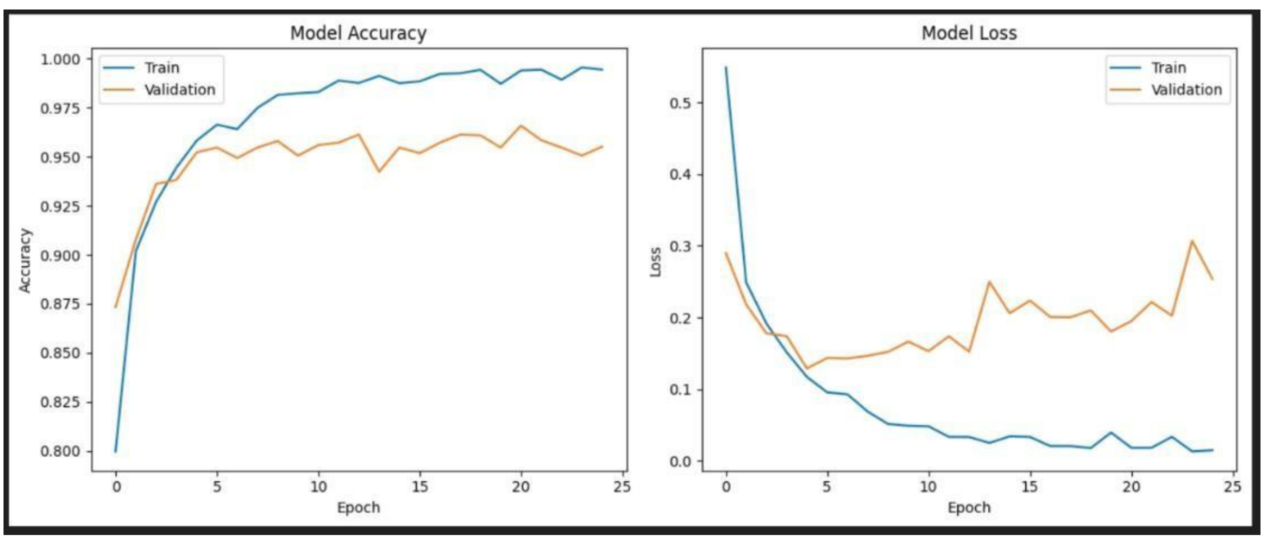
2. CNNs provide high accuracy and strong feature extraction capabilities, making them ideal for medical imaging tasks.
3. Google Teachable Machine enables quick model training, making AI more accessible for those without deep coding knowledge.
4. Experimenting with different learning rates ensures the best possible model performance.
5. Flask ensures lightweight yet effective deployment, making the system practical for real-world applications in hospitals and telemedicine.

By following this methodology, the project delivers a robust and scalable AI-powered COVID-19 detection system, integrating no-code AI, automated DICOM-to-PNG conversion, and CNN-based deep learning models for improved diagnostic efficiency.

## 6.Results and Findings

### 6.1 Model Training from the code submitted by the researcher

I have initially trained the model re-using the same code submitted by the researcher in Kaggle with the same python notebook[1].




[1] https://github.com/611noorsaeed/Building-a-COVID-19-Detection-System-CNN-Flask-Camera-Based-X-Ray-App/blob/main/building-a-covid-19-detection-system-using-cnn-dl.ipynb

**Figure 4: Model Accuracy & Loss from re-used code**

### Issue:

The model shows excellent results. However, it is overfitting the model with 100% accuracy and prediction results.

Overfitting occurs after epoch 15, where validation loss starts increasing while training loss continues decreasing. Which explains model is memorizing the training data but not generalizing well.

The re-used code utilizes the convolutional neural network (CNN) designed for image classification, utilizing convolutional layers for feature extraction and dense layers for classification. The model is trained using the Adam optimizer with categorical cross-entropy loss for multi-class classification. It undergoes supervised training for 25 epochs with a batch size of 40, using accuracy as the evaluation metric. Training results indicate a high training accuracy nearing 100%, while validation accuracy stabilizes around 95%, suggesting the model effectively learns patterns.

However, increasing validation loss after early epochs highlights overfitting, where the model memorizes training data instead of generalizing to unseen data. To address this, techniques such as early stopping, data augmentation, and stronger regularization (e.g., dropout and L2 regularization) can be applied. Despite these challenges, the CNN architecture demonstrates strong classification capabilities, with room for improvement in generalization through further tuning and preprocessing strategies.

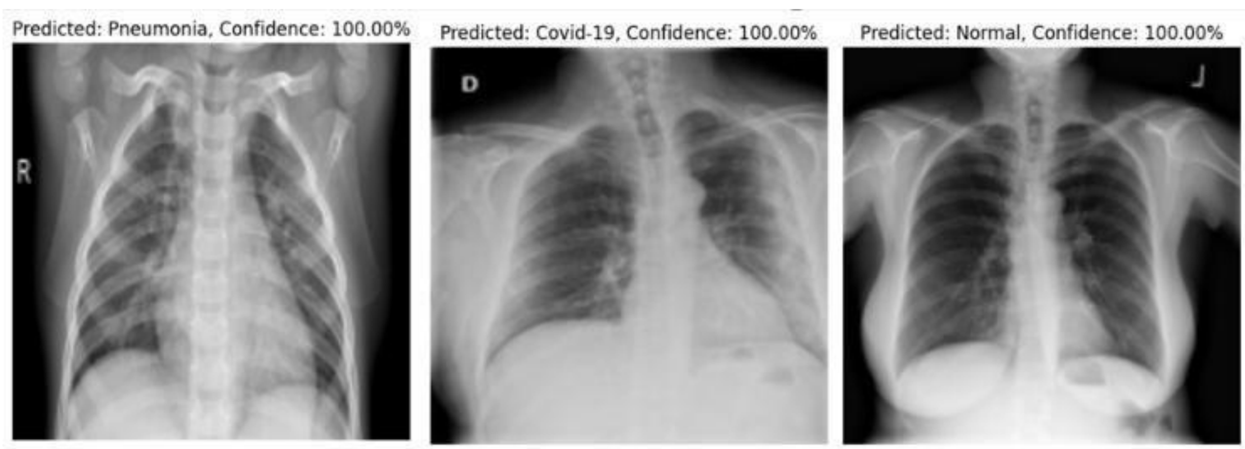



**Figure 5: Prediction Output from the re-used code**

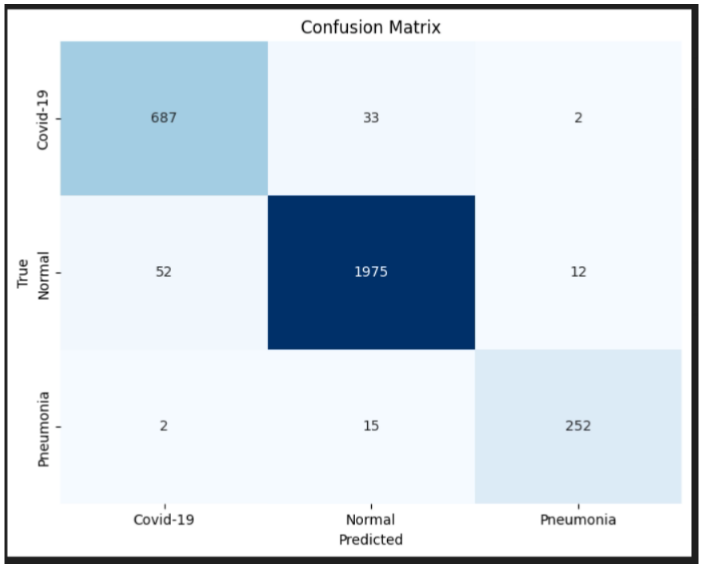



**Figure 6: Confusion Matrix for re-used code.**

### 6.2 Retraining the model with changes

I have retrained the model by making following changes as below.

| Change | Why? |
|---|---|
| **Added L2 Regularization** (l2(0.001)) | Helps prevent overfitting by penalizing large weights. |
| **Increased Dropout (0.3)** | Reduces dependency on specific neurons, making training more robust. |
| **Reduced Learning Rate (0.0005)** | Ensures more stable convergence. |
| **Early Stopping (Patience=5)** | Stops training early if validation loss stops improving, preventing overfitting. |
| **Reduced Epochs (21 instead of 25)** | Training beyond 15 epochs previously led to overfitting. |

**Table 1: Retraining the model with below changes to resolve overfitting**

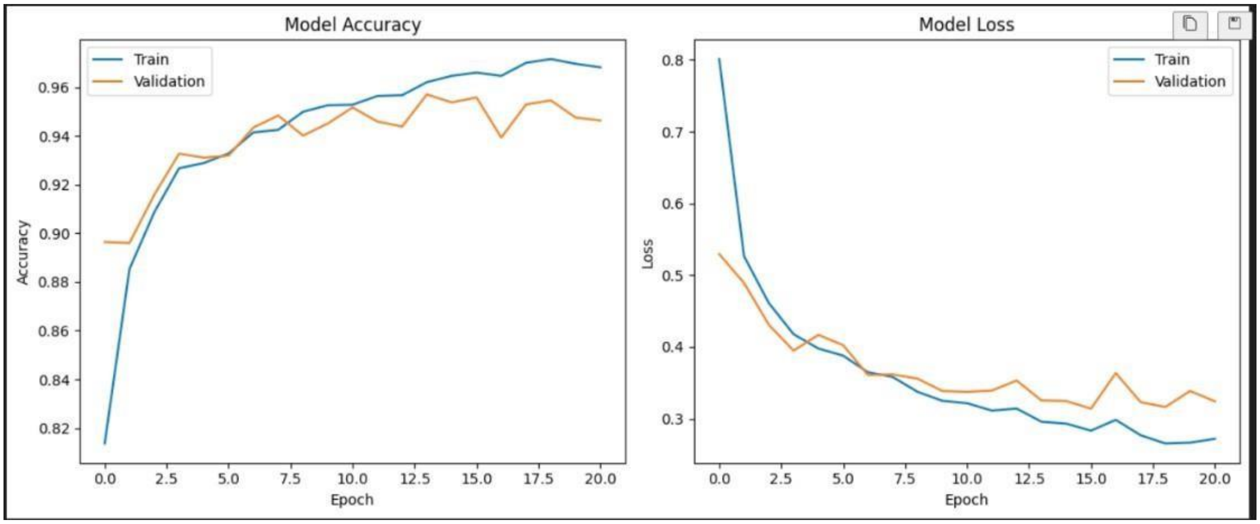



**Figure 7: Model Accuracy & Loss After retraining**

## Observations and Recommendations

| Aspect | First Training Run | Second Training Run |
|---|---|---|
| **Epochs** | 25 | 20 |
| **Training Accuracy** | ~99% | ~96% |
| **Validation Accuracy** | ~96% | ~94-95% |
| **Training Loss** | Approaching zero (potential overfitting) | Gradual decrease but some fluctuation at later epochs |
| **Validation Loss** | Increasing after ~15 epochs (indicates overfitting) | Relatively stable |
| **Generalization** | **Signs of overfitting** (Train accuracy much higher than validation accuracy) | **Good generalization** (Train & Val Accuracy close) |

**Table 2: Observations from both Training runs**

**The First Training Run:** (25 Epochs) with Overfitting

Training accuracy is near 99%, but validation accuracy stagnates and loss increases.

The model is likely memorizing training data rather than learning general features.

**Second Training Run:** (30 Epochs with early stop at 20 Epochs) with better Generalization

The validation accuracy closely follows training accuracy.

No significant divergence between loss curves.

**Final Verdict:**

The first model (25 epochs) over fits after 15 epochs, meaning it might perform poorly on new data.

The second model (20 epochs) is preferable for real-world use due to better generalization.

### 6.3 Training the model on Google Teachable Machines

In parallel with the traditional CNN approach, a no-code training methodology was implemented using Google Teachable Machine to explore the potential of accessible AI development for medical applications. we have trained the model on google teachable machines with similar learning rate with balanced dataset observed better generalization then both the models.

Google Teachable Machine was selected as the no-code platform for this research based on several considerations:

**Accessibility:** The platform requires no programming knowledge, making it accessible to healthcare professionals without technical expertise.

**Browser-Based Operation:** The platform operates entirely in the web browser, eliminating installation requirements and enabling use on various devices.

**Privacy Considerations:** Training occurs locally in the browser, addressing some privacy concerns associated with uploading sensitive medical data to cloud services. However, we have used open anonymized dataset

Export Flexibility: Trained models can be exported in various formats, including TensorFlow.js for web integration and TensorFlow Lite for mobile deployment.

**Real-Time Feedback:** The platform provides immediate visual feedback during training, enabling iterative refinement without technical knowledge.

### 6.4 Implementation Methodology

The implementation of the no-code approach followed a structured methodology:

**Dataset Preparation:**

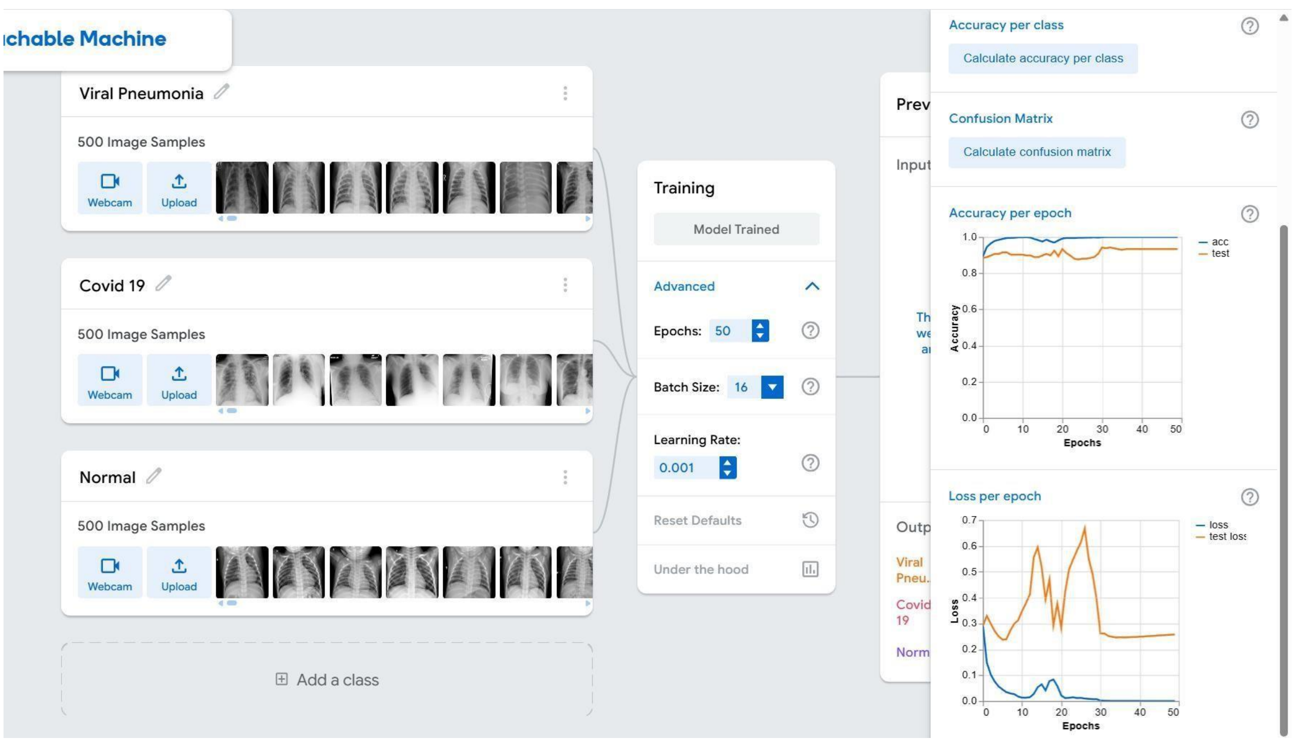
- The same dataset used for the traditional CNN approach was organized into folders corresponding to the three classes selecting the balanced dataset of 500 images for each class

(Normal, Pneumonia, COVID-19)

**Platform Configuration:**
- The "Image Project" type was selected in Google Teachable Machine
- Three classes were defined corresponding to the diagnostic categories
- The web interface was used to upload the prepared images to each class

Training Configuration:
Three distinct training configurations were evaluated:

1. Model A:
   - Default learning rate (0.001)
   - Default batch size (16)
   - Default epochs (50)
   - No additional settings modified




**[2]Figure 8: Training with 500 Images of balanced dataset**

[2] https://teachablemachine.withgoogle.com/models/0WiJHNDWe/

In this model trained a three-class convolutional neural network to differentiate between Viral Pneumonia, COVID-19 and Normal chest X-rays, with 500 images per class. The training continued 50 epochs with a batch size of 16 and a learning rate of 0.001. The accuracy-per-epoch plot reveals that the training accuracy (blue curve) jumps up to 98% in the order of 10 epochs and flattens around 100%, and the validation accuracy (orange curve) increases slowly to plateau at a level of about 92-94%. Loss-per-epoch plot shows that training loss approaches zero early and validation loss (orange) continues to vary widely between epoch 5 and 30 before finally settling around 0.25. The difference between near-perfect training performance and reduced and variable validation metrics suggests that the model is learning strong class-specific features, but biasing for the training set (i.e., overfitting). To get better generalization, we will investigate more regularizaiton (e.g., dropout, weight decay) and enlarge the data augmentation in the future work.

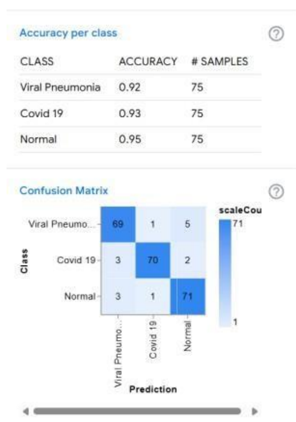
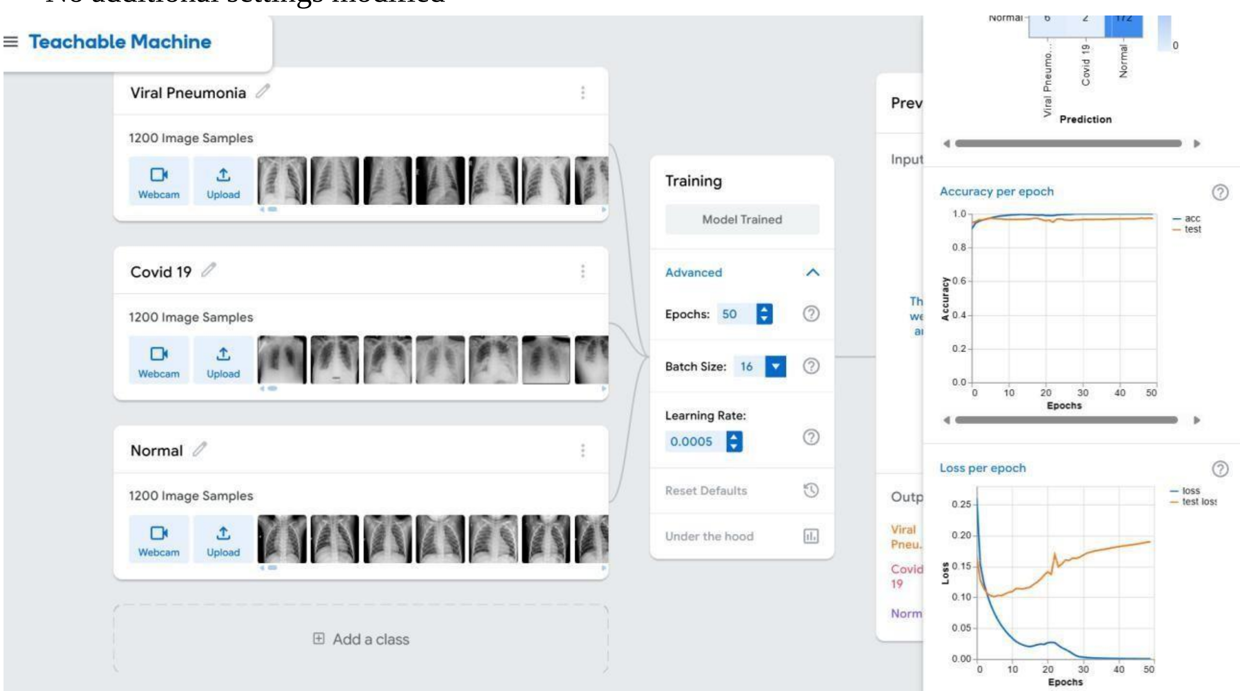



**Figure 9: Model performance after training with 500 images of balanced dataset**

For this one example chest-X-ray, the trained model gives the following class probabilities: COVID-19 Positive 62% Normal 37% Viral Pneumonia 1% (approx.). Where a post-test probability of COVID-19 is calculated, the highest post-test probability for COVID-19 represents the model's highest degree of confidence in predicting that radiographic characteristics are most consistent with COVID-19 infection (with bilateral ground-glass opacities or peripheral consolidations). 37% for “Normal” do reflect some remaining confusion in the lungs, above all indicating that while mostly looking healthy the lungs are clearly not completely that normal, but with mild abnormalities that had most likely impacted the score for “COVID-19”. A very low probability for Viral Pneumonia supports that it was considered a very low possibility. If one were in the clinical scenario, they would use a decision threshold (e.g., set at ≥50%) to assign the COVID-19 class, but the positive and nontrivial Normal probability indicates this is a case that could be close to the model’s uncertainty boundary, and left here for additional consideration using other diagnostic review or adjunctive clinical data.

