Middlesex University London

CST4090 Individual Project

Title: Enhancing Bone Fracture Detection and Classification through Edge Detection or Computer Vision and ML/DL in Medical Imaging.

Student Name: Anaele Harrison Tochukwu

Student ID: M00916575

Supervisor Name: Prof. Gao

Table of Content

Table of Contents Page

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
SVM	Support Vector Machine
VGG16	Visual Geometry Group
ResNet	Residual Network
Deep ConvNet	Deep Convolutional Neural Network
AI	Artificial Intelligence
CT	Computed Tomography
ML	Machine Learning
DL	Deep Learning
XAI	Explainable Artificial Intelligence
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic curve
AUC	Area Under the Curve
ReLU	Rectified Linear Unit
HTSCNs	Hidden Two-Stream Convolutional Networks
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Abstract

Diagnosis of bone fractures by experts through the examination of X-ray images is typically a manual and time-prolonged procedure. However, the evolution of machine learning and neural network know as DL has opened recent opportunities for medical or clinical image analysis or detection. In this research, we introduce a unique wide attribute integration approach that combines a deep ConvNet with an enhanced canny edge detection technique to effectively differentiate between fractured and healthy bone images.This hybrid approach leverages canny edge detection for feature extraction to identify the edges of objects in an image, which can serve as crucial features for identifying regions of interest and deep ConvNet for classification, improving the model's capacity to learn more meaningful features related to object boundaries and offering competitive computational times and diagnostic accuracy compared to other CNN models. The key advancement of this research lies in utilizing an enhanced canny edge algorithm to detect edges that accurately pinpoint fracture areas in the images. These edge-detected images are then input into the deep ConvNet for learning and assessment.

The performance of our model was compared against advanced deep ConvNet models, including ResNet and VGG16-SVM, using X-ray medical imaging data. The use of canny edge detection contributed to the model's robustness by emphasizing critical image features. The hybrid model demonstrated a significant increase in classification accuracy, achieving 98%, surpassing the results of the other models. The findings revealed that the deep ConvNet integrated with canny edge detection achieved top precision of 98%, recall of 98.5%, F1-score of 98%, and an accuracy of 98.08% for bone fracture diagnosis. This demonstrates that incorporating canny edge detection significantly enhances the effectiveness of deep ConvNet models. In conclusion, the hybrid deep ConvNet with canny edge detection offers a superior solution for bone fracture detection, providing a reliable tool for medical professionals.

Acknowledgement

Above all, I express my gratitude to God Almighty for His grace, guidance, and blessings throughout this journey. None of this would have been possible without His divine intervention.

I would like to acknowledge all the lecturers in the Data Science department of this esteemed institution for imparting their knowledge. My special thanks go to my supervisor, Prof. Gao XiaoHong, for his guidance, direction, and support in realizing this research. God bless you, Ma.

I remain deeply indebted to my wife, Ogechukwu, and I want to express my deepest gratitude to my unborn child. Though you have yet to take your first breath, you have already filled our hearts with joy and anticipation. Your presence has been a source of inspiration and motivation for me throughout this journey. I cannot wait to meet you and share the world with you. Thank you both for your moral, spiritual, and emotional support.

Special thanks to my brother, Chukwuma Anaele, for your emotional and financial support throughout this journey. I also extend my gratitude to my mother, Ezinne Beatrice Anaele, and my mother-in-law, Mrs. Ifeoma Onwugbufor. Thank you for your prayers and advice. I am indeed grateful.

CHAPTER 1: INTRODUCTION

1.1 Background

Bones are an important part of the human body. It supports a person's transition from one location to another and bone fractures can be caused by an incident. It is estimated that 2.7 million fractures occur every year [1] Nicholas et al (2018), a staggering number of people are affected by this condition, and the results of an unhealed fracture might bring lifelong damage or even death. After a bone break, prompt intervention is essential. A radiologist reviews an X-ray image to identify a bone break, and the patient needs proper medical attention. The effectiveness of treatment relies on the experience and expertise of a radiologist. The challenge lies in developing an automated system that can enhance accuracy and efficiency [2] Amirkolaee et al. (2022). The traditional approach is time-consuming and might result in human errors that can delay treatment, also known as manual examination. There is a limited number of experts or radiologists available in a remote area and in certain cases, bone fractures are extremely small and challenging for doctors to detect, necessitating the use of advanced methods to enhance diagnosis. This can be achieved through computer-aided diagnosis or machine learning techniques, which can significantly improve the precision of detecting bone fractures [3] T. Anu et al (2015). There are many methods available today for classifying and detecting illnesses, with machine learning, a subset of AI, being particularly effective in automating the identification, categorization, and prediction of diseases [4] Amodeo et al, (2021). Currently, deep learning, machine learning, and advanced imaging technologies are among the leading AI tools utilized in healthcare [5] Varoquaux et al, (2022). Deep learning models, constructed on several-layered ANNs, significantly improve efficiency [6] Bengio et al, (2021). Diagnostic imaging devices like X-rays, echography, and CT scan machines are used to capture images of various conditions, but X-rays are the most commonly utilized method for diagnosing bone fractures due to their affordability and wide availability [7] Yadav et al, (2020). Among various conditions, bone fractures are the most prevalent. Bones, which are solid structures, play a vital role in the human body by protecting essential organs like the heart, lungs, and brain [8] Zhang et al, (2020).

The human body consists approximately 206 bones, each with distinct patterns and forms [9] Hardalaç et al, (2022). Bones can be classified into several types, such as flat, short, long, sesamoid, and irregular. The thigh bone is the longest in the human, while the ear bone is the tiniest. Bone fractures are a common issue that tends to worsen over time. A fracture occurs when a bone is subjected to forces it cannot withstand, with long fractures being particularly common. Fractures can result from various incidents such as car incidents, gunshot wounds, or sports-related injuries [10] Lapeña et al, (2021). They can take many forms, including typical, **Transverse** fractures run roughly horizontally across the bone. **Oblique** fractures happen when a bone is broken at an angle, creating a diagonal fracture line. **Spiral** fractures happen when the bone is rotated, causing one or both sections to break in a spiral pattern. **Comminated** fractures cause the bone to be shattered into multiple fragments. **Avulsion** fractures occur when fragments of the bone are torn away from the main structure. **Impacted** fractures are the reverse of avulsion fractures, occurring when a piece of bone is driven into another piece of bone. **Fissure** fractures involve small splits within the bone. **Greenstick** fractures occur when the bone bends and partially breaks without fully breaking. Fractures can be classified into eight types according to their pattern, as below.