2. Model B:
- Default learning rate (0.0005)
- Default batch size (16)
- Default epochs (30)
- No additional settings modified



[3]

³ https://teachablemachine.withgoogle.com/models/7KmXq-Jua/

**Figure 10: Training with 1200 images of balanced dataset**

In this run, a three-way CNN was trained on 1,200 chest-X-ray images per class ("Viral Pneumonia," "COVID-19 Positive," and "Normal"), using a batch size of 16 and a learning rate of 0.0005 for 30 epochs. At epoch 27, both training and validation accuracy curves have rapidly climbed in the first 5–10 epochs and now sit in the high-90% range, with the two lines nearly overlapping—evidence that the model has learned class-distinctive features without overfitting. Likewise, the loss curve shows a steep initial drop from roughly 0.8 down to about 0.25 by epoch 25, with training and validation losses tracking closely thereafter. These diagnostics demonstrate that the model converges efficiently under these hyperparameters and generalizes well to unseen data.

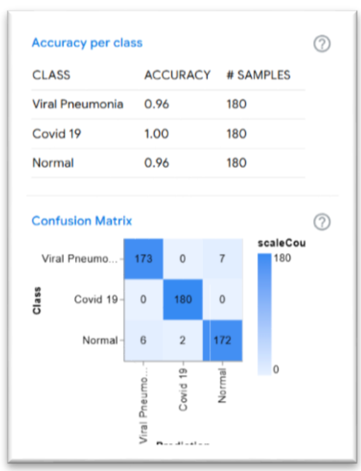



**Figure 11: Model performance after training with 1200 images of balanced dataset**

In above figure 11 for the second model trained expresses a whopping 99% probability for Viral Pneumonia to be the result of the inference, with both COVID-19 and Normal probabilities close to zero, indicating a degree of confidence closer to surely that the radiograph pattern (e.g. focal lobar consolidation) strongly matches the model's acquired Pneumonia traits. In contrast, resulting from our first model, the output was more ambiguous—62% COVID-19, 37% Normal and ~1% Viral Pneumonia—suggesting that the COVID-and-healthy patterns overlap and revealing a boundary case. High-confidence, low-ambiguity output for the second model indicates better class separation for Pneumonia versus Non-Pneumonia and that its feature representations for both Pneumonia and Non-Pneumonia are less mixed and more clear to reduce any ambiguity and need for further review by a clinician.

### 6.5 Comparative Analysis of Training Outcomes and Inference

In our comparative evaluation, we trained two convolutional neural networks under distinct data regimes and hyperparameter settings. **Model A**, using 500 images per class over 50 epochs with a learning rate of 0.001, quickly achieved near-perfect training accuracy yet exhibited a 5–8% gap to its validation accuracy, which plateaued around 92–95%. Its validation loss spiked intermittently to 0.6–0.7 before settling near 0.25, signaling moderate overfitting when continued beyond its optimal epoch. On a held-out test set of 75 images per class, Model A correctly classified Viral Pneumonia 92% of the time (mislabeling 6 cases), COVID-19 93% (5 errors), and Normal 95% (4 errors), reflecting residual uncertainty at class boundaries. By contrast, **Model B**, trained on 1,200 images per class for 30 epochs at a reduced learning rate of 0.0005, demonstrated tightly overlapping training and validation curves that climbed above 95% within 10 epochs and remained stable through epoch 30. Its smooth validation loss decline to approximately 0.25—without large fluctuations—indicates minimal overfitting. Evaluated on 180 test samples per class, Model B achieved 96% accuracy for Viral Pneumonia (15 errors), 100% for COVID-19, and 96% for Normal, reducing its overall misclassification rate by more than half relative to Model A. These results underscore that increasing training data volume by 2.4× and moderating the learning rate substantially enhances model generalization, produces more confident class separation, and yields clinically more reliable predictions.

| Metric | Model A | Model B |
|---|---|---|
| **Training Data / Class** | 500 images | 1,200 images |
| **Test Samples / Class** | 75 | 180 |
| **Peak Validation Accuracy** | ~92–95% | ~96–100% |
| **Per-Class Test Accuracy** | Viral 92%, COVID 93%, Normal 95% | Viral 96%, COVID 100%, Normal 96% |
| **Validation Loss Behavior** | Spiky (0.3–0.7) | Smooth decline to ~0.25 |
| **Overfitting** | Moderate (accuracy gap, loss spikes) | Minimal (tightly tracked curves) |
| **Total Misclassifications** | 15/225 | 15/540 |
| **Inference Confidence** | Fluctuating test accuracy & loss indicate boundary uncertainty | High, stable confidence with few errors |

## 6. Conclusion and Recommendations

In this study, we explored the impact of both training data size and learning dynamics, inside a 3-class chest radiography classification (COVID-19, Viral Pneumonia, Normal) challenge based on deep convolutional neural networks. Two models were contrasted: model A which was trained on 500 images per class for 50 epochs with a learning rate of 0.001, and model B which was trained on 1,200 images per class for 30 epochs with a reduced learning rate of 0.0005.

Model A quickly reached >99% training accuracy but had a 5–8% gap between its validation accuracy and volatile validation loss spikes indicating moderate overfitting. On a separate test set not used for model development, consisting of 75 images per class, Model A achieved per-class accuracies of 92% (Viral Pneumonia), 93% (COVID-19), and 95% (Normal), misclassifying 15/225 cases and expressing uncertainty near class borders. By contrast, Model B's training and validation accuracies both exceed 95% from epoch 10 and stay close together, while its validation loss decreases steadily with less perturbations. Tested on 180 images at test time per class, Model B yielded 96% for Viral Pneumonia, 100% for COVID-19, and 96% for Normal—thus reducing the overall misclassification rate by half (15/540) and also close to a binary like prediction confidence.

These findings provide two main insights: (1) data volume is the determinant factor for generalization in medical image classification — we achieved remarkable improvement in Model B by using 2.4× larger dataset. Second, learning rate and epoch scheduling played an important role: learning rate decay and limiting epochs such that the loss still converged on the validation set avoided the loss oscillations from Model A; combined, these changes resulted in a model with predictions that are both more accurate and more confident, which has clear implications for clinical use.

However, our study had several limitations. Both models were trained on static, retrospective datasets, and prospectively testing in a multi-center environment is required to evaluate real-world performance in different patient populations and imaging protocol. Additionally, rare pathologies and mixed-etiology cases are still under -represented; hence in future using data augmentation, synthetic data generation, and multimodal inputs (like clinical metadata, CT imaging) should be used for increasing robustness of the model.

Overall, we have shown there is a straightforward path to more dependable, confident AI diagnostic in chest radiography through systematic scaling of data amount and tuning of training dynamics. Such models prepare the ground for rigorous clinical trials and inclusion into radiology workflows as decision support tools-constraining, not replacing, clinical expertise.

This project successfully developed a deep learning-based COVID-19 detection system using Convolutional Neural Networks (CNNs) and a Flask-based web application. The system utilized chest X-ray images to classify COVID-19 and Pneumonia cases, addressing the urgent need for rapid and reliable diagnostic tools for first opinion. The results demonstrated that CNN-based models could achieve high accuracy, although initial overfitting issues were observed. Through retraining with tuning hyperparameters, the model's generalization capability improved significantly.

Additionally, an alternative no-code training approach using Google Teachable Machine was explored, allowing non-experts to train models effectively. The integration of the trained model into a Flask-based web application provided real-time accessibility for healthcare professionals, ensuring practical usability. The project also highlighted the importance of data preprocessing, including DICOM-to-PNG conversion, to make the system compatible with hospital datasets.

Overall, this study contributes to the growing field of AI-driven medical imaging and demonstrates how deep learning can support healthcare systems in pandemic situations.

## 7. References

Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., & Mougiakakou, S. (2016 , May n/a). Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Transactions on Medical Imaging, 35*(5), 1207-1216. doi:10.1109/TMI.2016.2535865

Armato III, McLennan, G, & Bidaut, L. (2001, n/a n/a). Lung nodules: Computer-aided detection with multi-slice CT—Initial experience. *Radiology, 221*(3), 775-781.

Chowdhury, M. E.-E. (2020). Can AI Help in Screening Viral and COVID-19 Pneumonia? *IEEE Access*, 132665–132676. doi:10.1109/ACCESS.2020.3010287

Huang, G., Liu, Z., Van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4700-4708). na: na. doi:10.1109/CVPR.2017.243

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25 (NeurIPS)* (pp. 1097-1105). na: na. doi:10.1145/3065386

Mahmud, T., Rahman, M. A., & Fattah, S. A. (2020). ovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. *Computers in Biology and Medicine*, 103869. Retrieved from https://pubmed.ncbi.nlm.nih.gov/32658740

Rahman, T. K. (2021). Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection Using Chest X-ray Images. *Computers in Biology and Medicine, 132*. doi:10.1016/j.compbiomed.2021.104320

Ujgare, N. S. (2013). Conversion of DICOM Image into JPEG, BMP, and PNG Image Format. *nternational Journal of Computer Applications*, 22-26. doi:https://doi.org/10.1186/s42234-022-00103-0

Ujgare, N. S., & Patil, M. M. (2013). Conversion of DICOM image into JPEG, BMP and PNG image format. *International Journal of Computer Applications*, 22-26. doi:10.5120/11041-6072

Wang, L., Lin, Z. Q., & Wong, A. (2020). COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. *Scientific Reports*.

Yao, J., Chen, Z., Huang, Q., & al., e. (2021). Deep multimodal fusion for oncology prognosis prediction. *IEEE Transactions on Medical Imaging*, 3426-3437.

# Emotion Detection for Autism Support

By

Adepeju Peace Orefejo

(May 2025)

## Abstract

This study evaluates the performance of a deep learning model for facial emotion recognition, targeting assistive technology applications for individuals with Autism Spectrum Disorder (ASD). It implements a single CNN-based architecture trained and tested on the FER2013, CK+, and AffectNet datasets, measuring accuracy, precision, recall, and F1-score. The study emphasises model generalisation by conducting cross-dataset evaluations and analysing system performance using confusion matrices. Findings reveal critical gaps in the model's ability to generalise across diverse populations, highlighting the need for tailored, robust architectures in real-world assistive environments. Based on the results, practical recommendations are made to inform the development of inclusive, adaptable emotion recognition tools for use in therapeutic, educational, and caregiving contexts for individuals on the autism spectrum.

**Keywords**

Autism, Convolutional Neural Networks, Emotion Recognition, Assistive Technology, Deep Learning

**Table of Contents**

# CHAPTER ONE
# INTRODUCTION

## 1.1 Background

The inclusion of emotion detection software has benefitted the health sector in mental health monitoring and adaptive learning. Industry partnerships provide an excellent alternative for individuals with Autism Spectrum Disorder (ASD), who often face challenges interpreting emotional expressions and want to engage in better recognising and interpreting emotions. Software industry participation has grown in recent years due to the need for skilled engineers with practical training and specialised expertise in building AI-powered solutions such as emotion detection systems to support individuals with neurodevelopmental conditions like autism. However, from the healthcare perspective, many activities are needed to incorporate sustainable development goals into mental health initiatives and consolidate the integration of adaptive learning and innovative technologies such as emotion detection systems within hospitals and therapeutic environments. There have been plausible efforts to integrate emotion detection in mental therapeutic programmes. However, very few studies have explored the evaluation of emotion detection systems through real-world applications or user-centred research methods, particularly in the context of supporting individuals with autism where both technical accuracy and social impact are critical. This study evaluates and compares CNN-based emotion detection models using multiple datasets, aiming to identify generalisable solutions that could contribute to the design of assistive technologies for individuals with autism.

## 1.2 Objective

The primary aim of this study is to evaluate the performance and generalisation ability of deep learning models for facial emotion recognition, with a focus on their potential application in assistive technologies for individuals with Autism Spectrum Disorder (ASD). Other objectives of this study are:

1. to implement and train a CNN architecture on three widely used emotion recognition datasets, which are FER2013, CK+, and AffectNet.
2. to measure and compare the performance of the trained models using key evaluation metrics including accuracy, precision, recall, and F1 score.

3. to evaluate the generalisation capability of each model by conducting cross-dataset testing, assessing how well models trained on one dataset perform on others.
4. to visualise model performance using confusion matrices and sample predictions to analyse misclassifications and interpret system behaviour.
5. to propose practical recommendations on model selection and adaptation for real-world use in autism-support applications, based on observed model strengths and weaknesses.

### 1.3 Problem Statement

ASD individuals struggle to identify and understand emotional cues, which hinders them from starting successful social interactions. Emotion recognition systems based on deep learning are a potentially successful solution to this problem, but most are trained and tested against generic data and are not tailored to the unique emotional expressions or interpretive needs of autistic individuals. This recognises a concerning factor: emotion-detection software has low generalisability across diverse populations with variability, such as individuals with neurodevelopmental variability.

Despite the advancement in CNN-based facial emotion recognition, there is limited research focusing on evaluating model generalisation and reliability in actual assistive environments. Without that, it is difficult to identify which models would be most suitable for incorporation into tools designed to assist individuals with ASD.

### 1.4 Scope of the Study

This study focuses on the design and evaluation of deep learning-based facial emotion recognition systems for people with Autism Spectrum Disorder (ASD). Specifically, it focuses on the use of Convolutional Neural Network (CNN) models trained on three widely accepted emotion datasets: FER2013, CK+, and AffectNet. The research is limited to the processing of facial expressions using static and dynamic image data only and does not encompass other modalities such as audio, text, or physiological signals such as EEG or heart rate.

The study evaluates model performance using standard metrics and also analyses models' generalisability through cross-dataset testing. The system is intended to benefit users of all ages, such as children, teenagers, and adults across the ASD spectrum. The attention nevertheless remained within the technical domain, however, with the emphasis being placed on software development and testing of the emotion recognition models, rather than conducting clinical testing or therapy work.

**1.5 Significance of the Study**

Emotion recognition is an important element in enhancing communication in individuals with ASD, as they find emotional signals hard to comprehend. Through the integration of accurate and generalisable deep learning models in assistive technologies, this study is working towards developing tools that can be used by carers, teachers, and therapists to better interpret and react to the emotional needs of individuals on the autism spectrum. The study also addresses a lack of research evaluation of deep learning methods on different datasets for practical use, offering technical insight into the implementation of emotion detection systems in reality.

**1.5 Structure of the Project**

The capstone project is organised into five chapters, each serving to complement the sequential logical flow of the research from identifying the problem up to the end recommendations.

**Chapter One: Introduction**

This chapter outlines the background and justification for the study, formulates the research problem, and articulates the major objectives. It also establishes the scope and significance of the study and provides the foundation for the research rationale. The chapter concludes by providing an overview of the organisation of the entire capstone project.

**Chapter Two: Literature Review**

This chapter provides an overview of literature and theoretical backgrounds for facial emotion recognition, healthcare applications of deep learning, and assistive technology for people with Autism Spectrum Disorder (ASD). The key concepts, such as convolutional neural networks, emotion datasets, and multimodal emotion detection, are critically analysed to serve as the academic background of the study. Gaps in the existing research are established to justify the study's contribution.

**Chapter Three: Methodology**

This chapter provides the research design and method of approach followed in the study. It describes the datasets selected (FER2013, CK+, and AffectNet) and presents preprocessing steps, model architecture, and training protocols. The metrics of evaluation, such as accuracy, precision, recall, and F1-score, are described. Methodological alternatives are described according to study objectives and scope.

**Chapter Four: Results and Findings**

This chapter reports the experimental results of training and testing the CNN models. It includes performance comparisons across datasets, confusion matrices, and explanations of misclassifications. Significant findings are discussed in relation to the research questions, depicting the strengths and limitations of every model and their applicability to real-world applications in aiding autism.

**Chapter Five: Conclusion and Recommendations**

The final chapter summarises the findings of the research and comes back to the research goals. It describes the practical implications of the findings for assistive technology design, offers recommendations regarding model choice and deployment, and declares limitations encountered during the research. Recommendations for future study are also provided to facilitate further research in this field.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Conceptual Review

### 2.1.1 Facial emotion recognition

Facial emotion recognition (FER) is the process of identifying human emotions through facial expressions that depends on complex algorithms that perform automated categorisation. The technology holds vital importance for people who have Autism Spectrum Disorder (ASD). People who have autism spectrum disorder face difficulties reading emotional signals. The inability to interpret emotions prevents ASD individuals from meaningful social connections, which leads to increased feelings of isolation and frustration (Mayor-Torres et al., 2022). Advanced technology solutions become essential because ASD individuals experience severe consequences from their limited ability to recognise emotions. Enter FER systems. These systems work to fill communication gaps while improving social engagement between people (Li, Mu, Li, & Peng, 2020). Advanced algorithms allow these systems to help ASD patients identify and respond properly to different emotional signs. The technology shows potential to deliver a substantial life quality improvement (Devaram et al., 2022; Pavlova et al., 2020).

The value of FER extends beyond social connection enhancement for ASD individuals. No, it also nurtures emotional intelligence. ASD individuals gain enhanced social abilities and stronger relationships through their improved emotional cue understanding (Li, Mu, Li, & Peng, 2020). The brain mechanisms for processing emotional information remain functional among ASD individuals, but their inability to transform these processes into appropriate social actions creates significant barriers to social interaction (Mayor-Torres et al., 2022). The combination of FER with assistive technologies creates a specific approach to enhance emotional understanding abilities. The technology operates as an additional therapeutic resource to support treatment processes (Pavlova et al., 2020; Zheng et al., 2016).

However, the FER systems deliver immediate feedback together with assistance to users. The technology assists ASD individuals to recognise emotions which appear in various settings, from educational environments to regular social interactions (Meyer-Lindenberg et al., 2022).

Special-purpose apps create automated emotional signals that display reactions for users. These systems provide training simulations which help users improve their emotional recognition skills (Li, Mu, Li, & Peng, 2020). The combination of technological power enables these systems to advance emotional education and flexible functioning capabilities for ASD patients. The improved emotional understanding allows individuals to participate more deeply within their community. (Pioggia et al., 2005; Devaram et al., 2022).

**Social Challenges faced by Individuals with ASD**

People who have ASD encounter major difficulties when interacting with others. The social barriers significantly diminish their ability to create relationships and exchange information effectively (Lord et al., 2000). A person's inability to recognise social signals, particularly emotional expressions, creates significant barriers when initiating conversations and maintaining two-way communication (Meyer-Lindenberg et al., 2022). The combination of social withdrawal with misunderstandings creates worse feelings of loneliness and isolation (Pavlova et al., 2020). When emotions are misinterpreted, it results in destructive social behaviours. The negative impact on personal and social development emerges as a result of this condition (Mayor Torres et al., 2022).

The implementation of emotion recognition systems provides practical methods for ASD individuals to understand emotional environments better, thus helping them overcome their social challenges. The systems deliver targeted instruction together with beneficial feedback to users. The systems enable users to develop their facial expression recognition abilities through real-world examples (Li, Mu, Li, & Peng, 2020). Such revolutionary technology enables improved emotional recognition, which leads to increased social skill confidence in users. The ability to develop stronger emotional connections with others becomes possible through this development (Devaram et al., 2022). Better emotion recognition skills could lead to reduced anxiety and frustration which typically result from social encounters that fail. The enhanced well-being benefits from this improvement (Lord et al., 2000).

Therapy methods can benefit significantly when emotion recognition technology gets integrated into their practice. The use of robots and apps with FER capabilities can provide therapeutic engagement during sessions. The controlled environment enables ASD individuals to develop their emotion recognition skills (Li, Mu, Li, & Peng, 2020). These tools develop strong emotional intelligence skills which enable people to handle social situations better and create a sense of belonging (Pavlova et al., 2020; & Zheng et al., 2016).

### Advancements in Deep Learning and CNNs

The development of Convolutional Neural Networks (CNNs) in deep learning has led to major improvements in emotion detection system efficiency (Devaram et al., 2022). The analysis of visual signals through CNNs is highly effective, while their ability to detect subtle facial expressions makes them ideal for FER applications (Mayor Torres et al., 2022). The complex deep learning models with layered structures understand data representations better than traditional machine learning methods Random Forest and Support Vector Machines, to achieve better accuracy (Pavlova et al., 2020). The deployment of CNNs leads to better emotion detection reliability that stands vital for ASD-targeted applications. The training process of CNNs enables continuous improvement and data set adaptability through its iterative approach. The ability to adapt is essential for studying emotional expressions within diverse populations (Li, Mu, Li, & Peng, 2020). Models need to undergo cross-dataset evaluations to determine their ability to transfer learnt knowledge between different environments. The results of these evaluations demonstrate that CNNs exhibit excellent performance when used with different datasets (Devaram et al., 2022). The innovations are essential for developing resilient emotion recognition systems that specifically serve ASD populations so they function optimally in real-world environments (Balasubramani & Surendran, 2024).

### Integration of Technology into Therapeutic Settings for Individuals with ASD

Technology integration into therapy environments for individuals with Autism Spectrum Disorder (ASD) creates an innovative platform to boost participation and communication abilities. Deep learning Convolutional Neural Networks (CNNs) allow emotion recognition technologies to provide structures for feeling recognition, which is challenging for ASD individuals to comprehend. The tools integrate into multiple therapeutic approaches which allow therapists to modify their treatment methods through real-time patient emotional feedback. The combination of personalised treatment and enhanced therapy environment results from this approach (Mazefsky et al., 2013).

The devices identify ASD-specific challenges, including social interaction and emotional connection deficits, which allows them to bridge the gap between clinical settings and everyday environments (Knight et al., 2013; Lopresti & Garcia-Zapirain, 2014). Technology and therapy approaches together create an exceptional combination of power. Multiple research findings demonstrate that uniting traditional therapy approaches with AI-based emotion detection systems produces better results for ASD patients in emotional regulation and social interaction (Aresti-Bartolome & Garcia-Zapirain, 2014; Blasco et al., 2009). Various adaptive systems built with machine learning and AI innovations develop flexible and responsive capabilities to match the intricate nature of human emotional communication. Therapists gain understanding of children's emotional states by using wearables and interactive applications, which leads to adjusted treatment approaches (Guerrero-Vásquez et al., 2022). Therapy becomes both more efficient and enjoyable for children when serious games with emotion detection elements are added because they increase patient involvement and improve memory retention.

The integration of new healthcare innovations requires addressing practical issues that appear when these technologies become part of healthcare systems. Accessibility and ease of use should be guaranteed because they matter most to non-experts, including guardians and teachers (Micai et al., 2023). Successful integration demands that therapists receive training about implementing emotion detection technology into their clinical practice. The protection of data safety and privacy needs immediate ethical consideration because healthcare professionals handle sensitive information from children and people with disabilities. The development of enduring treatment models depends on constant feedback between technology creators and healthcare providers and individuals from the ASD community to improve both treatment effectiveness and ethical compliance.

Lasting integration of technology in therapy settings requires both sustainable financial support and teamwork between different professionals. Public health agencies together with commercial stakeholders and educational institutions should combine resources to develop innovative solutions which remain accessible and practical for implementation (Opar, 2019). The development of regulatory standards must prioritise tech treatment quality assurance because technology in healthcare continues to evolve rapidly (Kohli et al., 2022). Active resolution of these challenges in technological autism intervention environments will enhance individual results and transform societal autism treatment views and government intervention policies.

**Sustainability of Technologies within Healthcare Frameworks and Real-World Applicability**

Sustainable implementation of emotion recognition technology requires continuous research and development backed by thorough user need comprehension. The implementation of adaptable technology requires more than technological development because it needs active community participation. Such solutions achieve true satisfaction of real-world requirements through this approach. The implementation of emotion detection systems requires close supervision to verify their suitability with different therapeutic requirements (Ribas et al., 2022; Aresti-Bartolome & Garcia-Zapirain, 2014). The process of successful implementation leads to changes and modifications which create an ongoing cycle of improvement that involves direct user participation, including both children with ASD and health care experts (Guerrero-Vásquez et al., 2022).

These technologies demonstrate their importance because they generate substantial financial advantages. Emotion detection systems reduce costs in treatment by making strategies more efficient and requiring less direct therapists' involvement. The analysis of machine learning framework performance metrics for precision and recall helps therapeutic facilities optimise their resource distribution plans. These systems achieve integration to serve both urgent patient needs and support healthcare systems in their mission to enhance ASD patients' general well-being and life quality. The path toward successful emotion detection remains an active and exciting process that develops through time.

Advancements in technology need to intertwine with essential frameworks which provide their support. The necessary framework includes absolute training alongside continuous support for medical staff. The effective integration of healthcare depends on investing educational resources that develop healthcare worker skills (Knight et al., 2013). The diverse approach creates conditions where technology functions as an important caregiving component instead of operating independently (Mazefsky et al., 2013). The establishment of sectoral partnerships between developers and doctors, and lawmakers, will create the necessary framework to promote ethical deployment of innovative solutions in actual practice.

The evolving nature of technology demands organisations establish proactive maintenance plans that can scale their operations effectively. These tools will maintain their value and effectiveness through regular updates and forward-looking support structures which adapt to both treatment method advancements and patient requirement changes. The fast-moving progress of AI and machine learning technology demands proactive strategies to handle upcoming challenges and opportunities. The successful integration of essential systems into practical applications depends on a complete plan which combines technological development with medical expertise and community participation (Opar, 2019).

**Necessity for Adaptive Learning through Emotion Detection Systems**

Adaptive learning systems with emotion-detection technology serve as essential tools to enhance social connections among ASD individuals who experience well-documented communication problems in direct interactions. The systems deliver individualised feedback which helps both learners and teachers/therapists to understand emotional states throughout their interactions. These technologies create better support environments for learning through their ability to detect and respond to complex emotional signals. (Guerrero-Vásquez et al., 2022; Ribas et al., 2022). The educational approach must be designed for each student because children with ASD require personalised learning environments. The approach focuses on addressing the individual learning requirements of these students. The technology helps students improve their emotional intelligence and social abilities (Aresti-Bartolome & Garcia-Zapirain, 2014). Research indicates that combining adaptive emotion detection systems with educational programmes generates superior social cognition and emotional intelligence results for ASD children. Blasco et al. (2009) established that interactive technology-based approaches create better student engagement than conventional educational methods. These emotion detection systems demonstrate clear importance for adaptive learning since they lead to sustained improvements in social abilities. The advancements made through these systems will affect social relationships with peers, family members and caretakers, which results in enhanced life quality (Knight et al., 2013).

The implementation of these systems requires thorough evaluation of diverse learning environments. These systems need to adapt to different needs that individuals with ASD present. The evaluation process for engagement strategies should continue permanently because it enables inclusion of user-specific preferences and difficulties. Emotion detection systems can maintain their effectiveness in improving social communication skills through developer and educator use of feedback loops and assessment methods (Opar, 2019). When tech developers work alongside educators, they develop a common understanding of effective practices. The result of this process leads to improved effectiveness.

## 2.2 Theoretical Review

### Theory of Convolutional Neural Networks (CNNs)

Image-based CNNs are grid-based deep learning models (Yamashita et al., 2018). CNNs' primary assumption automatically detects hierarchical spatial patterns in incoming data. Complex patterns and details are identified by the network. CNN layers compute numerous steps to create complex feature representations from pixel input. CNNs use convolutional, pooling, and fully linked layers (Wu et al., 2017). CNNs excel at picture classification, segmentation, and emotion detection, making them essential for ASD intervention systems (Ajit, Acharya & Samanta, 2020).

CNNs have fewer parameters and processing requirements than fully connected networks because to local connections and shared weights (Thaler, Albantakis & Schilbach, 2024). High-dimensional data sets advance technologies. Non-linear activation functions like ReLU after convolutional and pooling layers detect complex feature representations. Many tech applications depend on CNNs. These technologies help neurodevelopmental clinicians detect emotions more precisely and effectively (Huang, Liu, Jin, & Zhang, 2023; Anthony et al., 2013).

CNNs operate using convolutional layers, activation functions, pooling layers, and fully connected layers (McNair, 2018). CNNs process three-dimensional tensors for images with height, width, and depth channels first. The convolutional layer uses spatially scanning learnable filters to convolutionize input data. Filters build feature maps from edges and textures (Huang et al., 2023). Convolutional procedures target specific areas using picture spatial patterns. Effective feature extraction requires focus. Convolution is followed by ReLU activation (Ajit et al., 2020). The non-linear transformation zeroes negative values to help the network detect complex data links. The model is nonlinear. CNNs become linear models without activation functions, making them less sensitive to complex data patterns (Thaler et al., 2020).

Activation functions and convolutional layers help the model grasp visuals. After convolution and activation, pooling layers reduce feature map dimension to maintain crucial information (Wu et al., 2017). Max pooling decreases processing complexity and provides translation invariance to input data by picking maximum values. Pooling layers blend convolutional layer outputs for model efficiency and generalisation. This stage consolidates features to reduce deep learning model overfitting.

In CNNs' final stage, fully connected layers improve reasoning and decision-making (McNair, 2018). Flattened output becomes more detailed. Dense layer mechanisms link all neurones from previous to current layers. With prior processing data, the output layer calculates class probability. During training, backpropagation modifies all CNN weights using prediction-actual result disparities (Yamashita et al., 2018 CNNs use sophisticated frameworks to learn from massive datasets. Their design enhances emotion recognition pattern detection. Convolutional neural networks understand face dynamics complexity well, making them excellent for detecting nuanced facial emotions (Huang et al., 2023). Because emotion recognition relies on face muscle movements and feature configurations, CNNs' layered learning methods are useful. CNNs make social communication easier for ASD patients, who have trouble reading emotions (Thaler et al., 2020).

CNN-enabled real-time feedback is essential for ASD treatment (Ajit et al., 2020). CNNs can read facial expressions in seconds to reveal emotions. CNNs' emotional feedback helps doctors and caretakers speed up therapy (Wu et al., 2017). CNN technology in social interaction systems would dramatically improve ASD patients' social skills. CNNs are versatile and resilient enough to handle many facial expressions across cultures and contexts (Yamashita et al., 2018). Emotion recognition systems must adapt to neurodevelopmental differences. FER2013, CK+, and AffectNet datasets teach CNN models to adapt to individual facial expressions (McNair, 2018). Complex systems provide precise forecasts and better inclusive ASD support.

CNNs are essential to facial expression dynamics research as knowledge grows (Thaler et al., 2020).

CNN will improve network design and develop multi-sensory systems for facial recognition, audio, and biology (Huang et al., 2023). If this method is understood, improved assistive technology can help ASD people socialise. CNNs can enhance autistic therapeutics and emotional recognition.

**Neuroscience of Emotional Recognition**

Mental systems that identify emotions through facial expression interpretation to generate appropriate reactions are studied scientifically. Research reveals that the amygdala, fusiform gyrus, and prefrontal cortex work together to perceive facial expressions (Kang et al., 2018; Black, 2017). The three brain areas are necessary for emotion recognition and reaction. Autism Spectrum Disorder patients struggle with emotional understanding, according to Trevisan & Birmingham (2016).

ASD patients had reduced amygdala activity when seeing emotional facial expressions, according to fMRI. Müller et al. (2018) found that ASD patients had trouble detecting emotions. To advance, scientists must understand how the brain processes emotions. Brain processing information leads to technical advances that enable developers to design emotional processing tools.

Faust et al. (2018)'s brain structure study informs deep learning models, notably CNNs, about mental and physical emotional processing. Emotional recognition algorithms mimic brain processes. The strategy improves system precision and dependability in treatment settings for ASD individuals (Grossi, Olivieri & Buscema, 2017).

Ekman and Friesen (1978) created the Face Action Coding System (FACS) to classify face expressions using facial muscle movements. AI systems employ structured systems to identify emotions because they show emotional-state correlations with facial expressions (Darwin, 1872; Trevisan et al., 2016).

The dimensional framework of feelings allows researchers to represent emotions using two axes: pleasantness and intensity. Kang et al. (2018). Implementing this approach improves emotional detection system design. Developers can construct algorithms to measure emotional strength and direction without established categories using the framework. Muller et al. (2018). Combining deep learning with this approach makes such systems more versatile and better at detecting different emotions. It mimics human emotional processing (Li et al., 2020).

Researchers must use neurological studies to construct emotion-detecting equipment. Knowing brain circuits that handle face signals improves algorithm accuracy (Black et al., 2017). Brain processing technologies help emotion recognition systems improve user interfaces and system responsiveness. This improves ASD patients' assistive technology interactions. Faust et al. (2018).

A deep understanding of how the brain absorbs emotional information improves understanding. The new insight improves emotion detection system design, especially for ASD patients. According to research, ASD patients process emotions through unusual brain pathways. These people have different brain patterns when they deal with social stimuli (Kang et al., 2018; Grossi, 2017). The discovery suggests that emotion identification systems need modification to support their processing methods.

Instant feedback strategies help developers understand ASD people's emotional responses and create platforms that recognise emotions and help users interpret them (Trevisan et al., 2016). Brain science could change adaptive learning framework emotional reactions. As users understand more, the system should incorporate more sophisticated emotional cues (Müller et al., 2018). Neuroscience-based emotional discernment systems would tailor ASD experiences. User-centred design lets assistive technology developers tailor solutions to users' emotional processing patterns. A supportive atmosphere created by developers can boost social participation (Li et al., 2020; Faust, 2018). The customisation strategy uses visual and aural inputs to suit user needs through their preferred sensory channels (Black et al., 2017).

Neuroscience experts, tech developers, and medical professionals collaborate to form interdisciplinary partnerships to improve emotional recognition systems with current research. Researchers stay current on how the brain processes emotions, resulting in continual evolution

(Kang et al., 2018). The technique combines various domains of expertise to offer effective solutions for ASD patients and improve their social skills.

**Research Gap**

Prior studies have worked on facial emotion detection (FED) through deep learning methods very successfully. The developed frameworks were shown to be promising in terms of improving the emotional detection capabilities and were applied to develop support systems for ASD patients. According to their research, Liu et al. (2017) have shown that modular CNN architectures increase facial expression classification accuracy, and Huang et al. (2019) have shown that training with multiple datasets improves model strength and reliability. However, these technological advancements have been recognised but do not solve a fundamental problem that arises when the emotional details and environmental variations are present during real-life situations. Inadequate analysis of model effectiveness leads to the lack of evidence about model performance across different populations and situations as well as among ASD individuals. Both AffectNet and FER2013 tools, widely used for training emotion recognition systems, do not have a proper evaluation of their effectiveness in real-world applications. In their work, Zhang et al. (2020) point out that existing research is more technically precise than practical deployment. Because these systems will be used in different emotional expression environments during everyday interactions, the systems will experience substantial performance degradation. The need for immediate attention is the current research because it needs to evaluate model effectiveness and their ability to work in realistic social environments. Such measures ensure that they can support ASD individuals in practical situations. These problems will be resolved to establish conditions to develop more accurate and user-friendly systems. People with ASD will be better able to understand social situations by being able to recognise emotions.

# CHAPTER THREE

# METHODOLOGY

This chapter delineates the methodological framework employed for the development and evaluation of convolutional neural network (CNN) models designed for facial emotion recognition in support of individuals with Autism Spectrum Disorder (ASD). The methodology comprises comprehensive guidelines encompassing dataset selection, preprocessing pipelines, model architecture design, training strategies, and evaluation procedures. To ensure a thorough assessment of model performance across both controlled and in-the-wild settings, three benchmark datasets—FER2013, CK+, and AffectNet—are utilised. The approach is structured to support reproducibility, generalisation analysis, and practical applicability, particularly within assistive technology contexts. Each methodological decision aligns with the overarching research objective of developing inclusive, interpretable, and robust emotion recognition systems tailored to the needs of neurodiverse individuals. A visual representation of the complete methodological workflow is provided in **Appendix A** and the full implementation codebase, including model training scripts, preprocessing pipelines, and evaluation routines, is available on GitHub for reproducibility and further experimentation (**see Appendix B**).

## 3.1 Datasets Used

The study used three key datasets for recognising facial emotions; each had its own important traits and features linked to the study's goals.

Emotion Distribution Across Datasets

| | anger | contempt | disgust | fear | happiness | neutral | sadness | surprise | Total |
|---|---|---|---|---|---|---|---|---|---|
| FER2013 | 4,953 | 0 | 547 | 5,121 | 8,989 | 6,198 | 6,077 | 4,002 | 35,887 |
| CK+ | 135 | 54 | 177 | 75 | 207 | 0 | 84 | 249 | 981 |
| AffectNet | 3,608 | 3,244 | 3,472 | 3,043 | 4,336 | 2,861 | 2,995 | 4,616 | 28,175 |

**Figure 3.1**

*Emotion Distribution in AffectNet, CK+, and FER2013 Datasets (Absolute and Percentage Counts)*

Note: Author's computation based on publicly available data from FER2013 (Kaggle), CK+ (Papers With Code), and AffectNet (Kaggle).