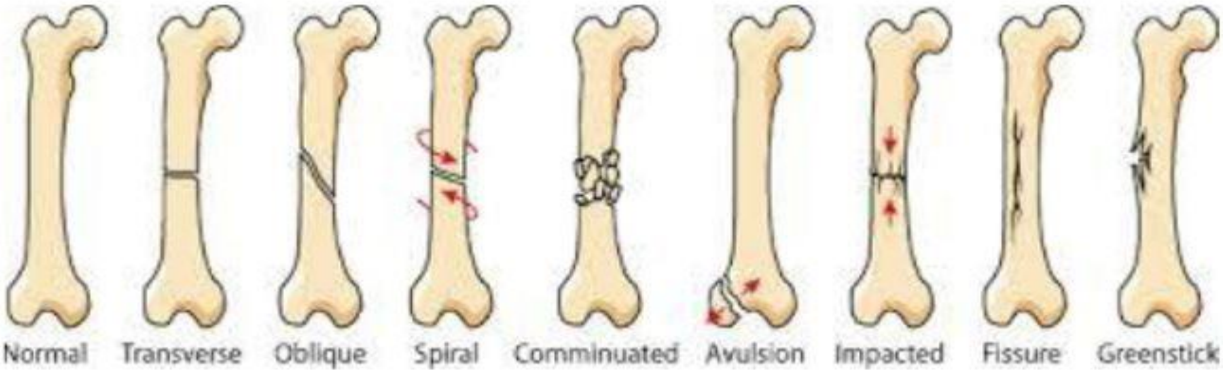


Figure 1.1: Types of Fracture (Karen M. Alexander)

Therefore, it is essential to accurately and quickly diagnose bone fractures for proper management and treatment [11] Ma et al (2020). Traditional methods for diagnosing and classifying bone fractures often involved identifying the fracture location using a bounding box.

These days, a designed system that can be relied upon and has efficient automation of DL and ML mothed is utilized to identify medical conditions instantly, improving the efficiency of the diagnosed bone fracture [12] Singh et al (2022). In this work, bone fracture will be detected with high accuracy. Therefore, upon detection, the proper identification shall be made to judge the bone fracture properly. There are more noises in the X-ray images, and they can be removed using the image preprocessing technique [13] Randy et al (2024). This will further reduce the clinical load on radiologists, enabling them to concentrate on more difficult cases and potentially prevent burnout. Consequently, there will be an increase in efficiency and improvement in patient outcomes in health care. An advanced bone fracture detection and classification system will have benefits in every aspect of the healthcare domain [14] Lakkimsetti et al (2024). The patients will get diagnosed faster and more precisely, which will lead to timely and appropriate treatments. Radiologists and health providers would benefit from a reduced workload on their part and increased diagnostic support, further increasing their efficiency and job satisfaction [15] Knud et al (2018). Healthcare facilities will also get better operational efficiency and reduction in costs related to misdiagnosis, among others, which include costs for prolonged care of patients [16] Ali et al (2024). In addition, medical researchers and developers benefit from the knowledge of the innovative way of using machine learning and canny edge detection in medical imaging that contributes to niche growth. The proposed method employs a specific technique to pinpoint the fracture site in bones. [Leonardo Tanzi] utilized various deep-learning methods to categorize bone breaks, aiming to identify the advantages of each approach and develop a more universal solution. They also highlighted the key factors that need to be considered to accomplish this objective and measured each study to their baseline. Recently, neural networks, particularly convolutional neural networks (CNN), has demonstrated remarkable success in classifying bone fractures, achieving accuracy comparable to that of human experts [17] Tanzi L et al (2020). A variety of literature is available for analyzing various kinds of human bones. In context paper, [18] Shuzhen et al, (2022) created a comprehensive fracture detection system employing deep ConvNet techniques. Firstly, image quality is enhanced through preprocessing methods. Afterward, data augmentation is implemented to enlarge the dataset. Lastly, Ada-ResNeSt is utilized to classify broken and not fractured bones, attaining an overall precision of 68.4%. In another paper, [19] Yang et al, (2019) employed contour features from x-ray images to detect broken bones. The system achieved an accuracy of 82.98%, which still requires improvement. In this work, [20] Cao et al (2015). Introduced a universal method for detecting bone fractures known as SRFFF, which is built on a discerning learning framework. This approach surpasses other bone breaks frameworks that rely on local features, SLRF and SVM classification. In the field of bone break, [21] D.H. Kim et al, (2017), the Inception-v3 network was fine-tuned using outer wrist X-rays to create a model that can accurately detect whether a recent case includes a fracture.

The innovation of Artificial Intelligence (AI) has been tested and explored by scholars across various industries to address challenges and generate predictive insights from processed data. AI has demonstrated significant potential in aiding disease diagnostics for patient treatment by analyzing and processing data using advanced AI technologies. Particularly, artificial intelligence, and more exactly the sub-field of Machine Learning, offers promising solutions for diagnostic and testing applications in medical radiography, particularly in detecting bone fractures. These methods enable scholars to make use of existing traditional data [22] Thomas et al (2019). Deep learning, a branch of machine learning, has demonstrated superior efficiency contrast to conventional machine learning algorithms in specific data processing tasks, as it can instantly extract features from input images and learn complex trends and specifics within the data. ConvNet are a form of neural network commonly utilized in deep learning, primarily for analyzing visual data and images. They are frequently used for tasks like trend detection, atribute identification, data division, boundary detection, binary classification, and object recognition. Nevertheless, deep learning algorithms typically require large datasets to achieve high accuracy. Handling vast amounts of image data can result in significantly longer computation times compared to traditional machine learning algorithms, which generally require smaller datasets for processing. [23] D. P. Yadav et al (2020) employed and customized a DNN for classification, attaining an efficiency and precision of 92.44% in differentiating between not fractured and broken bones through 5-fold cross-validation..

The primary contributions of this work include a hybrid model deep ConvNet integrated with canny edge detection, which is expected to yield better accuracy and lower computational costs for diagnosis of bone breaks, and an effective demonstration of integrating canny edge detection algorithms with deep learning models. The region of the fractured bone is achieved utilizing an advanced canny edge detection method. The edge images are then input into deep ConvNet model to extract features.

1.2 Research Aim

Develop a hybrid Deep ConvNet model integrating canny edge detection for accurate and efficient bone fracture detection and classification.

1.3 Research Statement

How to develop and apply deep learning models with canny edge detection to detect bone fractures in medical imaging.

1.4 Research Questions

1. What are the latest models and techniques used for fracture detection and classification in medical imaging?

2. How can a deep learning or machine learning model be developed for detecting and classifying bone fractures using medical imaging as classification features?

3. How can the viability and reliability of the proposed model be assessed?

1.5 Research Objectives

Objective 1: To collect and preprocess a comprehensive dataset of bone X-ray images.
Objective 2: To develop and train the hybrid deep ConvNet model using canny edge images.
Objective 3: To evaluate the performance of the hybrid Deep ConvNet against existing models.
Objective 4: Optimize the model for real-time application.

1.6 Research Scope

The study focuses on improving diagnostic accuracy and efficiency while addressing challenges like dataset quality. The goal is to enhance healthcare outcomes, particularly in underserved areas.

1.7 Thesis Organization

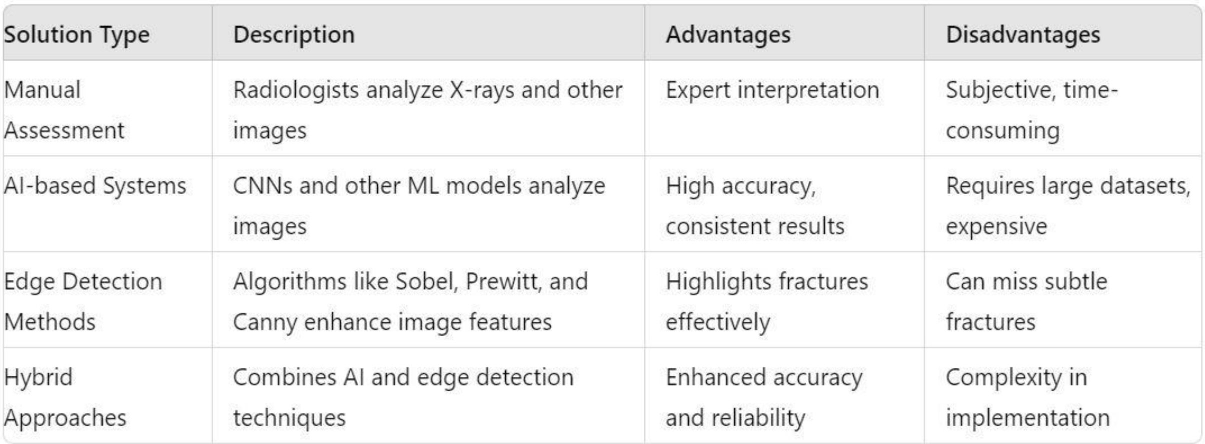
Chapter one presents a brief background information on the study and hence gives a good introduction to the research; chapter two discusses the literature review related to the topics; chapter three explains the methodology to adopt in achieving the aim and objectives of the research; chapter four contains the results and some discussions around the results and research findings; and finally, chapter five concludes the study and future research considerations.

Chapter 2: Literature Review

Bone fracture detection has been traditionally based on visual X-ray image assessment by radiologists. While this method is widely applied, it incurs a great deal of human error and variability. Thus, automated techniques have recently been introduced by advances in medical imaging and deep learning or machine learning to assist or replace manual assessments. State-of-the-art models for this domain include various machine learning models that go through medical images to detect fractures, comprising of ML and DL. Edge detection algorithms include canny and sobel operators applied on image features before applying deep learning [24] Shuzhen L et al (2022). All these methods increase the accuracy and consistency of any diagnosis.

2.1 Computer Vision Application in Medical Images

Commonly employed techniques rely on computer vision and image processing techniques. It serves an important role in the healthcare industry and has demonstrated effectiveness with satisfactory performance or achievement levels. Several approaches integrate computer vision with ML or DL techniques. For over a period, computer vision has been utilized in CT images bone break diagnosis, with ongoing research efforts aimed at improving and developing new techniques for enhanced performance. Despite the advances, the current solutions have their limitations and need improvements. Manual assessments can be subjective and prone to errors, making the results inconsistent. AI-based systems are accurate but require large, labelled datasets for training and development, which often incur significant costs. Even though edge detection methods are efficient, they may sometimes exclude subtle fractures and require a good image to work with [25] Chijioke O et al (2023). Hybrid approaches are very promising, but complex and require very elaborate integration. These challenges underscore the need for a better accuracy and more efficient system.



Solution Type	Description	Advantages	Disadvantages
Manual Assessment	Radiologists analyze X-rays and other images	Expert interpretation	Subjective, time-consuming
AI-based Systems	CNNs and other ML models analyze images	High accuracy, consistent results	Requires large datasets, expensive
Edge Detection Methods	Algorithms like Sobel, Prewitt, and Canny enhance image features	Highlights fractures effectively	Can miss subtle fractures
Hybrid Approaches	Combines AI and edge detection techniques	Enhanced accuracy and reliability	Complexity in implementation

Figure 2.1: Comparison of Fracture Detection Methods.

This paper, [26] Zhu et al, (2018) propose HTSCNs for improved action identification in video data by enhancing the fusion of spatial and temporal information. While this hidden fusion mechanism boosts performance, it increases computational complexity, which may limit its use in resource-constrained or real-time scenarios. Future studies should focus on prioritize improving process to lower computational demands while maintaining its benefits. [27] Dustakar S. R et al (2024) explores the application of V-Net for medical image segmentation, reporting an 88% overlap ratio and a 92% dice coefficient. It underscores the critical role of advanced neural architectures and optimization methods in overcoming challenges such as processing complexity/intricacy and sensitivity to noise. In 2019, the authors of [28] Y. D. Pranata et al categorized bone breaks by integrating intensity lines with speeded up robust features. They assessed the effectiveness of residual network and visual geometry group 16-layer network, both are pre-trained deep convNet models, with ResNet demonstrating the peak classification accuracy, reaching 98%.

2.2 Machine Learning Application in Medical Images

The application of ML to bone fracture detection offers a broad range of possibilities due to the variety of methodologies available. Numerous classification algorithms can be employed to differentiate between healthy and damaged bones. Some of the commonly used classification methods include XGBoost, SVM, Principal Component Analysis (PCA), Gaussian Mixture Models (GMM), and others. Machine learning techniques have demonstrated satisfactory accuracy levels in detecting bone breaks. The current solutions around the identification and categorization of bone fractures are a mix of the manual assessment conducted by radiologists and automated systems that involve machine learning algorithms. The traditional method hinges on the expertise of the radiologist for the interpretation of X-rays and other imaging modalities. An important current approach is the fully automatic use of machine learning models that can assist or even replace manual interpretation in imaging. These used DL and ML to sift through imaging data and detect fractures with high accuracy. Moreover, features of an image are boosted, and fractures become very distinct using edge-detection techniques, such as sobel, prewitt, and canny [29] Chao-Lung Y et al (2024). Many use a combination of AI and edge-detection techniques to form a hybrid approach. Such works have been developed in the past, [30] Amani Al-Ghraibah et al (2024) developed a method for classifying X-ray images in long bone using novel features and an SVM. The accuracy of the model is 80%, which needs to be improved. [31] Avinash et al, (2015) implemented various image analysis techniques, such as corner and edge detection, to identify unique features, followed using a support vector machine classifier for accurate classification. Despite these techniques performing reasonably well in bone breaks tasks, they are less commonly used today due to several drawbacks. Manual features have restricted illustration capabilities when dealing with large, convoluted datasets. Additionally, the graphics processing unit cannot accelerate the feature computation approach, making it inefficient and less suitable for immediate applications analysed against deep learning-based techniques. In similar research, [32] Valan et al (2023) created a procedure for CAD prostate cancer evaluation focusing on deriving key radiomic features and applying optimized linear support vector machine techniques. This study validates the effectiveness of integrating computational methods with radio-logical applications. [33] Luo et al, (2021) utilized their specialized insight to create a predictive model for identifying bone breaks, resulting in a precision of 86.57%. In this research, [34] Ankur et al (2017) presents a method for detecting femur fractures through image processing, using morphological techniques and SVM classification to differentiate between normal and fractured bone, underlining the femur's significance in the human body.