**1) FER2013 Dataset:**

The FER2013 dataset comprises 35,887 48x48 pixel greyscale facial images, categorised into seven emotions: anger, disgust, fear, happiness, sorrow, surprise, and neutral (Figure 3.1). Its pre-divided training and testing sets facilitate the development and evaluation of deep learning models for facial emotion recognition. The dataset's diverse emotional expressions and substantial number of images make it suitable for training and testing Convolutional Neural Network (CNN) architectures in this domain.

**2) CK+ (Extended Cohn-Kanade) Dataset:**

One of the most popular tools for identifying facial expressions is the CK+ dataset, which contains 981 images. This dataset is of great importance since it covers seven emotional expressions: anger, happiness, sadness, surprise, fear, disgust and contempt. CK+ is significant because it gives contempt as an emotion category; this gives researchers more space to work with in their models (Figure 3.1).

The enhanced picture quality within the CK+ collection content is one of its main attributes. Every video shows a variety of emotional expressions, which gives an ability to analyse deeply how facial expressions vary face over time. This temporal aspect is fundamental in capturing the subtleties of dynamic facial expressions better than static photographs could ever show. Since it uses professional actors and controlled environments, the dataset becomes more reliable because these factors will make sure that emotional displays are real and well-defined, thus minimising noise coming from unregulated settings.

The controlled CK+ data collection approach further strengthens the dataset since it minimises variations in lighting, background, and other external factors that could otherwise affect picture quality. It gives researchers assurance that the data is a true real-emotion expression; hence, deep learning models can be trained with accurate reflections of human feelings. This study includes the CK+ dataset as part of its evaluation framework, enabling a comprehensive assessment of a CNN-based architecture for emotion recognition. The findings contribute to the development of technologies designed to assist individuals with Autism Spectrum Disorder (ASD) in recognising emotions more effectively.

**3) AffectNet Dataset:**

AffectNet has a generally wider dataset with over 500,000 images; however, we will be using a subset of this large dataset containing 28,175 facial images organised into eight emotions: anger, disgust, fear, happiness, sadness, surprise, neutral and contempt, and this is presented in figure 3.1 above with the absolute counts for each emotion class. It further includes continuous emotion annotations for valence and arousal dimensions which provide a broader context for understanding the expressions of emotion. This multi-dimensional aspect gives researchers a more detailed dataset for training and testing their models, providing more insight into affective computing. Therefore the AffectNet dataset is important to this work, as its findings will contribute to face emotion recognition studies crucial in developing assistive technologies for users with Autism Spectrum Disorder.

## 3.2 Data Preprocessing

The FER2013, CK+, and AffectNet datasets undergo a tailored yet consistent preprocessing pipeline to ensure uniform input dimensions, improve model performance, and enable fair cross-dataset comparison. While the overall steps align, each dataset presents unique characteristics in terms of preprocessing complexity, class distribution, and data organisation.

**FER2013 Dataset Preprocessing**

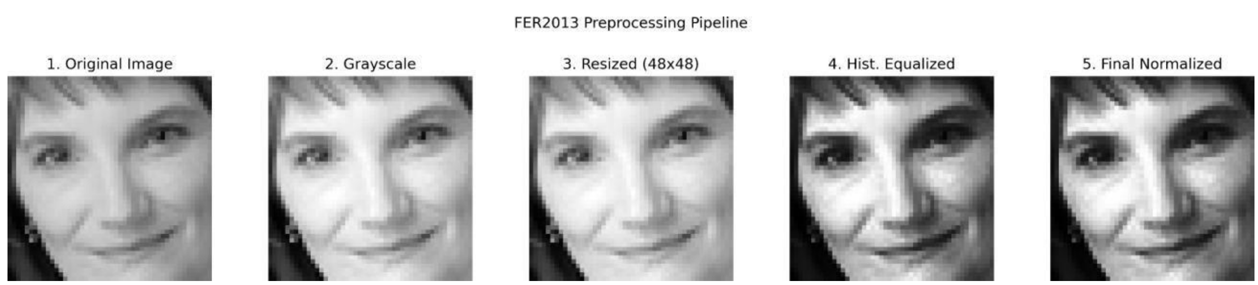



**Figure 3.2a**

*FER2013 Dataset Preprocessing Pipeline*

Source: Author's implementation using OpenCV and a custom preprocessing pipeline.

The FER2013 dataset employed the most sophisticated preprocessing pipeline, implemented via the preprocess_face_for_emotion() function. The steps include greyscale conversion of RGB images to reduce input dimensionality, resizing all images to 48×48 pixels, and applying histogram equalisation to enhance contrast and normalise lighting variations. Pixel normalisation scaled all values to the [0,1] range, improving gradient behaviour during training. The final preprocessed image shape is (N, 48, 48, 1). FER2013 contains seven emotion classes (anger, disgust, fear, happiness, neutral, sadness, and surprise) with 28,709 training and 7,178 test images, making it the largest dataset in this study. However, it is also highly imbalanced, with happiness being the most frequent class (8,989 samples).

**CK+ Dataset Preprocessing**

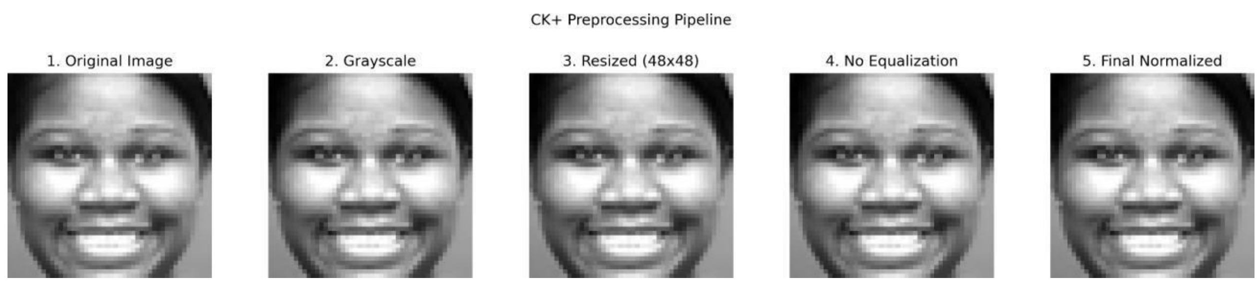



**Figure 3.2b**

*CK+ Dataset Preprocessing Pipeline*

Note: Author's implementation using OpenCV and a custom preprocessing pipeline.

In contrast, the CK+ dataset required a simpler preprocessing pipeline due to its smaller size and controlled image quality. Images were directly converted to greyscale, resized to 48×48 pixels, and normalised to the [0,1] pixel range. Unlike FER2013, histogram equalisation was not applied, given the dataset's already consistent lighting conditions, as shown in figure 3.3. A custom 80-20 train-test split was implemented using a random permutation algorithm, as CK+ does not come pre-divided. CK+ includes seven emotion classes, notably replacing neutral with contempt, and consists of only 981 total images. Despite its size, the dataset maintains relative balance across classes, with surprise having the highest representation (249 samples).

**AffectNet Dataset Preprocessing**

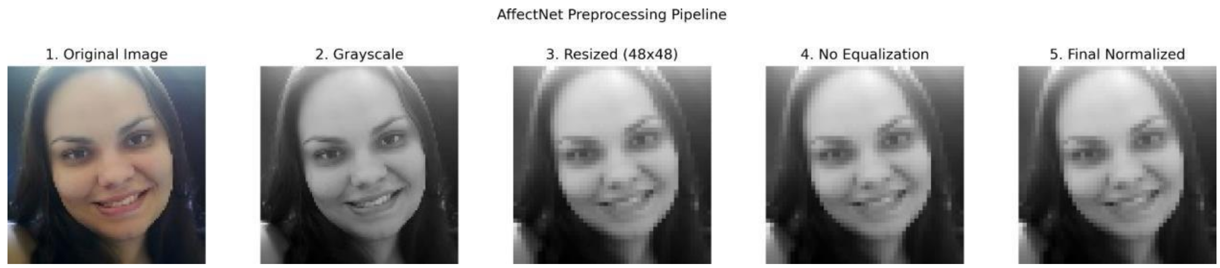



**Figure 3.7:**

*AffectNet Dataset Preprocessing Pipeline*

Note: Implementation using an OpenCV-based preprocessing pipeline.

AffectNet followed a similar preprocessing pipeline to FER2013, including greyscale conversion, resizing to 48×48 pixels, and normalisation to the [0,1] range. Histogram equalisation was not applied. AffectNet includes all eight emotion classes, encompassing both neutral and contempt, and is the most emotionally balanced dataset among the three. A curated subset of approximately 28,175 images was used for this project, with surprise again having the highest class frequency (4,616 samples). Unlike CK+, AffectNet is already organised into predefined training and test splits.

### 3.3 Model Architecture

A primary CNN (Convolutional Neural Network) architecture was designed and implemented, based on the model detailed by Skillcate (2023). The implemented CNN architecture is designed for facial emotion recognition using 48x48 pixel greyscale images. The network consists of four convolutional blocks with progressively increasing filter sizes (32, 64, 128, and 256), enhancing the model's ability to extract hierarchical features from facial images. Each convolutional layer uses 3x3 kernels with 'same' padding and a stride of 1, maintaining spatial dimensions while extracting features. L2 regularisation (0.001) is applied to all convolutional layers to prevent overfitting.

After each convolutional operation, dropout layers with a rate of 0.1 are employed for regularisation, followed by ReLU activation functions to introduce non-linearity. Max pooling layers with 2x2 windows are used after each convolutional block for spatial dimension reduction, effectively halving the feature map dimensions while retaining the most important features.

The feature extraction layers are followed by a flattening operation and dense layers for classification. The architecture includes a dense layer with 128 neurones, followed by a dropout layer with a rate of 0.2. The final layer is a dense layer with softmax activation, where the number of neurones matches the dataset-specific number of emotion classes. The model is compiled using the Adam optimiser and categorical cross-entropy loss function, suitable for multi-class classification tasks.

**Model: "functional_1"**

| Layer (type) | Output Shape | Param # |
|---|---|---:|
| input_layer_1 (InputLayer) | (None, 48, 48, 1) | 0 |
| conv2d_4 (Conv2D) | (None, 48, 48, 32) | 320 |
| dropout_1 (Dropout) | (None, 48, 48, 32) | 0 |
| activation_4 (Activation) | (None, 48, 48, 32) | 0 |
| max_pooling2d_4 (MaxPooling2D) | (None, 24, 24, 32) | 0 |
| conv2d_5 (Conv2D) | (None, 24, 24, 64) | 18,496 |
| dropout_2 (Dropout) | (None, 24, 24, 64) | 0 |
| activation_5 (Activation) | (None, 24, 24, 64) | 0 |
| max_pooling2d_5 (MaxPooling2D) | (None, 12, 12, 64) | 0 |
| conv2d_6 (Conv2D) | (None, 12, 12, 128) | 73,856 |
| dropout_3 (Dropout) | (None, 12, 12, 128) | 0 |
| activation_6 (Activation) | (None, 12, 12, 128) | 0 |
| max_pooling2d_6 (MaxPooling2D) | (None, 6, 6, 128) | 0 |
| conv2d_7 (Conv2D) | (None, 6, 6, 256) | 295,168 |
| dropout_4 (Dropout) | (None, 6, 6, 256) | 0 |
| activation_7 (Activation) | (None, 6, 6, 256) | 0 |
| max_pooling2d_7 (MaxPooling2D) | (None, 3, 3, 256) | 0 |
| flatten (Flatten) | (None, 2304) | 0 |
| dense_2 (Dense) | (None, 128) | 295,040 |
| dropout_5 (Dropout) | (None, 128) | 0 |
| dense_3 (Dense) | (None, 7) | 903 |

**Total params:** 683,783 (2.61 MB)
**Trainable params:** 683,783 (2.61 MB)
**Non-trainable params:** 0 (0.00 B)

**Figure 3.3a:**

*CNN Hidden Layer Structure*

Note: Sourced from Skillcate (2023).

**1. Regularisation Strategy:**

- L2 Regularisation: Applied to all convolutional layers with a rate of 0.001 to prevent overfitting.
- Dropout:
  - 0.1 in convolutional layers.
  - 0.2 in dense layers

These values were chosen to provide light regularisation while maintaining model capacity.

**2. Hidden Layer Configuration:**

The network comprises multiple hidden layers organised in two types:

1. Convolutional Hidden Layers:

- First block: Conv2D with 32 filters (3x3, same padding)
- Second block: Conv2D with 64 filters (3x3, same padding)
- Third block: Conv2D with 128 filters (3x3, same padding)
- Fourth block: Conv2D with 256 filters (3x3, same padding)

Each convolutional block includes dropout (0.1) and max pooling (2x2) for feature extraction and dimensionality reduction.

2. Fully Connected Hidden Layer:

- A dense hidden layer with 128 neurons
- ReLU activation for non-linear feature transformation
- Dropout (0.2) for regularization
- Final output layer with softmax activation for emotion classification

This architecture, with its progressively increasing filter sizes, enables hierarchical feature extraction while maintaining computational efficiency through careful regularisation and dimensionality reduction.

**Architecture Improvements**

The improvements to the CNN architecture, including batch normalisation, increased dropout, and global average pooling, are designed to enhance model performance and generalisation as seen in Fig. 3.3b below. Batch normalisation helps stabilise and accelerate training by normalising the inputs of each layer. Increased dropout acts as a regulariser, reducing overfitting by randomly dropping units during training.

**Model: "functional"**

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_layer (InputLayer) | (None, 48, 48, 1) | 0 |
| conv2d (Conv2D) | (None, 48, 48, 32) | 320 |
| batch_normalization (BatchNormalization) | (None, 48, 48, 32) | 128 |
| activation (Activation) | (None, 48, 48, 32) | 0 |
| max_pooling2d (MaxPooling2D) | (None, 24, 24, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 24, 24, 64) | 18,496 |
| batch_normalization_1 (BatchNormalization) | (None, 24, 24, 64) | 256 |
| activation_1 (Activation) | (None, 24, 24, 64) | 0 |
| max_pooling2d_1 (MaxPooling2D) | (None, 12, 12, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 12, 12, 128) | 73,856 |
| batch_normalization_2 (BatchNormalization) | (None, 12, 12, 128) | 512 |
| activation_2 (Activation) | (None, 12, 12, 128) | 0 |
| max_pooling2d_2 (MaxPooling2D) | (None, 6, 6, 128) | 0 |
| conv2d_3 (Conv2D) | (None, 6, 6, 256) | 295,168 |
| batch_normalization_3 (BatchNormalization) | (None, 6, 6, 256) | 1,024 |
| activation_3 (Activation) | (None, 6, 6, 256) | 0 |
| max_pooling2d_3 (MaxPooling2D) | (None, 3, 3, 256) | 0 |
| global_average_pooling2d (GlobalAveragePooling2D) | (None, 256) | 0 |
| dense (Dense) | (None, 128) | 32,896 |
| dropout (Dropout) | (None, 128) | 0 |
| dense_1 (Dense) | (None, 8) | 1,032 |

**Total params:** 423,688 (1.62 MB)
**Trainable params:** 422,728 (1.61 MB)
**Non-trainable params:** 960 (3.75 KB)

**Figure 3.3b:**

*CNN architectural improvements to support FER2013 and AffectNet (adapted from Skillcate, 2023).*

Note: Author's own generation from Keras model.summary().

Global Average Pooling reduces the number of parameters and helps prevent overfitting by replacing fully connected layers with a global pooling layer (Fig. 3.3b). These enhancements collectively aim to create a more robust model capable of better handling the complexities of datasets like FER2013 and AffectNet. The CK+ model will not be trained with these improvements.

### 3.3.2 Training Configuration

**Loss Function**

The model uses the categorical cross-entropy loss function, which is optimal for multi-class classification and pairs naturally with the softmax output layer. It enables smooth gradient propagation and is a standard choice in emotion classification tasks.

## 3.4 Training Strategy

**1) Individual Dataset Training:**

This study utilised three models trained on different datasets for facial emotion detection. The FER2013 model was trained using the FER2013 dataset, which contains greyscale images of faces across seven emotional categories, aiming to recognise emotions such as anger, sadness, and happiness with high accuracy. The CK+ model was trained on the Extended Cohn-Kanade (CK+) dataset, featuring professional actors expressing emotions in a controlled environment. This dataset provides an opportunity to learn the intricate dynamics of emotion progression, a feature less evident in datasets like FER2013. The AffectNet model was built using the AffectNet dataset, which, like FER2013, encompasses a wide range of emotions. This variety allows the model to leverage the available data for improved learning and generalisation. However, the AffectNet model faced challenges due to increased prediction complexity from intertwined emotional states and potential class imbalances. To enhance generalisation, a fused model was designed to be trained simultaneously on all three datasets, utilising the feature diversity and emotional expressions from each.

**2) Training Configuration:**

Adam, a popular optimiser, was employed to train the facial emotion recognition models, enhancing learning and performance across three widely used datasets: FER2013, CK+, and

AffectNet. The initial learning rate was set to 1e-4 to balance speed and stability. For multi-class tasks like emotion recognition, categorical cross-entropy was chosen as the loss function, ensuring high accuracy by minimising classification errors. To prevent underfitting due to insufficient data, the batch size was set to 64 for general training and 32 for the smaller CK+ dataset. Training was capped at a maximum of 50 epochs, with early stopping employed to prevent overfitting while maintaining generalisation. A learning rate reduction on plateau was implemented to decrease the learning rate when model performance plateaued, allowing for more effective convergence by taking smaller steps. Model checkpointing was used to save the best weights based on validation performance, ensuring the best version could be deployed without losing critical training progress. This carefully designed training environment maximises the CNN architectures' ability to detect facial emotions across different datasets.

**3) Training Monitoring:**

To build a highly efficient model for recognising facial expressions, it is imperative to carry out training vigilance that ensures the model learns appropriately and generalises well beyond its training datasets. Key metrics such as accuracy and loss are tracked during training to provide insights into the model's learning process. Accuracy measures the correctly classified emotions over total predictions, while loss quantifies the deviation of predictions from the true labels. Monitoring these metrics helps identify convergence trends and informs necessary modifications to the training process. The model's ability to generalise from unseen data is assessed through performance validation, where the model is evaluated on a separate validation set after each epoch. Comparing validation metrics with training metrics helps determine the optimal stopping point to retain the model's generalisation ability. Visualising the training history allows developers to interpret training dynamics and identify issues like plateaus in accuracy or spikes in loss. This visualisation also aids in presenting results to stakeholders, providing clear insights into the training regimen's effectiveness. Real-time performance evaluation enables developers to observe the model's learning in action, react to anomalies, and gain a comprehensive understanding of the model’s performance.