2.3 Deep Learning Application in Medical Images

Detecting bone breaks with deep learning utilizes various methods, such as Artificial Neural Networks, Deep ConvNets, RNNs, ResNets, Inception V3, R-CNN, and others. The use of DL in bone break detection gained significant traction after 2019, as these approaches produced compelling and valuable outcomes. Deep learning has addressed many of the limitations of previous methods, such as the challenge of analyzing images from a single anatomical position, which sometimes led to missed detection. By employing deep learning techniques like faster R-CNN and customized Crack Net, even minute cracks in CT images can now be accurately diagnosis. The development of an improved system for detecting and classifying bone fractures faces numerous challenges. A few of the research works, earlier, were done for designing the real-time system. This optimal solution was not available for the machine learning method and needed manual features. The DL method, on the other hand, can automatically extract the features and offers improved detection, but its processing cost is very high and handles noisy images; hence, a big dataset is needed for model training. Accuracy and consistency of labeled data are paramount [35] Vedika B et al (2024). Combining edge detection algorithms with AI models requires some advanced-level balancing techniques to balance their pros and cons. Moreover, the system should be generalizable to various kinds of fractures and imaging conditions. Numerous papers have been published, [36] Kitamura et al (2019) designed an ensemble learning convolutional neural network model for detecting ankle breaks, with an accuracy of 81%. [37] Sai Charan M et al (2024) developed a Hybrid Yolo NAS model using X-ray images to identify hand bone breaks. [38] Tanzi et al, (2020) utilized the InceptionV3 model to categorize X-ray images of the proximal femur, achieving an 86% accuracy rate in distinguishing between healthy and damaged femurs. [39] Mohamed et al, (2023) created a model using deep learning for ResNet50 to pinpoint pelvis bone break in scan images, leveraging convolutional layers and receptor fields to address the classification problem with enhanced GPU performance. In this context, [40] Beyaz et al, (2020) used deep ConvNet to extract features or derive attributes from medical imaging dataset, achieving validation accuracy of 83%. [41] Jones et al, (2020) used the same dataset as Beyaz et al in Ref [40], employed deep ConvNet to extract features, improving the result with validation accuracy of 97.4%. [42] Ma et al, (2020) detected bone breaks using a two-stage process initially, a faster R-CNN was employed to pinpoint twenty bone breaks positions, followed using a specialized CrackNet for classifying the fractures. Their method achieved a 90.11% accuracy in distinguishing between normal and fractured bones. [43] Kitamura et al, (2019) An ensemble-based deep convNet model was created to detect ankle fractures, using ResNet for attribute selection. This approach successfully identified 81% of both normal and fractured bones.

2.4 Explainable AI Application in Medical Images

Explainable AI (XAI) [46] is a method that aims to clarify the reasoning behind the decisions made by machine learning or deep learning algorithms. In this context, certain studies focus on understanding why a model predicts a result as either a normal bone or a fractured one. It is employed to improve the model's transparency, providing insights into the reasoning behind its conclusions and the factors influencing its decisions. In this regard, [44] Dupuis et al, (2022) developed the Rayvolve, an Al Al-powered tool that employs a deep learning model for detecting bone breaks in kids. Their outer validation method accurately classified normal and damaged bones with a 95% accuracy rate. This paper, [45] Shahab S B et al (2023) explores the advancements in Explainable AI (XAI) for improving clarity, equity, and understandability in AI powered decision frameworks, particularly in healthcare. It presents novel XAI solutions validated in clinical scenarios, aiming to enhance AI integration in medical practice. Similar study, [46] Guang Yang et al (2022) reviews the progress of Explainable AI (XAI) in addressing the absence of transparency in AI systems, especially in DL models used in healthcare. It highlights the challenges of integrating AI into clinical practice due to insufficient explainability and presents novel XAI solutions validated through multi-modal, multi-centre data in clinical scenarios, demonstrating the potential of these XAI methods to improve AI integration in broader clinical applications. Fractured bones often exhibit a noticeable change in curvature near the injury site. This local characteristic can be effectively used for identification. In this study, we will use geometric concepts, including a real-time hybrid deep ConvNet system with a canny edge detector, to enhance accuracy and improve the model's performance.

In summary, these papers contribute to a deeper understanding of bone fracture detection, offering useful knowledge and systematic strategies. Nonetheless, more comprehensive and practical research is needed to close the gap between theoretical concepts and real-world implementation, enabling healthcare professionals to make well-advised and thoughtful decisions when implementing these methods in different medical areas.

Chapter 3: Methodology and Materials

3.1 Design Flowchart

The flowchart, illustrated in Figure 3.1, outlines the comprehensive process involved in developing the bone fracture detection model. The initial stage is data acquisition from medical imaging sources, specifically X-ray images. The raw images captured are digitized, cleaned, and pre-processed to ensure quality and consistency. This systematic approach ensures the development of a reliable bone fracture detection model.

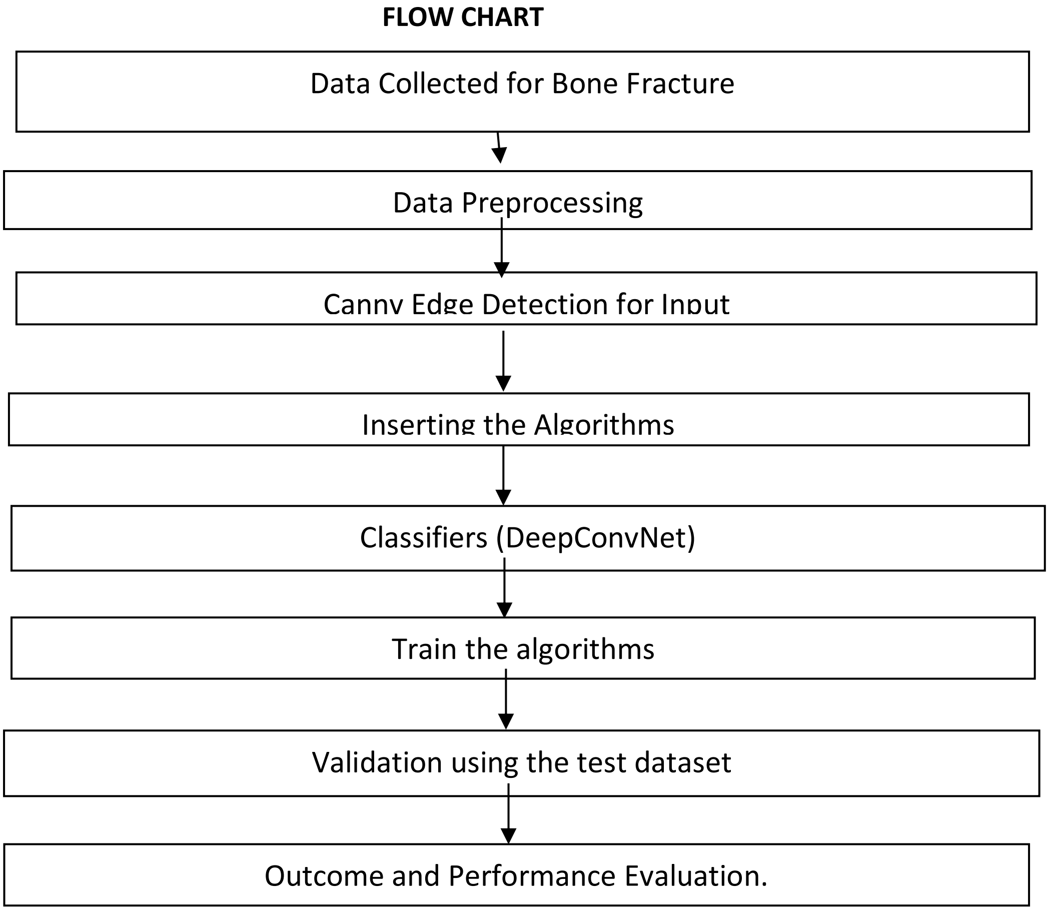

FLOW CHART
Data Collected for Bone Fracture
Data Preprocessing
Canny Edge Detection for Input
Inserting the Algorithms
Classifiers (DeepConvNet)
Train the algorithms
Validation using the test dataset
Outcome and Performance Evaluation.

Figure 3.1: The model's design process flowchart

This study involved developing four binary classifications Deep ConvNet models. The first and fourth models had similar architectures, but the fourth model incorporated canny edge detection. The second model was a ResNet with a total of 7 blocks, consisting of 5 residual blocks, 2 fully connected layers, 1 dense layer, and 1 output layer. The third model, the VGG16 model to create a feature extractor, which was used to extract features from the images in the datasets. The labels were reshaped for SVM compatibility, and PCA was applied to low the dimensionality of the features. Grid Search CV was then used to fine-tune the SVM classifier, optimizing hyperparameters such as C and the kernel. The first model had 6 blocks, consisting of 3 convolutional blocks and 3 fully connected layers (2 dense layers and 1 output layer), while the fourth model had 7 blocks, with 4 convolutional blocks and 3 fully connected layers (2 dense layers and 1 output layer), incorporating Canny edge detection to improve effectiveness. All models were trained on a dataset specifically collected and organized for this study, and their effectiveness was evaluated using multiple performance metrics on the test images.

The processed data was then split into three subsets: 87% for training, 8% for validation, and 5% for reserved testing. The images were normalized, resized to 224x224 pixels, and managed in batches. Each generator was configured to handle data specifically for training, testing, and validation, with appropriate settings for shuffling and class mode tailored to the three models. The models were trained and validated using the training and validation datasets, and predictions were made by testing the models on previously unseen images from the test set. The models' effectiveness was then assessed using key metrics like precision, recall, F1-score, and accuracy, all based on the predictions from the test set.

3.2 Data Collection

In bone fracture detection, accurate data is obtained by capturing and recording medical images. In a clinical setting, images collected through X-rays or CT scans are crucial for healthcare professionals to analyze, diagnose, monitor, and treat medical conditions. These images are stored as digital files in a hospital’s imaging system, allowing for easy access, comparison, and sharing among healthcare providers. X-rays produce two-proportional images, whereas CT scans provide comprehensive three-proportional views of the body. By converting physical structures into digital images, X-ray and CT technology enable precise and informed medical care. [47] Laura Marie Fayad, (2024). Bone X-rays are useful for detecting fractures, while chest X-rays help diagnose lung conditions. Dental X-rays reveal issues such as tooth decay and bone loss. CT scans, using X-rays, produce detailed 3D images by capturing various views of the body. CT scans are also used to monitor treatment effectiveness, such as assessing the size of a tumor after chemotherapy. The collected data is then digitized, cleaned, and pre-processed to ensure it is suitable for deep learning or machine learning model development. In this research, the dataset was created with the aim to build X-ray images of bone fractures. The datasets used for this research include one for healthy bone conditions and the last dataset for fracture conditions. For each set, the four experimental trials for analysis and model training and were used for each to make comparisons in their performance. The link to the dataset: https://universe.roboflow.com/sivathmika-c-7j9xt/bonefracturedataset_siva/dataset/1

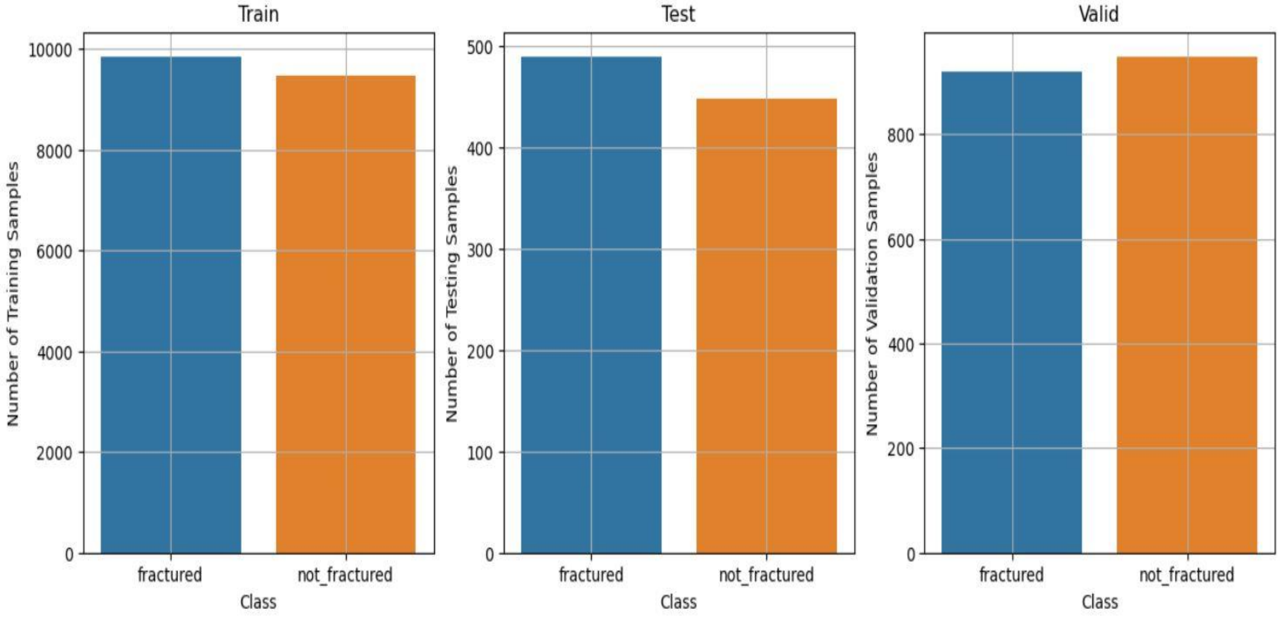


Figure 3.2: Fractured and Non-Fractured Samples Across Training, Testing, and Validation Sets.

Chart 3.2 involves the data collection process, where two classes of bone fracture X-ray images were used to align with the goals of the proposed approach for binary classify in this research. Fractured images are represented by blue bars, while non-fractured images are represented by orange bars. The dataset collected for this project is organized into a main folder that includes subfolders for training, validation, and testing. Each subfolder contains separate files for fractured and non-fractured images. All X-ray images are in JPG format, with an original resolution of 640x640 pixels, stretching them to fit these dimensions. This ensures that the input size is consistent across the dataset, which is crucial for the neural network's performance. However, stretching might distort the images slightly since it alters the original aspect ratio. Some commonly used methods include applying auto-orient transformations to the images based on their metadata, ensuring that all images are properly aligned before undergoing further processing and no classes were altered or removed during preprocessing, meaning all existing classes remained unchanged, and no new class labels were introduced, and images are randomly rotated between -12 and +12 degrees.