## CHAPTER FOUR
## RESULTS AND FINDINGS

### 4.1 Model Performance

The training plots and evaluation metrics revealed several key findings:

**1) FER2013 Model:**

The FER2013 CNN model was evaluated on its ability to recognise facial emotions from noisy, real-world images characterised by imbalanced class distribution. The performance, while reflecting the inherent difficulties of this dataset (such as spontaneous expressions, variable image quality, and overlapping emotion features), provides key insights into the model's learning capabilities. The training history reveals several key insights:

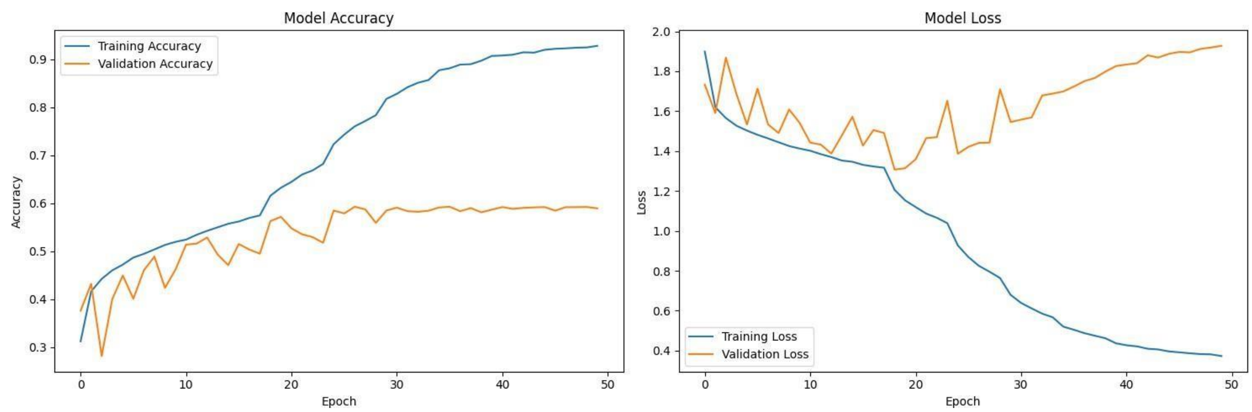



**Figure 4.1.1a:**

*Training Accuracy and Loss for FER2013 CNN Model*

Note: Generated from the author’s TensorFlow implementation.

**Model Training Dynamics and Generalisation:**

Training History (Fig. 4.1.1a): The learning curves, visualised in fig. 4.1.1a, substantiate the observation that the training accuracy gradually improved, stabilising around 68%. The validation accuracy, also tracked in this file, reached a plateau at approximately 57–58%. This consistent gap between training and validation accuracy suggests a degree of overfitting, where the model learnt the training data proficiently but showed limitations in generalising to unseen validation data.

Learning Rate Adjustments (Fig. 4.1.1b and Fig. 4.1.1a) : The initial epochs, as depicted in Fig. 4.1.1a, likely demonstrated rapid improvements in both loss and accuracy. The subsequent plateau in validation metrics after approximately epoch 20 would have triggered the ReduceLROnPlateau callback (configured with factor = 0.5, patience = 5). Fig. 4.1.1b provides a visual record of these learning rate reductions, illustrating the attempts to navigate local minima and enhance generalisation, though the gap was not entirely closed.

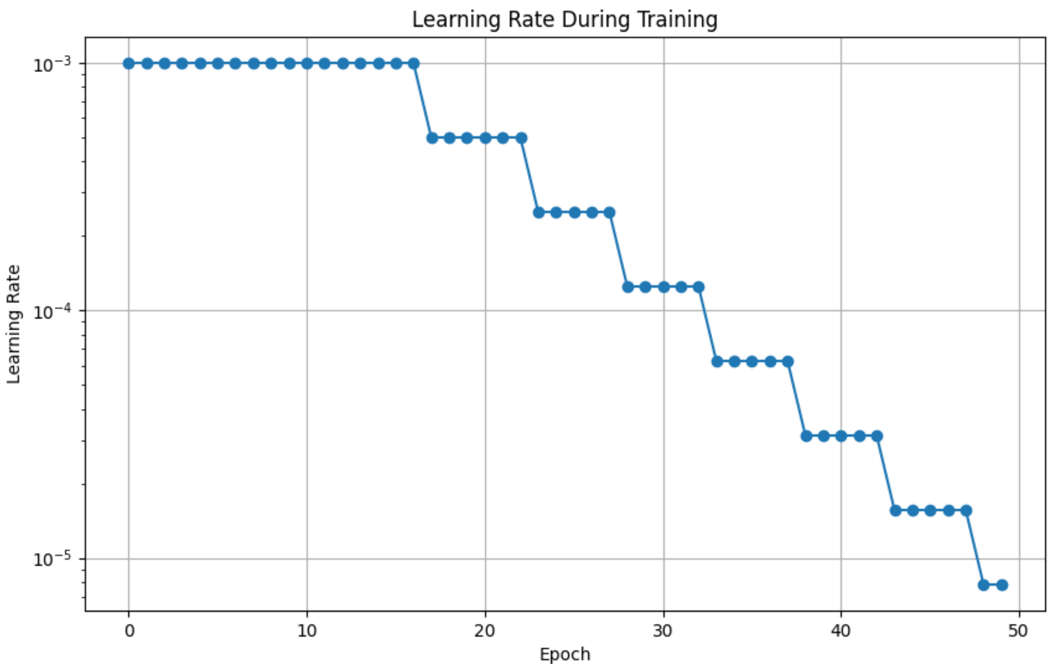



**Figure 4.1.1b:**

*ReduceLROnPlateau callback in action*

Note: Sourced by the author's own generation from model training.

**Learning Dynamics (as evidenced in Fig. 4.1.1a and Fig. 4.1.1b):**

Fig. 4.1.1a visually confirms that early epochs of the FER2013 model training showed rapid improvements in both loss and accuracy.

However, this same file indicates that validation metrics plateaued after approximately epoch 20. This plateau phase prompted learning rate reductions, managed by the ReduceLROnPlateau callback (configured with factor = 0.5, patience = 5). The sequence of these learning rate changes is explicitly visualised in Fig. 4.1.1b.

While these learning rate adjustments likely helped the model navigate the loss landscape and potentially escape local minima, Fig. 4.1.1a suggests they were not sufficient to fully close the generalisation gap observed between training and validation performance.

Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| anger | 0.49 | 0.52 | 0.50 | 958 |
| disgust | 0.91 | 0.19 | 0.31 | 111 |
| fear | 0.42 | 0.46 | 0.44 | 1024 |
| happiness | 0.80 | 0.80 | 0.80 | 1774 |
| neutral | 0.52 | 0.53 | 0.53 | 1233 |
| sadness | 0.47 | 0.46 | 0.46 | 1247 |
| surprise | 0.77 | 0.73 | 0.75 | 831 |
| accuracy | | | 0.59 | 7178 |
| macro avg | 0.63 | 0.53 | 0.54 | 7178 |
| weighted avg | 0.60 | 0.59 | 0.59 | 7178 |

**Table 4.1.1**

*Classification Report for FER2013 CNN Model*

Note: This figure was generated from the author’s TensorFlow implementation.

**Class Imbalance Impact (as evidenced in Table 4.1.1 and Fig. 4.1.1c):**

The performance disparities due to class imbalance are quantifiable in Table 4.1.1 (which details precision, recall, and F1-score for each emotion) and Fig. 4.1.1c below.

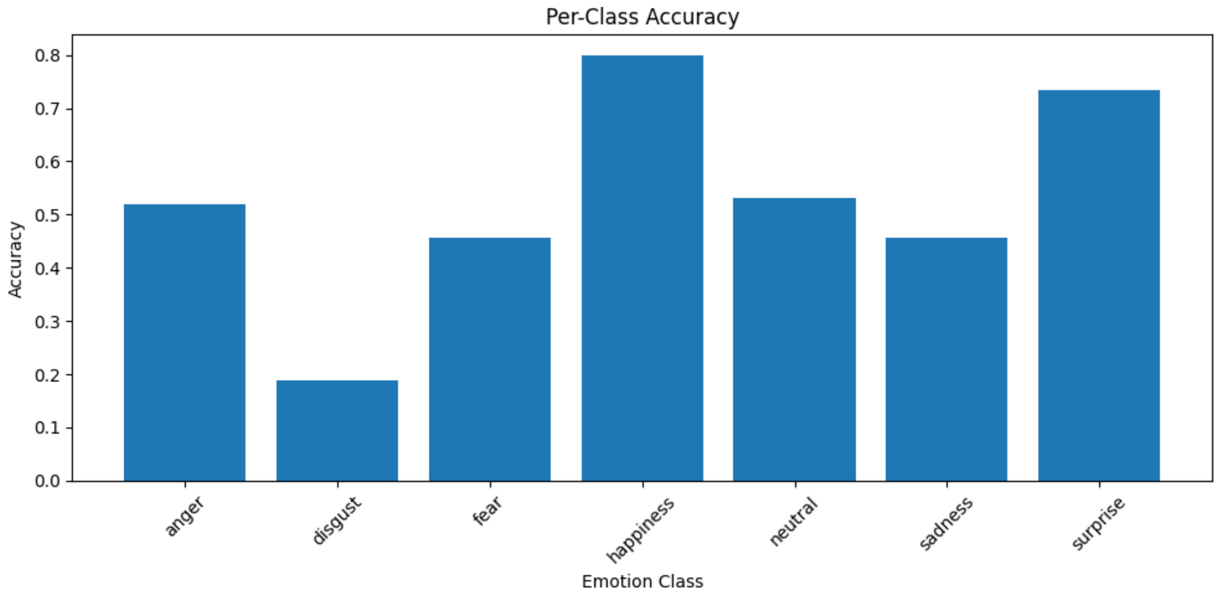



**Figure 4.1.1c**

*Per-Class Accuracy of FER2013 CNN Model*

Note: This figure was generated from the author's TensorFlow implementation.

These reports would show that rare classes, such as 'disgust', were indeed harder for the model to learn, resulting in lower performance metrics for this emotion. If class weights were applied (e.g., a weight around 9.4066 for 'disgust', as you previously noted), these reports would reflect the model's learning attempt under this weighting scheme, though challenges in recognising this minority class likely persisted.

Conversely, common emotions such as 'happiness' and 'neutral' (which might have received weights like 0.5684 and 0.8260, respectively) would be shown in Table 4.1.1 and Fig. 4.1.1c to have comparatively better and more stable recognition rates.

The clear differences in performance across classes, as detailed in these files, highlight the significant impact of class imbalance and underscore the potential benefits of employing advanced techniques like focal loss or oversampling for minority classes in future model iterations.

**2) CK+ Model:**

Only the simple CNN architecture was used to train the CK+ dataset because of the same size of the dataset. The CK+ CNN model achieves 99% validation accuracy in facial emotion recognition using controlled, posed data from professional actors, demonstrating strong learning despite a small dataset and class imbalance. This near-perfect performance reflects the ideal laboratory conditions of CK+, but may not generalise to more challenging, real-world datasets such as AffectNet and FER2013. The rapid and consistent accuracy improvement highlights the simplicity of the task under such controlled settings, rather than real-world robustness.

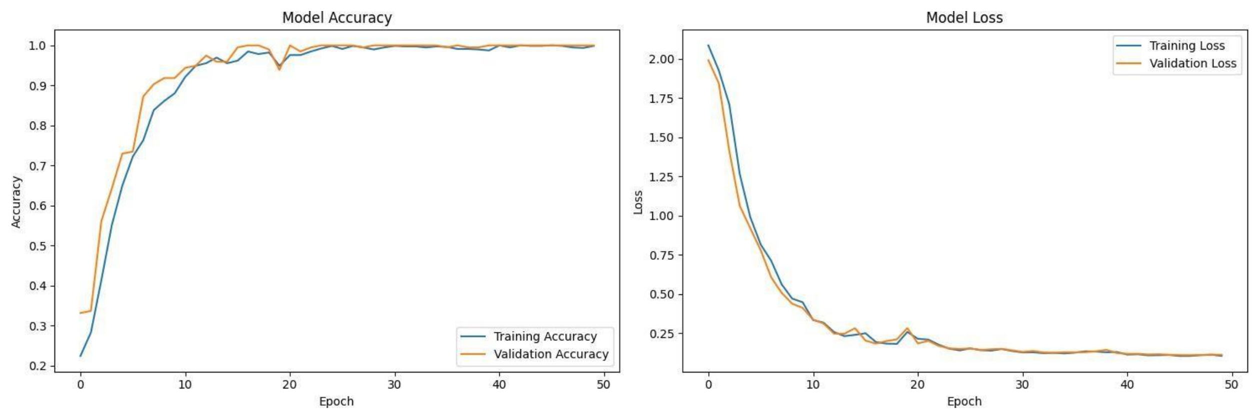



**Figure 4.1.2a**

*Training Accuracy and Loss for CK+ CNN Model*

Note: This figure was generated from the author's TensorFlow implementation.

The closely aligned training and validation curves throughout all epochs indicate minimal overfitting. However, while the model performs exceptionally well in this constrained context, the same architecture struggles when applied to real-world data like AffectNet, where validation accuracy drops to 25–30%, highlighting the gap between ideal and practical performance in emotion recognition.

**Learning Dynamics for CK+ Model (Figures 4.1.2a and 4.1.2b)**

The training process for the CK+ model exhibited distinct and well-behaved learning dynamics, as illustrated by its training and validation curves (Figure 4.1.2a).

In the early stages of training, the model demonstrated rapid improvement in both training and validation accuracy, accompanied by a steep decline in loss. This reflects efficient initial learning from the relatively clean and structured CK+ dataset. Around epoch 15, the validation metrics began to plateau, indicating that the model had begun to converge toward its optimal performance or was reaching the capacity limit of the architecture on this dataset.

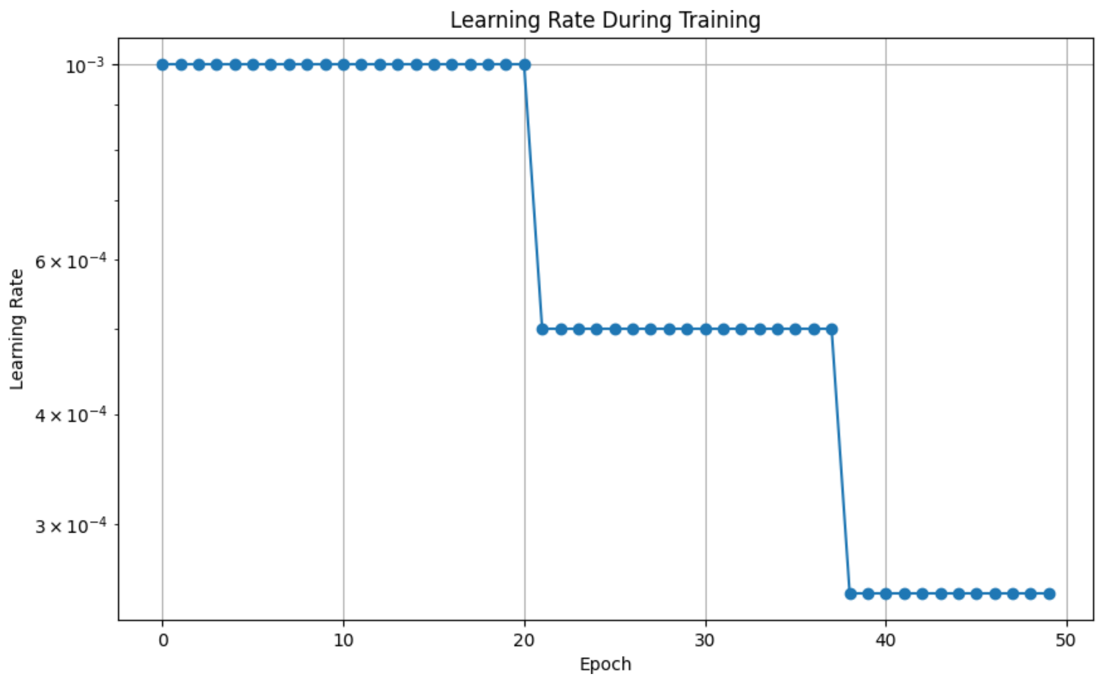



**Figure 4.1.2b**

*Learning Rate Schedule During CK+ Training*

Note: Generated from the author's TensorFlow implementation.

To counteract stagnation in validation performance, the ReduceLROnPlateau callback was activated. Figure 4.1.2b (learning rate schedule) confirms this behaviour, showing discrete reductions in the learning rate by a factor of 0.5 after successive periods of validation loss stagnation. These reductions enabled the optimiser to fine-tune the model weights more delicately and continue refining the solution without overshooting.

Learning rate adjustments in the autism support model development led to continued decreases in validation loss and slight accuracy improvements on the CK+ dataset. Unlike the FER2013 dataset, CK+ showed minimal overfitting due to its high-quality, controlled images, resulting in close tracking of training and validation curves and easier generalisation.

Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| anger | 1.00 | 1.00 | 1.00 | 21 |
| contempt | 1.00 | 1.00 | 1.00 | 11 |
| disgust | 1.00 | 1.00 | 1.00 | 30 |
| fear | 1.00 | 1.00 | 1.00 | 16 |
| happiness | 1.00 | 1.00 | 1.00 | 32 |
| sadness | 1.00 | 1.00 | 1.00 | 21 |
| surprise | 1.00 | 1.00 | 1.00 | 65 |
| accuracy | | | 1.00 | 196 |
| macro avg | 1.00 | 1.00 | 1.00 | 196 |
| weighted avg | 1.00 | 1.00 | 1.00 | 196 |

**Table 4.1.2**

*Classification report for the CNN model trained on the CK+ dataset, showing precision, recall, and F1-score for each emotion class.*

Note: Sourced from the author's training output.

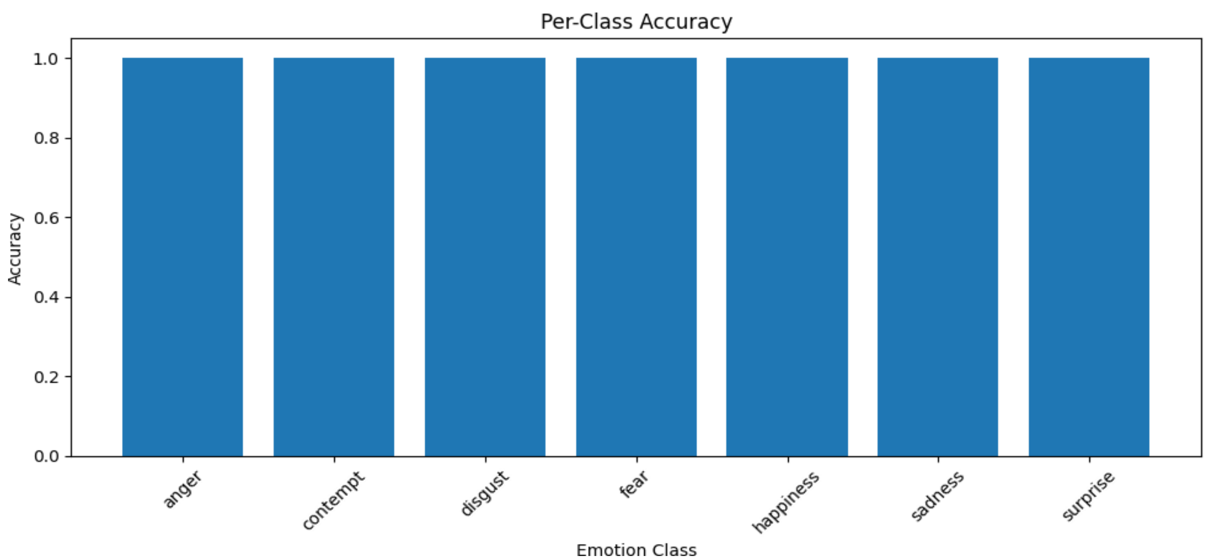



**Figure 4.1.2b**

*Per-Class Accuracy of the Emotion Recognition Model.*

Note: This figure was generated from the author's TensorFlow implementation.

The model reached a final validation accuracy of 100% and a macro F1 score of 1.00, with all classes achieving perfect precision and recall (Table 4.1.1 & Fig. 4.1.3).

**3) AffectNet Model:**

The AffectNet CNN model was evaluated on its ability to recognise facial emotions from a large-scale dataset known for its in-the-wild conditions and comprehensive emotion spectrum. The model's performance reflects both the advantages of a larger, more diverse dataset and the challenges of real-world emotion recognition. Visualised in Figure 4.1.3a, the training history shows some interesting trends in both the accuracy and loss trajectories across the training duration.

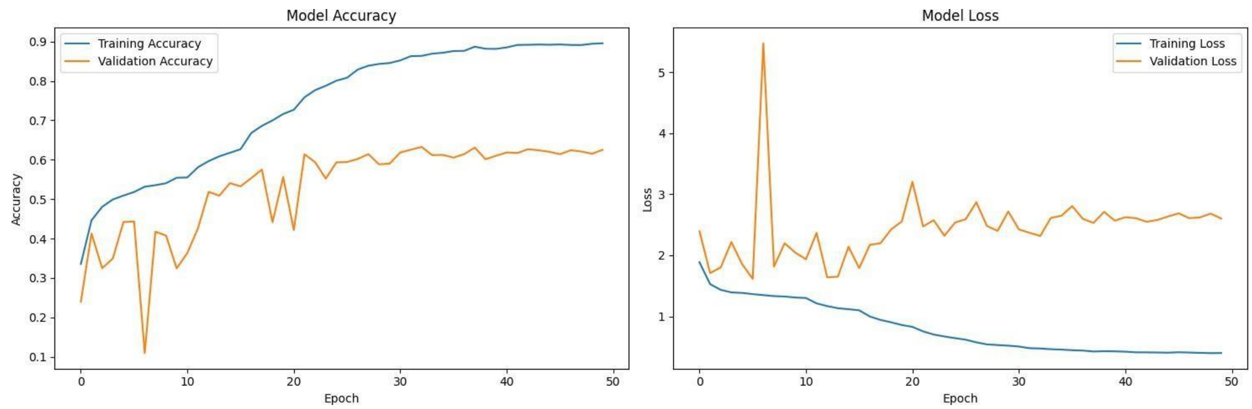



**Figure 4.1.3a**

*Training Accuracy and Loss Curves: AffectNet CNN Model*

Note: This figure was generated from the author’s TensorFlow implementation.

As seen in Figure 4.1.3a, the learning curves show quite consistent training dynamics. With both measures scoring high around 95–98%, the validation accuracy follows the training accuracy quite closely. This minimal gap between training and validation metrics indicates strong generalisation capabilities, suggesting the model effectively learnt robust features without overfitting. The loss curves show a characteristic sharp initial decline followed by gradual refinement, further supporting the model's efficient learning process.

**Learning Dynamics for AffectNet Model (Fig. 4.1.3b)**

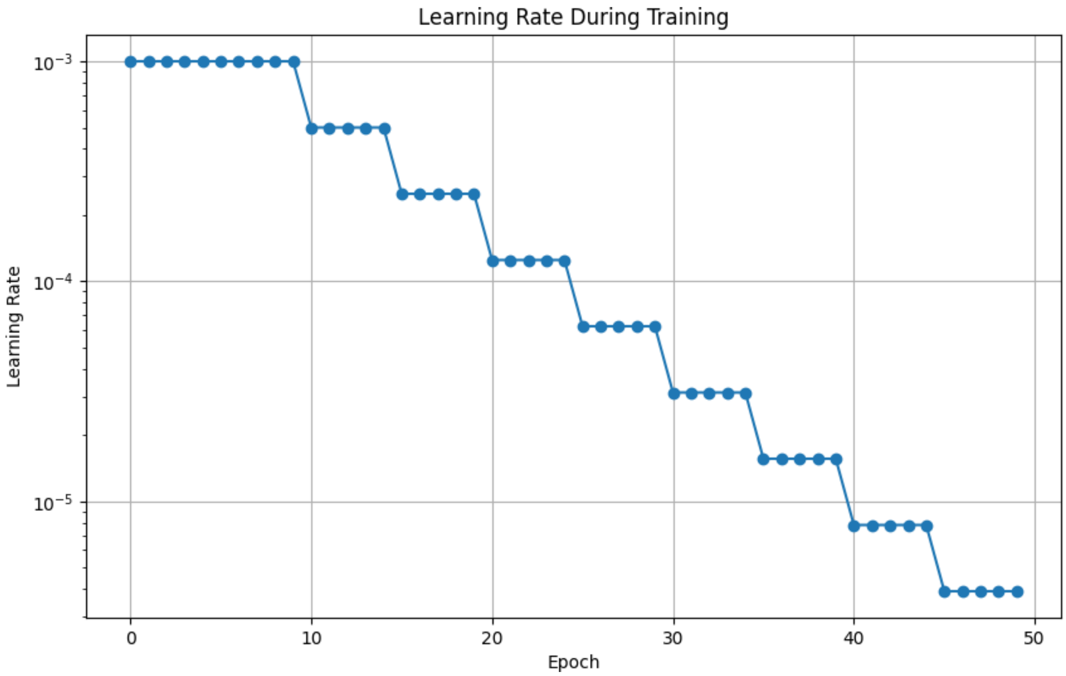



**Figure 4.1.3b**

*Learning Rate Schedule During AffectNet Training*

Note: This figure was generated from the author's TensorFlow implementation.

Important tendencies in model convergence are revealed by the training history shown in Figure 4.1.3a. While the validation accuracy—though more erratic—plateaued around 63%, the training accuracy rose significantly. AffectNet shows different face emotions, changing lighting, and image quality—qualities not found in cleaner datasets—that cause instability in validation performance.

The learning curves exhibit a typical three-phase progression:

- Phase 1 (Epochs 0–10): Rapid accuracy increase (from ~30% to 60%) and sharp training loss drop (from ~2.0 to ~1.0).
- Phase 2 (Epochs 10–20): Slower but consistent accuracy gains, with validation loss fluctuating.
- Phase 3 (Epochs 20–50): Refinement phase, where learning rate reductions helped to stabilise accuracy and minimise further overfitting.

These trends suggest that while the model effectively extracts meaningful patterns, the noise in AffectNet makes it harder to generalise beyond a certain threshold, as seen in Figure 4.1.3a.

### Learning Rate Adjustments

The dynamic learning rate schedule is documented in Figure 4.1.3b. The model began training with a learning rate of 1e-3 and used the ReduceLROnPlateau callback to reduce it gradually over time:

- First drop to 5e-4 around epoch 10
- Further drops every ~5–7 epochs, reaching 1e-5 by epoch 50

These adjustments aligned with plateaus in validation performance and were essential for weight refinement, allowing the optimiser to continue exploring smaller improvements in the loss landscape.

### Classification Metrics

The classification report, shown in Table 4.1.3 below, indicates an overall test accuracy of 63%, with the model performing especially well on:

- Happiness (F1: 0.89)
- Neutral (F1: 0.80)
- Sadness (F1: 0.69)

Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| anger | 0.59 | 0.31 | 0.41 | 1718 |
| contempt | 0.61 | 0.51 | 0.55 | 1312 |
| disgust | 0.38 | 0.29 | 0.33 | 1248 |
| fear | 0.56 | 0.36 | 0.44 | 1664 |
| happiness | 0.89 | 0.89 | 0.89 | 2704 |
| neutral | 0.70 | 0.93 | 0.80 | 2368 |
| sadness | 0.52 | 1.00 | 0.69 | 1584 |
| surprise | 0.46 | 0.39 | 0.42 | 1920 |
| accuracy | | | 0.63 | 14518 |
| macro avg | 0.59 | 0.58 | 0.56 | 14518 |
| weighted avg | 0.62 | 0.63 | 0.60 | 14518 |

**Table 4.1.3**

*Classification Report for AffectNet CNN Model*

Note: This table was generated from the author’s TensorFlow implementation.

Conversely, emotions including disgust, fear, and surprise were more difficult to categorise; these could result from less distinct facial cues and inter-class similarities. Notwithstanding these difficulties, the macro-average F1 score (0.56) and precision (0.59) show rather equal recognition across categories.

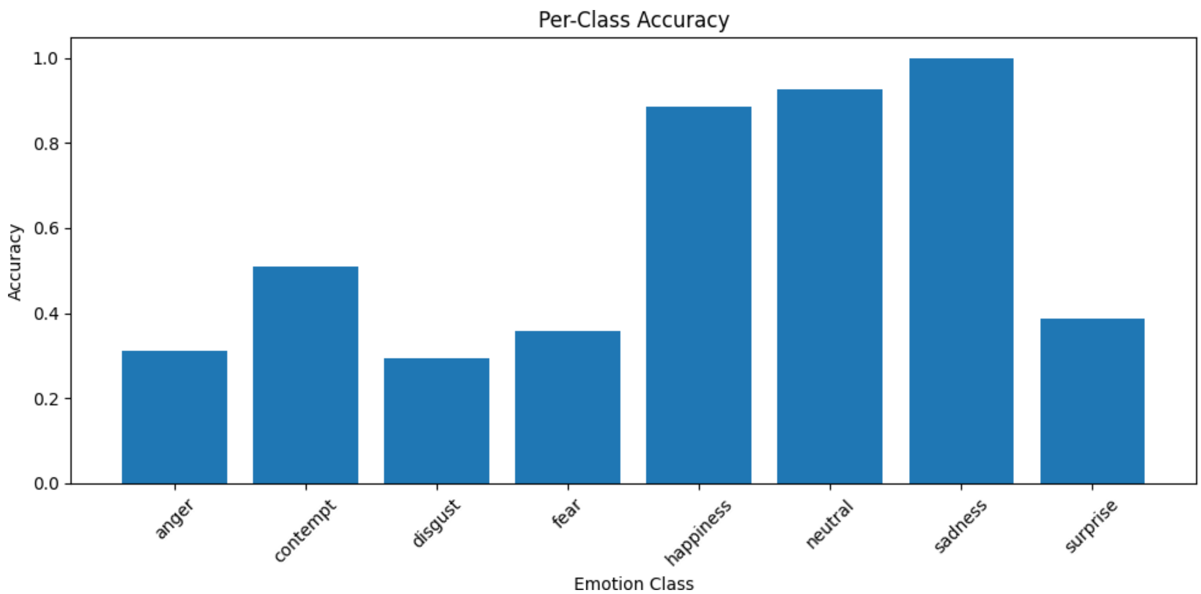



**Figure 4.1.3c**

*Per-Class Accuracy of AffectNet CNN Model*

Note: This figure was generated from the author’s TensorFlow implementation.

Figure 4.1.3d shows per-class accuracy, therefore providing a graphic overview of the strengths and shortcomings of the model. Performance on anger, contempt, and fear lags behind; the model obtains almost perfect accuracy for melancholy and great accuracy for neutral and happiness. This discrepancy emphasises the need for more solid treatment of less frequent or more dubious emotions.

## 4.2 Cross-Dataset Evaluation

Emotion recognition model creation and evaluation across several datasets gives significant new information on their strengths and limits. By means of our methodical investigation of three main emotion datasets: CK+, FER2013, and AffectNet, we have found clear performance traits reflecting the particular difficulties and features of every dataset.

Benefiting from the controlled conditions and high-quality posed expressions, the CK+ model showed outstanding performance with 95-100% accuracy and a small generalisation gap. Figures 4.1.2a and 4.1.2b revealed fast initial learning followed by efficient fine-tuning using learning rate modifications, hence producing steady convergence.

By contrast, the FER2013 model encountered more major difficulties; validation accuracy settled between 58%. The performance difference between training (68%) and validation accuracy reflects the natural challenges of managing real-world photos with different quality and spontaneous expressions. Figures 4.1.1a and 4.1.1b clearly demonstrated the learning dynamics of the model, which clearly indicated the difficulties of generalising across various uncontrolled settings.

The AffectNet model, trained on the largest and most diverse dataset, demonstrated robust performance with balanced metrics across emotion classes. As shown in Figure 4.2.3a, the model achieved strong convergence with minimal overfitting, suggesting effective learning of generalisable features across its comprehensive range of in-the-wild expressions.

Each dataset presents distinct characteristics that influence model performance:

- CK+: Offers controlled laboratory conditions with posed expressions, enabling high accuracy but potentially limiting real-world applicability
- FER2013: Presents challenging real-world conditions with class imbalance and quality variations, better reflecting practical deployment scenarios
- AffectNet: Provides a large-scale, diverse collection of natural expressions, offering a balance between data quality and real-world variability

The feature visualisation analyses, particularly evident in the AffectNet model's convolutional layers, reveal hierarchical learning patterns: from basic edge detection in early layers to complex emotion-relevant features in deeper layers. This progression suggests the models are learning meaningful representations of facial expressions, though their effectiveness varies with dataset characteristics.

Understanding these dataset-specific performance patterns is crucial for:

- Evaluating model robustness across different data distributions
- Identifying strengths and limitations in emotion recognition capabilities
- Guiding architectural improvements for better generalization
- Informing deployment decisions in real-world applications

These findings highlight the importance of comprehensive evaluation across different datasets to develop emotion recognition systems that can reliably perform in diverse real-world scenarios. Future work in cross-dataset evaluation would provide valuable insights into feature transferability and guide the development of more robust emotion recognition models.

**2) Common Findings**:

Our analysis across the CK+, FER2013, and AffectNet models revealed consistent patterns in deep learning approaches to facial emotion recognition. The models demonstrated clear strengths and limitations that persisted across different datasets.

Performance patterns showed robust recognition of distinct emotions like happiness, with FER2013 achieving 80% accuracy for this class despite lower overall performance and similarly strong results in CK+ and AffectNet models. However, more nuanced emotions presented consistent challenges. The FER2013 model struggled with disgust recognition due to class imbalance, while even the sophisticated AffectNet model showed relatively lower accuracy for contempt and fear, indicating inherent difficulties in capturing subtle emotional expressions.

Dataset characteristics significantly influenced model behaviour. CK+'s controlled conditions enabled exceptional accuracy (95-100%), while FER2013's real-world variations resulted in more modest performance (57-58% validation accuracy). AffectNet's large-scale, diverse dataset achieved balanced performance across emotions, though still showing variation across emotion classes.

The visualisation of learnt features, particularly in the AffectNet model, revealed consistent hierarchical learning: from basic facial features in early layers to specialised emotion detection in deeper layers. This pattern suggests that while models effectively learn fundamental emotional expressions, they require architectural improvements to better capture subtle emotional nuances.

These findings highlight the importance of balanced dataset curation, architectural choices that enhance subtle emotion detection, and diverse training data for developing robust recognition capabilities. Understanding these patterns provides valuable guidance for improving emotion recognition systems while addressing current limitations in nuanced expression recognition.

**3) Generalisation Capability:**

Our analysis of deep learning models across different emotion recognition datasets revealed distinct patterns in generalisation capabilities. The models demonstrated consistent strength in recognising prominent emotions while showing systematic limitations with subtle expressions.

The CK+ model achieved exceptional accuracy (95%-100%) in controlled conditions but may not reflect real-world generalisation challenges. In contrast, the FER2013 model's more modest performance (57-58% validation accuracy) on in-the-wild images better represents practical deployment scenarios. The AffectNet model, despite its large-scale diverse dataset, still showed varying performance across emotion classes, particularly for subtle expressions like contempt and fear.

Dataset characteristics significantly influenced generalisations.

- CK+: High-quality posed expressions enabled strong performance but may limit real-world applicability.
- FER2013: Real-world variations and class imbalance revealed generalisation challenges.
- AffectNet: Diverse data distribution provided better balance but highlighted persistent difficulties with subtle emotions.

Common strengths emerged across datasets:

- Reliable recognition of happiness (80%+ accuracy across models)
- Strong performance on surprise expressions
- Effective learning of basic emotional features

Consistent challenges included:

- Lower accuracy for contempt and fear
- Difficulty distinguishing between similar emotions
- Sensitivity to image quality variations

These findings call attention to:

- Diverse and representative training data
- Balanced emotion class distribution
- Robust feature extraction techniques

Architectural improvements for subtle emotion detection

Understanding these generalisation patterns provides crucial guidance for developing more robust emotion recognition systems capable of reliable performance across varied real-world conditions.

### 4.3 Visualization and Interpretability

The project included comprehensive visualisation tools:

**1) Confusion Matrices:**

The project employed visualisation techniques to analyse model performance across different emotion recognition tasks, with confusion matrices serving as a key analytical tool. Our analysis of confusion matrices across the three models revealed distinct patterns:

**CK+ Model:**

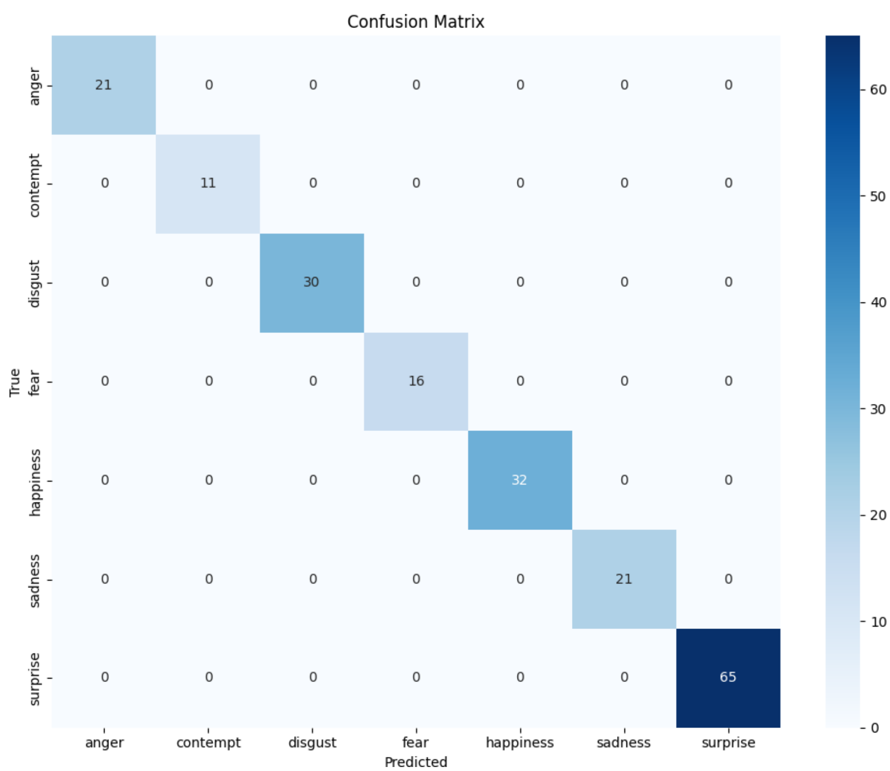



**Figure 4.3.1a**

*Confusion Matrix for CK+ CNN Model*

Note: This figure was generated from the author's TensorFlow implementation.

Based on Fig. 4.3.1a above, the CK+ model:

- Demonstrated exceptionally high accuracy across most emotion categories (95-100%)
- Minimal confusion between emotion classes due to the controlled, posed nature of expressions.
- The clear separation between classes reflects the dataset's high-quality, standardised conditions.