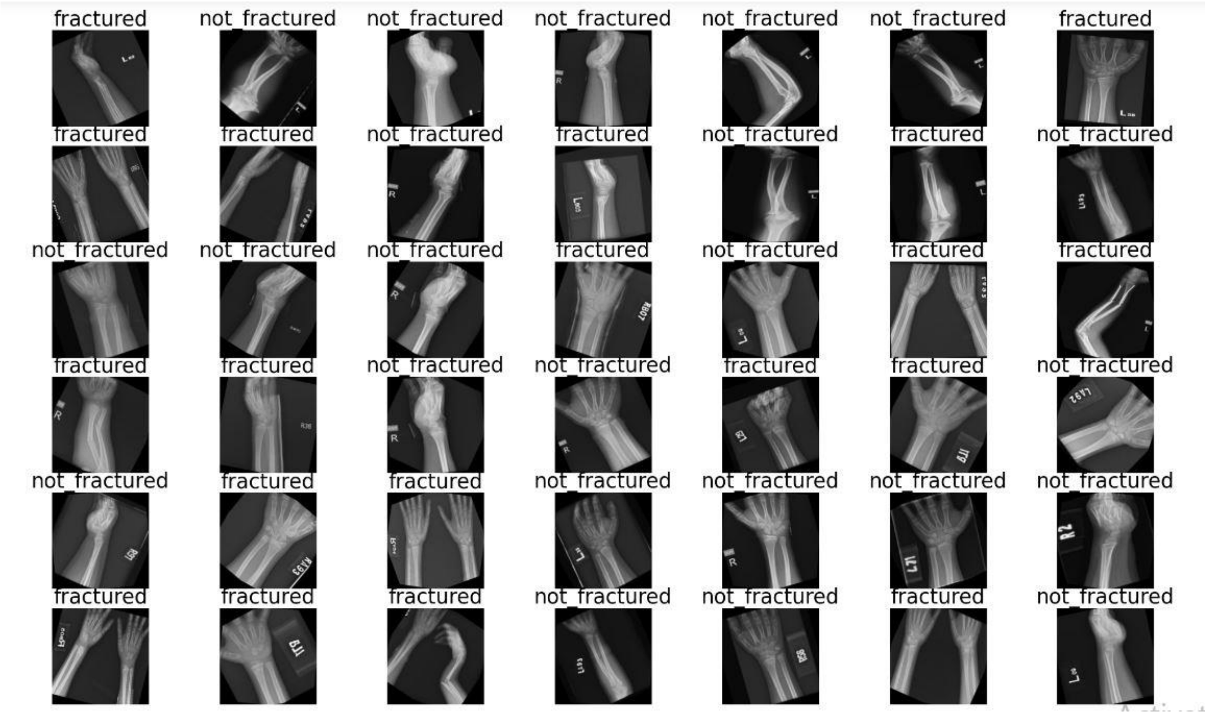


Figure 3.3: X-Ray Classification of Fractured and Healthy Bones

This augmentation simulates slight variations in image capture angles, which helps the model learn to recognize the target features regardless of their orientation. These preprocessing steps aim to standardize the input data, enhance the variability of the training set, and enhance the model's ability to handle various image positions and distortions.

3.3 Data Preprocessing

The image data collected requires thorough preprocessing before it can be used as input for the model. This is an essential stage because the accuracy of the model could be significantly impacted by a poorly pre-processed data set. At this stage, the image dataset is checked for various issues such as inconsistent resolutions, poor image quality (e.g., blurry or overexposed images), and incorrect colour spaces. Normalization of pixel values is performed to ensure consistency across the dataset, and any long or improper file names are renamed to maintain consistency and clarity, especially when labelling images during the data visualization or plotting stages. However, most of the work done in this preprocessing stage is on feature extraction. The features of the data collected require some form of conversion to represent the discriminatory indicators of the various types of bone fractures. Here is the data pre-processed required for this study, firstly pre-processed by removing any kind of noise through filters, then using techniques such as image augmentation (rotation, scaling, and flipping) for detecting edges based on an improved canny edge algorithm [17] Khaled et al (2023). Canny edge detection is a specialized method employed to detect the boundaries. within an image, which is particularly useful in highlighting the boundaries of different structures within the body, such as organs, bones, or lesions. This method enhances the clarity and definition of important features, making them easier to analysed. Effective preprocessing, including canny edge detection, is essential for reliable and accurate interpretation of medical images, leading to better diagnosis and treatment planning.

Data preprocessing in medical imaging is a critical stage that enhances the standard and usability of the data before it is analysed or used for diagnostic purposes. This process involves several stages, including noise reduction, normalization, segmentation, and canny edge detection, to improve the accuracy of the subsequent analysis. Noise reduction techniques, such as filtering, help remove artifacts and irrelevant information from the images. Normalization ensures that the data is consistent and comparable across different images by adjusting pixel values. Segmentation involves dividing the image into regions or structures of interest, such as separating tissues from bones, to focus on specific areas relevant to diagnosis.

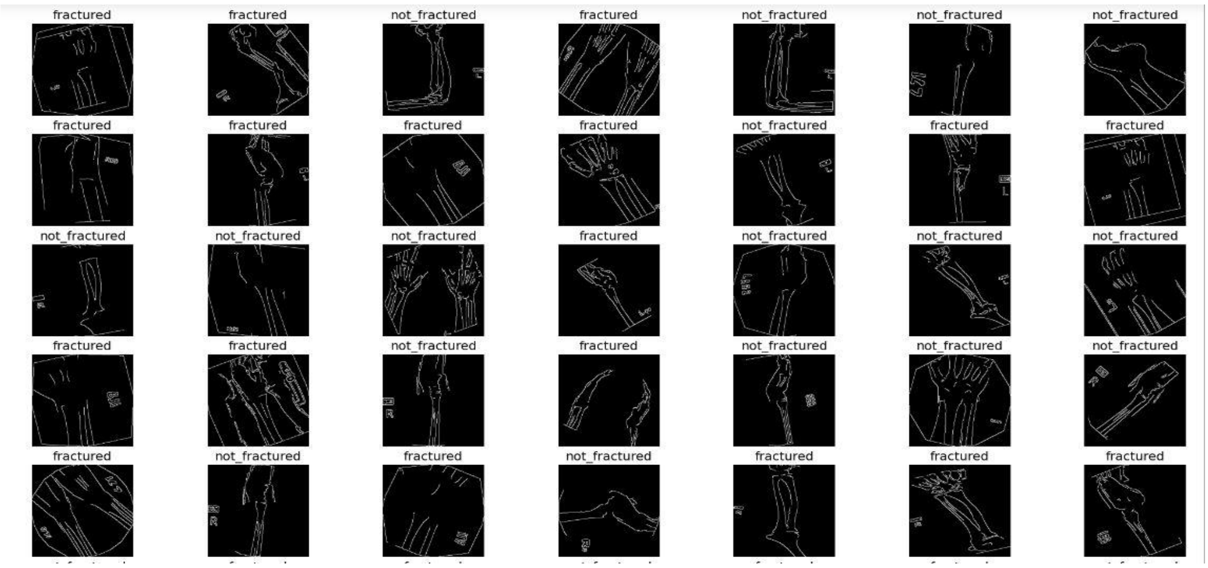


Figure 3.4: Canny Edge Detection Visualization of Fractured and Non-Fractured Bones

In the context of this project, these functions are used to prepare bone fracture X-ray images before processing them into the deep learning model. By resizing the images and applying canny edge detection, the preprocessing enhances relevant features (like fractures) and standardizes the input size, improving the model's capability to learn and generate accurate classifications. The canny edge detection emphasizes the edges, which is particularly useful for identifying fractures in X-ray images, as it highlights discontinuities and structural changes in the bone.