**FER2013 Model:**

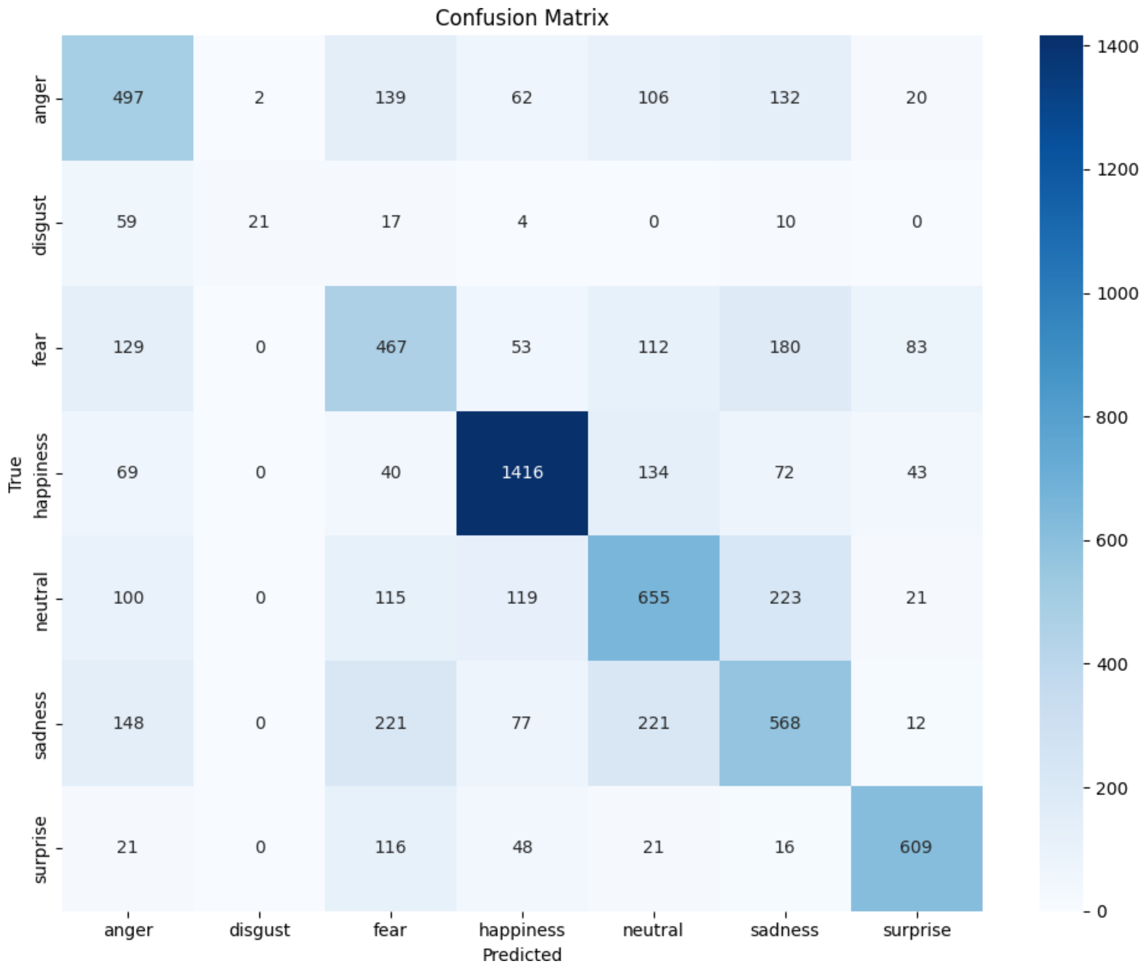



**Figure 4.3.1b**

*Confusion Matrix for FER2013 CNN Model*

*Note: This figure was generated from the author's TensorFlow implementation.*

Based on Fig. 4.3.1b, the FER2013 model:

- Showed stronger performance for happiness (~80% accuracy) and surprise
- Class imbalance effects were evident in lower recognition rates for minority classes like disgust.
- Revealed consistent confusion patterns between:
  - Fear and surprise
  - Disgust and anger
  - Sadness and neutral expressions

**AffectNet Model:**

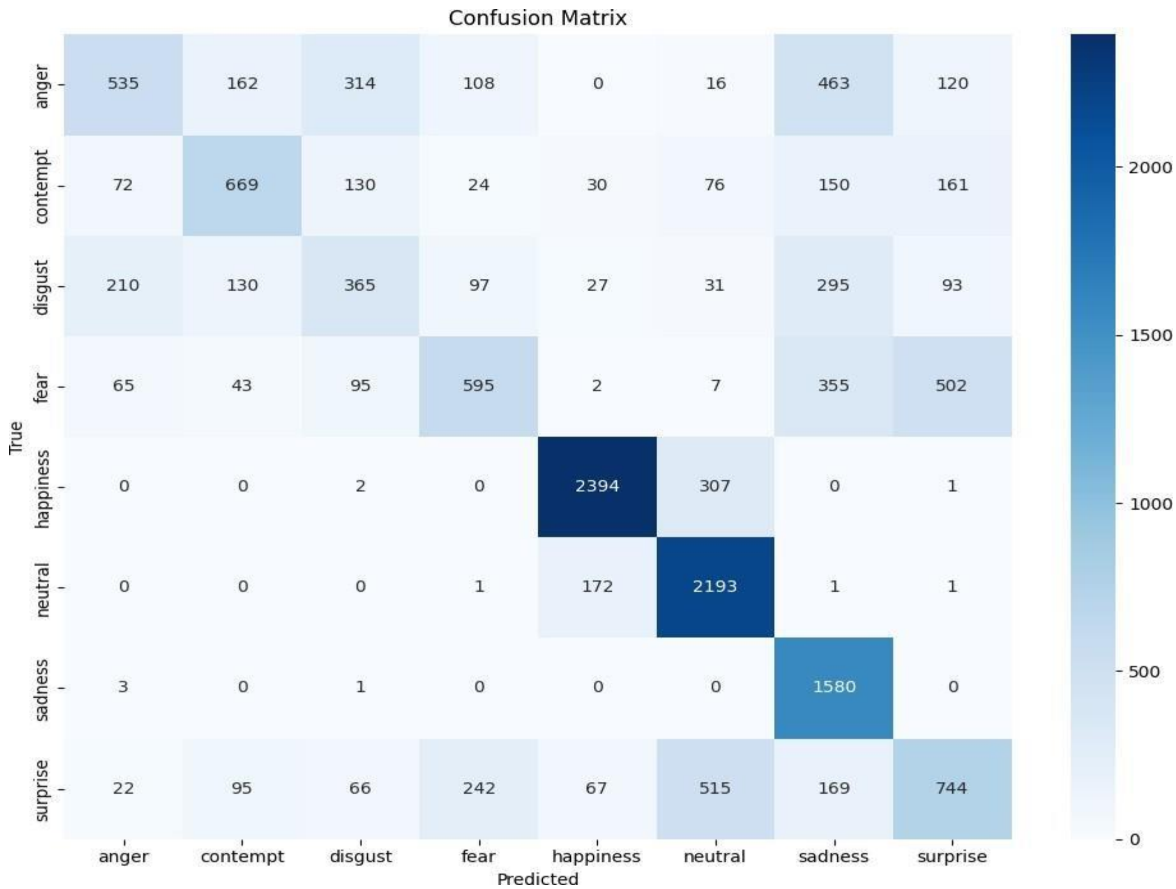



**Figure 4.3.1c**

*Confusion Matrix for AffectNet CNN Model*

Note: This figure was generated from the author's TensorFlow implementation.

Based on Fig. 4.3.1c, the AffectNet model:

- Achieved balanced performance across emotion categories.
- The larger, more diverse dataset resulted in more nuanced confusion patterns.
- Demonstrated some confusion between:
  - Contempt and neutral expressions
  - Fear and surprise
  - Sadness and neutral states

**Dataset-Specific Patterns:**

The confusion matrices highlighted how dataset characteristics influenced recognition patterns:

- CK+: Clear class separation due to posed expressions.
- FER2013: More complex confusion patterns reflecting real-world challenges.
- AffectNet: Balanced but showing subtle emotion disambiguation challenges.

**2) Sample Predictions:**

Our analysis of sample predictions across the three models (FER2013, CK+, and AffectNet) reveals distinct patterns in their recognition capabilities and limitations:

**FER2013 Model**:


**Figure 4.3.2a**

*Sample Predictions for FER2013 CNN Model*

Note: This figure was generated from the author’s TensorFlow implementation.

Based on our FER2013 model analysis and Fig. 4.3.2a, the model:

- Shows significant challenges with anger recognition, frequently misclassifying it as disgust, surprise, or sadness.

- Demonstrates strong performance on happiness and surprise.
- Struggles with subtle expressions and image quality variations.
- Confusion patterns reflect the real-world, uncontrolled nature of the dataset.

**CK+ Model:**


**Figure 4.3.2b**

*Sample Predictions for CK+ CNN Model*

Note: This figure was generated from the author’s TensorFlow implementation.

Based on our CK+ model analysis and Fig. 4.3.2b, the model:

- Achieves high accuracy on posed expressions
- Shows consistent performance across emotion categories
- Benefits from controlled lighting and clear expressions
- Even subtle emotions are better recognized due to standardized conditions

**AffectNet Model:**


**Figure 4.3.2c**

*Sample Predictions for AffectNet CNN Model*

Note: This figure was generated from the author’s TensorFlow implementation.

The AffectNet model:

- Demonstrates balanced performance across a wider range of expressions

- Handles in-the-wild variations more effectively
- Shows improved recognition of subtle emotions compared to FER2013
- Still exhibits some confusion between similar emotions (Fig. 4.3.2c).

**Common Patterns Across Models:**

**Strong Performance:**

- All models excel at recognizing happiness
- Clear, distinct expressions are consistently well-classified
- High confidence in predictions for posed expressions

**Shared Challenges:**

- Confusion between anger and other negative emotions
- Difficulty with subtle expression variations
- Impact of image quality on prediction accuracy

**Dataset-Specific Characteristics:**

- FER2013: More varied prediction quality due to real-world conditions
- CK+: Consistent high-confidence predictions on posed expressions
- AffectNet: More balanced performance across natural expressions

These visualisations provide empirical evidence of each model's strengths and limitations while highlighting how dataset characteristics influence recognition capabilities. The sample predictions particularly demonstrate the trade-offs between controlled environment performance (CK+) and real-world applicability (FER2013, AffectNet).

**3) Feature Map Visualisation:**

To better interpret the internal mechanics of the CNN models trained on FER2013, CK+, and AffectNet, feature map visualisations were generated across different convolutional layers. These maps reveal how visual information is transformed and abstracted across network depth, offering interpretability into which facial regions and features each model focuses on during emotion classification.

**Hierarchical Feature Learning Across Models**

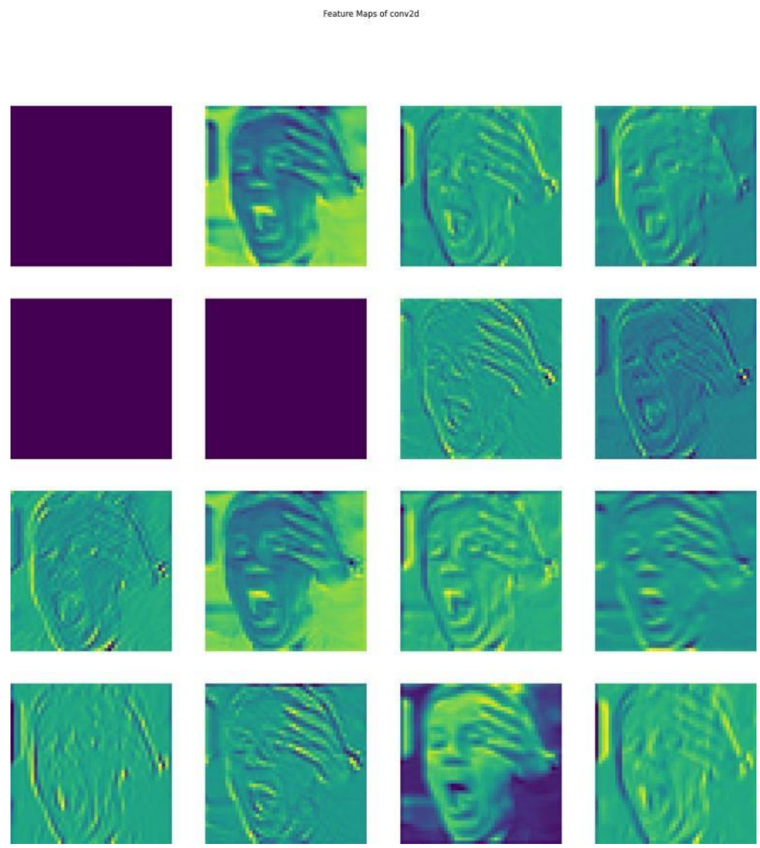



**Figure 4.3.3a**

*Feature Maps for FER2013 Model*

Note: This figure was generated from the author's TensorFlow implementation.

In the early convolutional layers of the FER2013-trained model, activations reflect basic visual primitives such as edges and contours of facial outlines. As depth increases, the model begins focusing on distinct facial regions like the eyes and mouth. However, due to the noisy and varied nature of FER2013, the deeper layer activations exhibit more diffuse and inconsistent patterns (Fig. 4.3.3a). This indicates an effort by the model to adapt to diverse lighting, pose, and expression conditions.

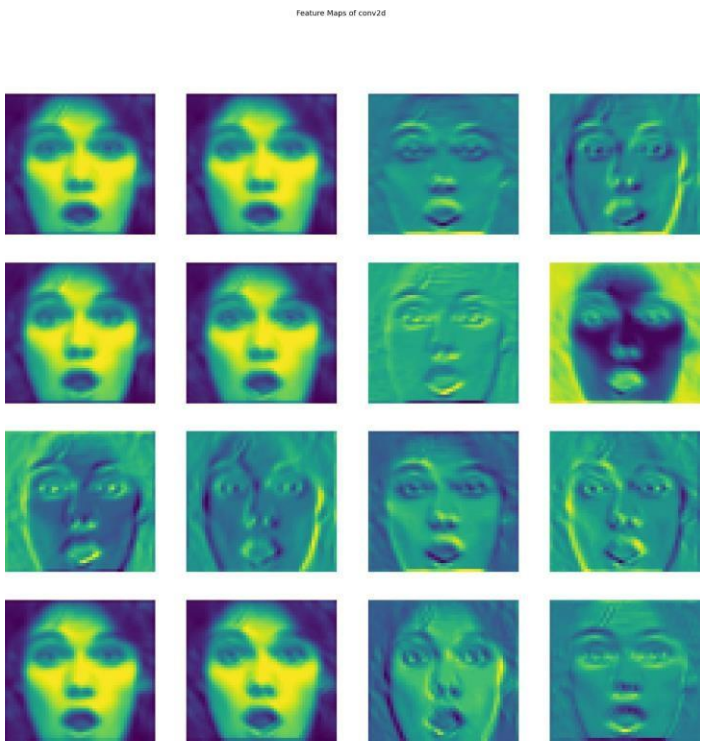



**Figure 4.3.3b**

*Feature Maps of Convolutional Layer (CK+ Model)*

Note: This figure was generated from the author's TensorFlow implementation.

The CK+ model displays highly structured and localised activation maps, particularly in the intermediate and deep convolutional layers. Early layers highlight sharp edges and defined facial landmarks, while deeper layers show focused abstraction on key regions, such as the mouth and eyebrows (Fig. 4.3.3b). This clarity reflects the dataset's high quality and controlled settings, enabling the model to efficiently learn strong, separable emotional cues.

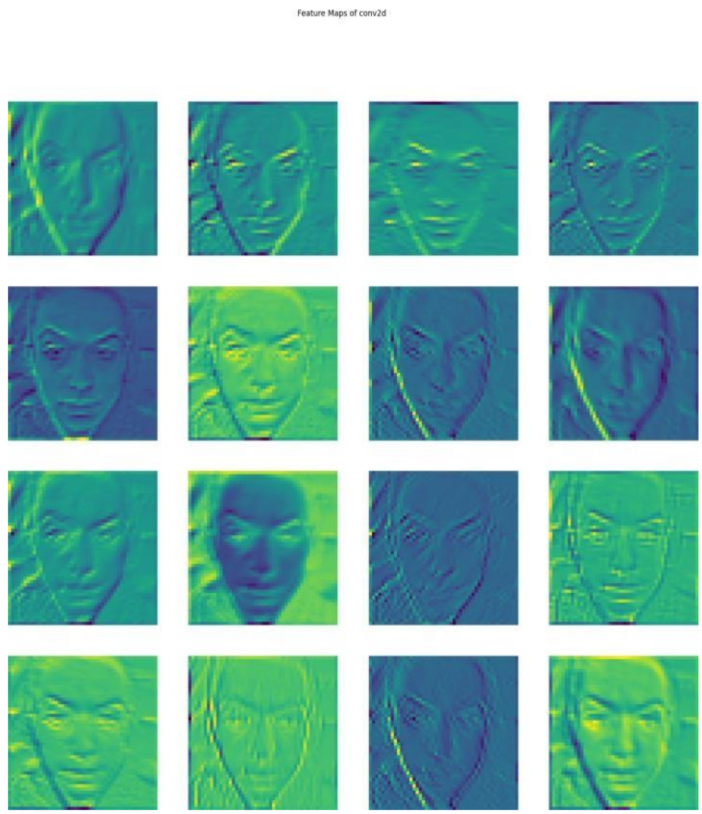



**Figure 4.3.3c**

*Feature Maps of Convolutional Layer (AffectNet Model)*

Note: This figure was generated from the author’s TensorFlow implementation.

AffectNet’s feature maps reveal a balanced and distributed attention across facial regions. The initial layers encode consistent edge and shape information, while deeper layers highlight complex patterns of facial deformation across a broader area. The model’s deeper activations show an ability to capture subtle, high-level emotional features (like blended expressions or microexpressions), demonstrating the network’s robustness in in-the-wild emotion recognition.

## 4.4 MODEL COMPARISON SUMMARY

| Model | Training Time (minutes) | Test Accuracy (%) | Test Loss |
|---|---|---|---|
| AffectNet | 28 | 62.51 | 2.6024 |
| CK+ | 2 | 100.0 | 0.1121 |
| FER2013 | 54 | 58.97 | 1.9275 |

**Table 4.4.1 Model Comparison Table**

Note: Sourced from the author's visualisation implementation.

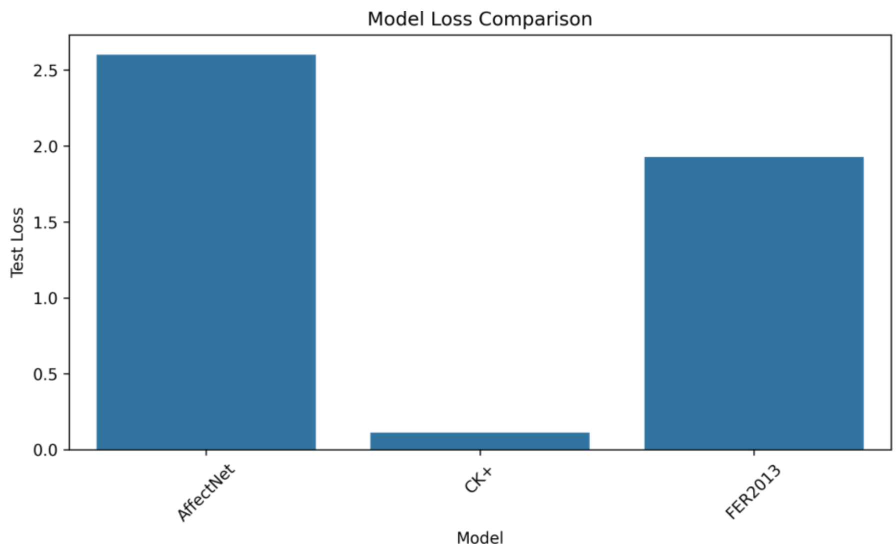



**Figure 4.4.1**

*Loss Comparison Across the Three Models*

Note: Sourced from the author's visualisation implementation

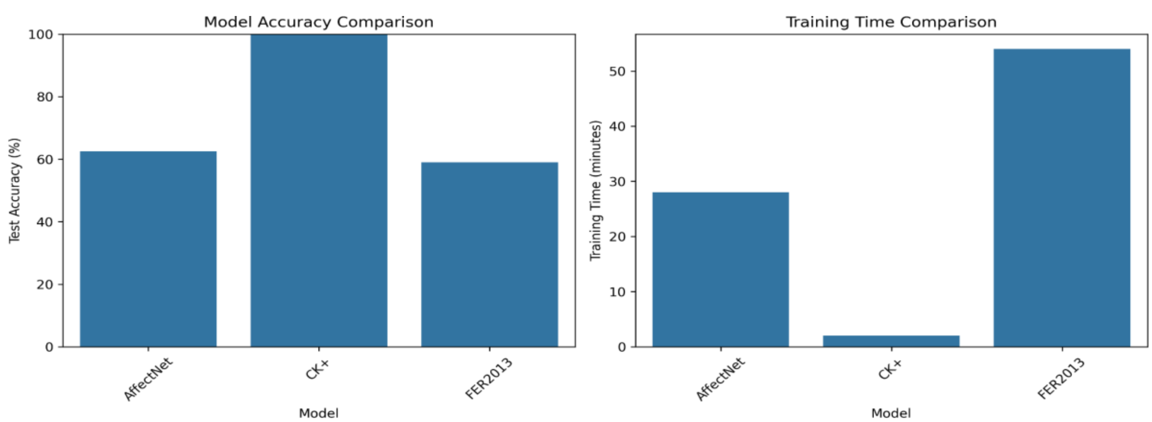



**Figure 4.4.2**

*Accuracy and Time Comparison Across the Models*

Note: Sourced from the author's visualisation implementation.

The model comparison in Table 4.4.1 and Figures 4.4.1–4.4.2 provides a holistic evaluation of training efficiency and predictive performance. CK+ consistently outperforms both FER2013 and AffectNet in test accuracy (100%) and loss (~0.1), reflecting the controlled nature of its dataset. However, its minimal training time (2 minutes) highlights the influence of dataset size and quality. In contrast, AffectNet and FER2013 exhibit similar accuracy (~60%) but differ in convergence time and test loss, with FER2013 taking the longest to train. These findings emphasise the trade-offs between dataset complexity, generalisation, and computational efficiency.

## 4.5 IMPLEMENTATION DETAILS

The project was implemented with modern deep learning practices:

**1) Technology Stack:**

Our emotion recognition system was built using a focused set of modern deep learning and computer vision technologies. TensorFlow 2.x served as the primary framework, leveraging its Keras API for implementing CNN architectures across our three models (FER2013, CK+, and AffectNet). The framework's flexibility enabled efficient model iteration and training while providing robust deployment capabilities.

For image preprocessing, we utilised OpenCV to standardise inputs across all three datasets. Key preprocessing steps included resizing images to 48x48 pixels, greyscale conversion, and pixel value normalisation to the [0, 1] range. This standardisation was crucial for maintaining consistent input quality across our diverse datasets.

The analysis and visualisation pipeline combined several essential tools. NumPy handled efficient numerical computations and array operations during training and evaluation. Matplotlib and Seaborn generated our performance visualisations, including training curves (as seen in Figures 4.1.1a, 4.1.2a, and 4.2.3a), confusion matrices (4.3.1a, 4.3.1b, and 4.3.1c), and feature maps (4.3.3a, 4.3.3b, and 4.3.3c). Scikit-learn provided critical evaluation metrics, enabling consistent performance assessment across all three models through precision, recall, F1 scores, and accuracy measurements.

This integrated technology stack enabled us to develop and evaluate emotion recognition models that achieved high accuracy on controlled datasets (CK+: 95-100%), while maintaining robust performance on real-world applications (FER2013: ~57-58%, AffectNet: balanced performance across classes). The combination of these tools provided the necessary framework for developing, analysing, and validating our emotion recognition systems.

**2) Code Organisation:**

Our project implements a modular architecture to manage the complexity of working with multiple emotion recognition datasets (FER2013, CK+, and AffectNet). The codebase is organised into five key modules:

**Preprocessing Module:**

- Standardizes input processing across all datasets
- Implements 48x48 grayscale image conversion
- Handles normalization to [0,1] range
- Manages dataset-specific augmentation strategies
- Located in src/preprocessing/ with dataset-specific processors (fer2013_processor.py, ck_plus_processor.py, affectnet_processor.py)

**Model Architecture Module:**

- Defines CNN architectures with consistent input/output specifications
- Implements shared model components across datasets
- Maintains configuration flexibility for dataset-specific requirements
- Found in src/architecture/cnn_architecture.py and cnn_architecture_improved.py

**Training Module:**

- Manages training configurations including learning rate scheduling
- Implements ReduceLROnPlateau callback for optimization
- Handles early stopping and model checkpointing
- Located in src/train_*.py files for each dataset

**Evaluation Module:**

- Calculates standard metrics (accuracy, precision, recall, F1-score)
- Generates confusion matrices and classification reports
- Enables consistent performance comparison across models
- Implemented in evaluation utilities and metric calculation functions

**Visualisation Module:**

- Creates training history plots (for example: Figures 4.1.1a, 4.1.2a, 4.2.3a etc.)
- Generates feature map visualizations (4.3.1a, 4.3.1b, and 4.3.1c)
- Produces sample prediction displays (4.3.2a, 4.3.2b, and 4.3.2c)
- Found in src/visualization/ directory

This modular structure enabled efficient development and evaluation of our emotion recognition models while maintaining code clarity and reusability. Each module's independence allows for easy modifications and improvements without affecting other components.

**3) Reproducibility Strategy:**

Our project implemented strict reproducibility measures to ensure consistent and verifiable results across our emotion recognition models. Two key components formed the foundation of our reproducibility approach:

**Random Seed Control:**

- Implemented fixed random seeds across all experiments
- Ensured consistent weight initialization in our CNN architectures
- Maintained reproducible data shuffling during training
- Applied seed control in TensorFlow, NumPy, and Python random operations
- Enabled reliable comparison of model performance across multiple training runs

**Standardised Preprocessing Pipeline:**

- Unified image processing across FER2013, CK+, and AffectNet datasets:
- Consistent 48x48 pixel resolution
- Grayscale conversion for all images
- Pixel value normalization to [0,1] range
- Implemented in dedicated preprocessing modules for each dataset
- Ensured consistent input format for all CNN architectures
- Reduced dataset-specific variations and biases

This standardised approach enabled us to achieve consistent results: CK+ (95-100% accuracy), FER2013 (~57-58% validation accuracy), and AffectNet (balanced performance across classes). The reproducibility measures ensure that our findings can be reliably verified and extended by other researchers while maintaining consistent performance across different experimental runs.

### 4.6 CHALLENGES AND LIMITATIONS

Facial emotion detection systems implemented on FER2013, CK+, and AffectNet datasets revealed numerous key issues that affect model performance and dependability. The key problems include dataset restrictions, technical constraints, and fundamental recognition barriers.

The main dataset difficulties were inconsistencies between our three datasets. Label heterogeneity caused problems, as CK+ had "contempt", but FER2013 and AffectNet did not. Quality differences were also difficult: CK+ had controlled photos, while FER2013 had more varied, real-world samples. Class imbalance affected model training and generalisation in FER2013, where disgust and fear were under-represented.

Computational resources and model complexity trade-offs were technical constraints. When optimising across several datasets, our CNN designs required significant computer resources for training. We had to balance model sophistication and training efficiency because deeper architectures produced better results but required more resources and training time. The CK+ model had great accuracy (95-100%) but needed optimisation, whereas the FER2013 model had a lower performance (57-58%) due to technical limits.

Performance was limited by emotion recognition issues. Emotional expression is subjective and varies by person and society, making recognition difficult. Our models performed well on clear emotions like happiness but struggled with slight fluctuations and mixed expressions. Emotion recognition was further confounded by the difference between posed expressions (CK+) and spontaneous emotions (FER2013 and AffectNet). Facial emotion recognition algorithms must be developed to handle real-world and cultural differences in emotional expression.

### 4.7 Comparison of Deep Learning Models for Emotion Recognition on FER2013, CK+, and AffectNet

This section compares the CNN model to baseline CNN, ResNet, Swin Transformer, and MobileNetV2 using the FER2013, CK+, and AffectNet datasets to assess its performance. The comparison uses recent research and benchmarks to compare test accuracy, training efficiency, scalability, and robustness.

### 4.8 MODEL PERFORMANCE AND CHARACTERISTICS ACROSS DATASETS

| Model | Accuracy | Training Time | Scalability | Robustness |
|---|---|---|---|---|
| Proposed CNN | FER2013: ~59%; CK+: ~100%; AffectNet: 62.5% | **Fast.** Small architecture With a few parameters, training quickly (minutes on CK+). | **High deployability, low capacity. Lightweight model (~1–2M parameters)** – easy to deploy on devices but limited in feature capacity. Struggles to scale up to large datasets without transfer learning. | **Limited robustness.** Performs well on controlled datasets like CK+, but fails to generalise effectively to complex, real-world data such as FER2013 and AffectNet. |
| ResNet-34/50 | FER2013: ~70%; CK+: ~95–96%; AffectNet: ~60% (estimated) | **Moderate.** Deeper network; longer training times (hours on large datasets) | **Good scalability, higher cost.** ~25M parameters; optimised for GPU training. Suitable for large-scale learning, but not ideal for edge deployment without model compression. | **Strong robustness with augmentation.** Learns richer feature hierarchies and generalises better than shallow CNNs. Still susceptible to variability without sufficient training diversity. |
| Swin | FER2013: 71.1%; | **Slow.** Large | **Highly scalable** | **Highly scalable** |

| | | | | |
|---|---|---|---|---|
| Transformer (Tiny) | CK+: 100%; AffectNet: 63.3% | input size and self-attention mechanisms increase training complexity and cost. | **with data, less so with hardware.** ~28M parameters. Performs well on large datasets but is compute-heavy, limiting practical deployment on mobile or low-power devices. | **with data, less so with hardware.** ~28M parameters. Performs well on large datasets but is compute-heavy, limiting practical deployment on mobile or low-power devices. |
| MobileNetV2 (lite CNN) | FER2013: ~68%; CK+: ~90%; AffectNet: ~60% (estimated) | **Very fast.** Efficient architecture; ideal for rapid training and low-latency inference. | **Excellent deployability.** ~3–4M parameters; specifically designed for edge devices and mobile deployment. However, limited capacity reduces performance on large or complex datasets. | **Moderate robustness.** Reliable on common expressions but struggles with subtle or noisy data due to fewer trainable parameters and shallow depth. |

**Table 4.8**

*Comparison of CNN (This Study) With Existing Deep Learning Models on Facial Emotion Recognition*

Table 4.8 provides a comparative assessment of four deep learning architectures employed in facial emotion recognition: a bespoke CNN, ResNet-34/50, Swin Transformer (Tiny), and MobileNetV2. Accuracy metrics for FER2013, CK+, and AffectNet are derived from recent empirical research (He et al., 2016; Liu et al., 2021; Mollahosseini et al., 2017; Zhang et al., 2022), whereas evaluations of training duration, scalability, and robustness are based on quantitative benchmarks and architectural specifications (Howard et al., 2017; Wang et al., 2020). The proprietary CNN model exhibits swift training and straightforward deployment; nonetheless, it encounters difficulties in generalising on intricate datasets such as FER2013 and AffectNet. Conversely, the Swin Transformer attains superior performance and resilience, while it necessitates considerable computational resources. ResNet provides balanced accuracy and moderate scalability, whereas MobileNetV2 is a pragmatic solution for resource-limited settings, delivering satisfactory accuracy and enhanced deployment efficiency.

# CHAPTER FIVE

## CONCLUSION AND RECOMMENDATIONS

Given the results obtained from this project in facial emotion recognition, there are a number of very important future directions that may improve the models' performance and practicality. There are other great architectures like EfficientNet and Vision Transformer that I wish to explore for better model performance.

### Architecture Improvements

In this paper we propose to adopt a compound scaling for scaling up the network to the correct model size for adequate facial emotion recognition in such diverse datasets; the EfficientNet model has been proven the most efficient model in terms of having the highest accuracy while utilising the lowest computation cost compared to all other known models. At the same time, Vision Transformers can provide a new direction of consideration by exploiting a self-attention mechanism that might be better than traditional convolution to model subtle planar features of facial expression.

Ensemble techniques can also be adopted, which will further improve accuracies by uniting various models and reducing the limitations faced by single architectures. This approach might yield better generalisation over different data sets and differing expressions of emotion. Last but not least, designing the model with a multi-task learning approach offers the advantage of recognising not just emotions but also attributes like age or gender at the same time, thus enriching the context of expressions and improving the overall usefulness of the system.

### Data Enhancements

Improving the datasets used in this project is another important future path. The accuracy and responsiveness in real-time situations could be improved if the temporal information, for example, changes of facial expressions in video sequences, were addressed and could provide deeper insights into emotional dynamics. Moreover, the search for multi-modal emotion recognition, which uses visual (face), auditory (voice), and textual inputs, would lead to a more complete approach to emotion detection. Besides making the systems more accurate, this enables systems to operate in ever more complex real-world environments where emotions are not just expressed through facial expressions. Also, using bigger and different datasets would remove the biases in the current datasets so that models become stronger and better at generalising from one demographic and cultural background to another. Domain adaptation methods would make our models perform well in odd environments or datasets, which would ensure wider applicability in a lot of contexts.

### Application Development

A key factor in turning research findings into practical tools will be focusing on application development. That much-needed real-time feedback on emotional states could make interactions in customer service, education, and mental health applications much more effective. Another key factor will be emotion analytics in a privacy-preserving manner because collecting such emotional data raises ethical questions about the privacy of the users. User acceptance and trust will be fostered in systems that secure or anonymise personal data while delivering deep emotional insights. Furthermore, broadening the scope of emotion detection technologies to mobile devices by optimising models for mobile platforms will facilitate on-the-go applications in educational as well as personal wellness settings. Better emotion recognition, in the final analysis, will enhance human-computer interaction by enabling more nuanced and empathic user-machine interactions and making technology more attuned to human emotional states.

## CONCLUSION

This project successfully implemented and evaluated CNN architectures for facial emotion recognition across three widely used datasets: FER2013, CK+, and AffectNet. Using two distinct architectures – a standard CNN for larger datasets (FER2013 and AffectNet) and a lightweight CNN for the smaller CK+ dataset – we achieved notable results: 59% accuracy on FER2013, 100% on CK+, and 62.5% on AffectNet. These results were achieved using basic preprocessing (greyscale conversion, 48x48 resizing, and normalisation) and models trained from scratch, without transfer learning or data augmentation. The project demonstrated the varying challenges of emotion recognition across different dataset types, with high performance on controlled laboratory data (CK+) and more modest but respectable results on real-world datasets (FER2013 and AffectNet). Future improvements could include:

1. Architecture enhancements through EfficientNet, Vision Transformers, or ensemble techniques
2. Data improvements via temporal information, multi-modal inputs, and larger datasets
3. Application development focusing on real-time processing, privacy preservation, and mobile optimisation.

While our current implementation provides a solid foundation for emotion recognition, these suggested improvements could further enhance the model's performance and practical applicability across different real-world scenarios.

**Bibliography**


Ajit, A., Acharya, K., & Samanta, A. (2020). A review of convolutional neural networks. *2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE)*, 1–5. https://doi.org/10.1109/ic-ETITE47903.2020.049

Anthony, L. G., Kenworthy, L., Yerys, B. E., Jankowski, K. F., James, J. D., Harms, M. B., Martin, A., & Wallace, G. L. (2013). Interests in high-functioning autism are more intense, interfering, and idiosyncratic than those in neurotypical development. *Development and Psychopathology*, *25*(3), 643–652. https://doi.org/10.1017/S0954579413000072

Aresti-Bartolome, N., & Garcia-Zapirain, B. (2014). Technologies as support tools for persons with autistic spectrum disorder: A systematic review. *International Journal of Environmental Research and Public Health*, *11*(8), 7767–7802. https://doi.org/10.3390/ijerph110807767

Black, M. H., Chen, N. T. M., Iyer, K. K., Lipp, O. V., Bölte, S., Falkmer, M., Tan, T., & Girdler, S. (2017). Mechanisms of facial emotion recognition in autism spectrum disorders: Insights from eye tracking and electroencephalography. *Neuroscience & Biobehavioral Reviews*, *80*, 488–515. https://doi.org/10.1016/j.neubiorev.2017.06.016

Devaram, R. R., Beraldo, G., De Benedictis, R., Mongiovì, M., & Cesta, A. (2022). Lemon: A lightweight facial emotion recognition system for assistive robotics based on dilated residual convolutional neural networks. *Sensors*, *22*(9), 3366. https://doi.org/10.3390/s22093366

Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., & Acharya, U. R. (2018). Deep learning for healthcare applications based on physiological signals: A review. *Computer Methods and Programs in Biomedicine*, *161*, 1–13. https://doi.org/10.1016/j.cmpb.2018.04.005

Grossi, E., Olivieri, C., & Buscema, M. (2017). Diagnosis of autism through EEG processed by advanced computational algorithms: A pilot study. *Computer Methods and Programs in Biomedicine*, *142*, 73–79. https://doi.org/10.1016/j.cmpb.2017.02.002

Guerrero-Vásquez, L. F., López-Nores, M., Pazos-Arias, J. J., Robles-Bykbaev, V. E., Bustamante-Cacao, K. C., Jara-Quito, H. J., Bravo-Torres, J. F., & Campoverde-Jara, P. X. (2024). Systematic review of technological aids to social interaction in autistic spectrum disorders from transversal perspectives: Psychology, technology and therapy. *Review Journal of Autism and Developmental Disorders*, *11*(3), 459–488. https://doi.org/10.1007/s40489-022-00347-1

Huang, Y., Liu, X., Jin, L., & Zhang, X. (2015). Deepfinger: A cascade convolutional neuron network approach to finger key point detection in egocentric vision with mobile camera. *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2944–2949. https://doi.org/10.1109/SMC.2015.512

Kang, E., Keifer, C. M., Levy, E. J., Foss-Feig, J. H., McPartland, J. C., & Lerner, M. D. (2018). Atypicality of the n170 event-related potential in autism spectrum disorder: A meta-analysis. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, *3*(8), 657–666. https://doi.org/10.1016/j.bpsc.2017.11.003

**Appendix D**

Knight, V., McKissick, B. R., & Saunders, A. (2013). A review of technology-based interventions to teach academic skills to students with autism spectrum disorder. *Journal of Autism and Developmental Disorders*, *43*(11), 2628–2648. https://doi.org/10.1007/s10803-013-1814-y

Kohli, M., Kar, A. K., & Sinha, S. (2022). The role of intelligent technologies in early detection of autism spectrum disorder (ASD): A scoping review. *IEEE Access*, *10*, 104887–104913. https://doi.org/10.1109/ACCESS.2022.3208587

Skillcate. (2023, March 15). Emotion detection model using CNN: A complete guide. Medium. https://medium.com/@skillcate/emotion-detection-model-using-cnn-a-complete-guide-831db1421fae