3.4 Model Formulation

This is the crucial stage where the choice of the model to implement is made. The choice of the model is dependent based on the nature of the problem the research intends to solve. Choosing an appropriate deep learning or machine learning model is pivotal for accurate prediction of fractures in the bone. Obviously, when the term bone fracture detection is mentioned, the first model that comes to mind is a classification model. The bone break is categorized into two types such as healthy and fractures. For such a classification problem, there are many deep learning or machine learning models in use today that accurately perform fracture detection and classification. Four models were designed to classify bone fracture X-ray images into two categories: healthy and fractured. The models were named Deep ConvNet, ResNet, VGG16-SVM and Deep ConvNet with canny edge detection model. The CNN algorithm classifies images by identifying key features, objects, and extracting relevant information from the input data, assigning importance to them. Additionally, it offers significant advantages, such as faster processing and being more cost-effective due to reduced computational effort.

The fundamental architecture of the deep ConvNet models with integrated canny edge detection, developed for this project. The models are composed of two main components: the feature extraction module serves as the initial part, responsible for identifying patterns and features within the input data and the classification module the input images are classified into binary categories using the outputs from the feature extraction stage, which are passed through a flattened layer.

The deep convNet utilized in this project leverages various critical components convolutional layers, ReLU activation functions, batch normalization, max-pooling, and dropout techniques to enhance feature learning, improve performance, and reduce overfitting. Below is a summary of how each of these components contributes to the model's effectiveness:

Convolutional Layers are fundamental to the deep convNet architecture, developed to instantly learn and extract spatial features from input images. They utilize filters that move across the input, capturing various patterns such as boundaries, textures, and shapes, which are crucial for accurate image classification. This feature extraction process is vital for the model's ability to recognize and distinguish between various classes, serving as the foundation of the CNN.

The Rectified Linear Unit (ReLU) activation function is applied in convolutional layers to add non-linearity, enabling the network to capture more intricate data representations. ReLU effectively addresses the vanishing gradient problem by setting all negative values to zero, ensuring that the model trains faster and learns more efficiently. Its simplicity and efficiency make ReLU a preferred choice, improving the model’s ability to capture intricate trends. Batch normalization is implemented after convolutional layers to normalize the activations, stabilizing the learning process and accelerating training. By normalizing the input to each layer, batch normalization reduces internal covariate shift, ensuring that the data remains within a stable range. This leads to faster convergence, improved performance, and increased robustness of the model, making it less sensitive to the initial weights and learning rate.

Max-pooling is a down-sampling technique used to reduce the spatial dimensions of the feature maps while preserving the most salient features. It works by selecting the maximum value within a defined window (typically 2x2), which helps to retain important features while substantially decreasing the number of parameters. This process not only makes the model more computationally efficient but also helps in attaining spatial invariance, allowing the model to identify patterns regardless of their location within the image

The dropout technique is a regularization method used to prevent overfitting by randomly disabling a portion of neurons during each training iteration. By setting a specific dropout rate (e.g., 30%), the model is forced to learn redundant and robust features, making it less reliant on specific pathways and thus more generalizable. This enhances the model's effectiveness on unobserved data, contributing to a more reliable and effective classification process.

Instead of flattening the feature maps, the Global Average Pooling layer calculates the mean value of each feature map, decreasing the data size while retaining significant spatial information. This method is more efficient and often works well for classification tasks. Residual blocks are a feature of ResNet (Residual Networks) and allow very deep networks to be trained more effectively. They help avoid the vanishing gradient problem by using shortcut connections. A fully connected layer connects each input neuron to every output neuron, and it is typically used at the end of the network to integrate all the learned features and make a final prediction

These components work synergistically to form a powerful feature extraction and classification framework within the CNN architecture. Convolutional layers focus on feature learning, ReLU introduces essential non-linear transformations, batch normalization stabilizes training, max pooling reduces data dimensionality, and dropout techniques ensure generalization by preventing overfitting. Together, they create a robust and efficient neural network model designed for high performance in binary classification tasks, making the architecture well-suited for real-world applications, such as those explored in medical imaging in this project. In this proposed research, I evaluated the performance of the Deep ConvNet model, ResNet model, VGG16-SVM model, and a hybrid deep ConvNet with canny edge detection. All these models were created from scratch specifically for this experiment. The presented models can be applied for real-time bone detection in humans through the development of web and mobile applications to assist doctors.

3.4.1 Deep ConvNet Model

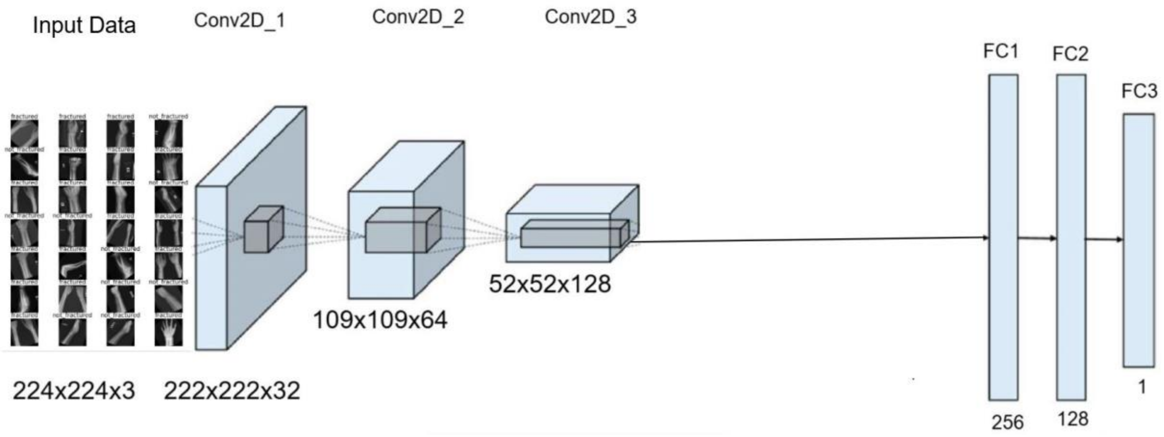
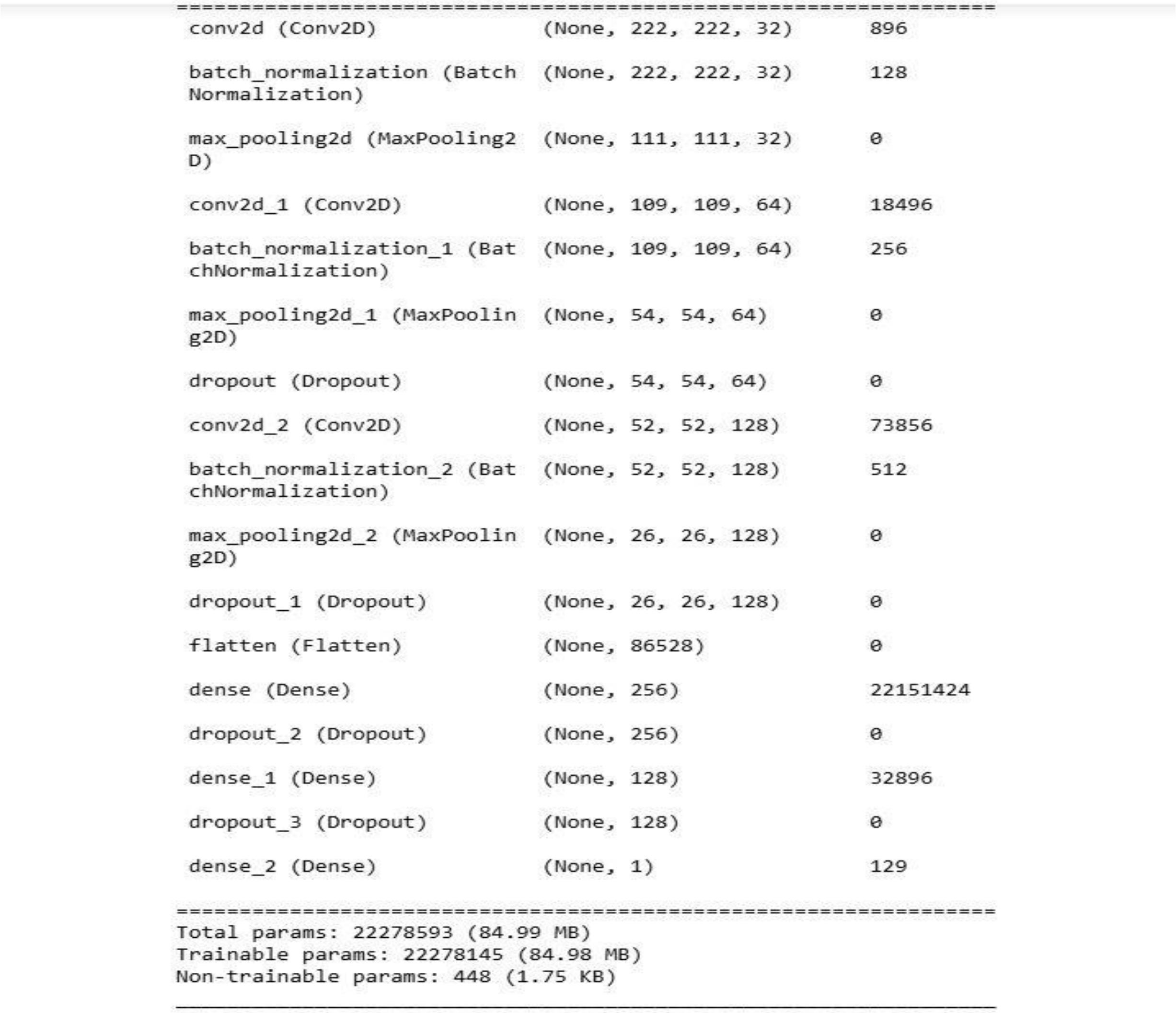


Figure 3.5: Deep ConvNet Architecture Detection

The first model, called Deep ConvNet, was created from scratch for this experiment, utilizing a basic convolutional neural network architecture. It follows a sequential structure with multiple convolutional, pooling, and dense layers that allow the model to learn complex patterns from the input data. Below is a detailed explanation of each component in the model. The Detect Deep ConvNet model's design features an increasing number of neurons in its convolutional layers, starting with 32 neurons in the first layer, 64 neurons in the second and 128 neurons in the final convolutional layer. Each MaxPooling layer consistently uses a 2x2 window size across all blocks, helping to down sample the feature maps and reduce the computational load while retaining critical features. The input pre-processed X-ray images were resized to 224 x 224 pixels, with 3 channels indicating that the images are in RGB format. The first layer uses a 3x3 filter with 32 filters and applies padding set to 'valid' by default, meaning no padding is added around the input image. As a result, the dimensions reduce by 1 pixel on each side, leading to an output shape of 222 x 222 pixels with 32 feature maps. Batch normalization does not change the spatial dimensions but normalizes the output stabilizes the training by normalizing the output of the convolutional layer, so the shape remains the same. The max pooling operation with a 2x2 window in each layer reduces the spatial dimensions by half, resulting in an output shape of 111 x 111 pixels in the first layer. In the second layer, further feature extraction is performed with 64 filters, with dropout added for regularization. Finally, deeper feature extraction is achieved with 128 filters, with additional dropout applied to prevent overfitting in the later layers.

In the final stage of the convolutional neural network (CNN), the model transitions from feature extraction through convolutional layers to decision-making using fully connected (dense) layers. These layers are essential for transforming the learned features into the final output prediction. The Flatten layer reshapes the 3D feature maps from the last convolutional layer into a 1D vector, converting them into a single flat array of size 86,528 for input into the fully connected layers. The fully connected layers serve as the final decision-making component of the model. After the convolutional blocks extract spatial and hierarchical features from the input images, these dense layers synthesize and interpret the data to produce the final classification output. Dropout is used to ensure that the model remains generalizable, while the combination of dense layers enables sophisticated pattern recognition, making this architecture effective for image classification tasks. This model is designed for a binary classification task, likely on images of size 224x224. It uses convolutional layers to extract features from the images, pooling layers to down-sample the data, and fully connected layers to make final predictions. Dropout and batch normalization help to avoid overfitting and improve performance. The final layer uses a Sigmoid activation to output a probability, suitable for binary classification.



```
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 batch_normalization (Batch  (None, 222, 222, 32)      128
 Normalization)

 max_pooling2d (MaxPooling2  (None, 111, 111, 32)      0
 D)

 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496

 batch_normalization_1 (Bat  (None, 109, 109, 64)      256
 chNormalization)

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0
 g2D)

 dropout (Dropout)           (None, 54, 54, 64)        0

 conv2d_2 (Conv2D)           (None, 52, 52, 128)       73856

 batch_normalization_2 (Bat  (None, 52, 52, 128)       512
 chNormalization)

 max_pooling2d_2 (MaxPoolin  (None, 26, 26, 128)       0
 g2D)

 dropout_1 (Dropout)         (None, 26, 26, 128)       0

 flatten (Flatten)           (None, 86528)             0

 dense (Dense)               (None, 256)               22151424

 dropout_2 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 128)               32896

 dropout_3 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 1)                 129

=================================================================
Total params: 22278593 (84.99 MB)
Trainable params: 22278145 (84.98 MB)
Non-trainable params: 448 (1.75 KB)
_________________________________________________________________
```

Figure 3.6: The architecture of the deep ConvNet mode.

The model has a total of 22,278,593 parameters, with most concentrated in the dense layers, highlighting its high learning capacity. Out of these, 22,278,145 are trainable parameters that are adjusted during training to minimize the loss, while 448 are non-trainable parameters, likely from batch normalization layers, and remain fixed during training.

3.4.2 ResNet Model

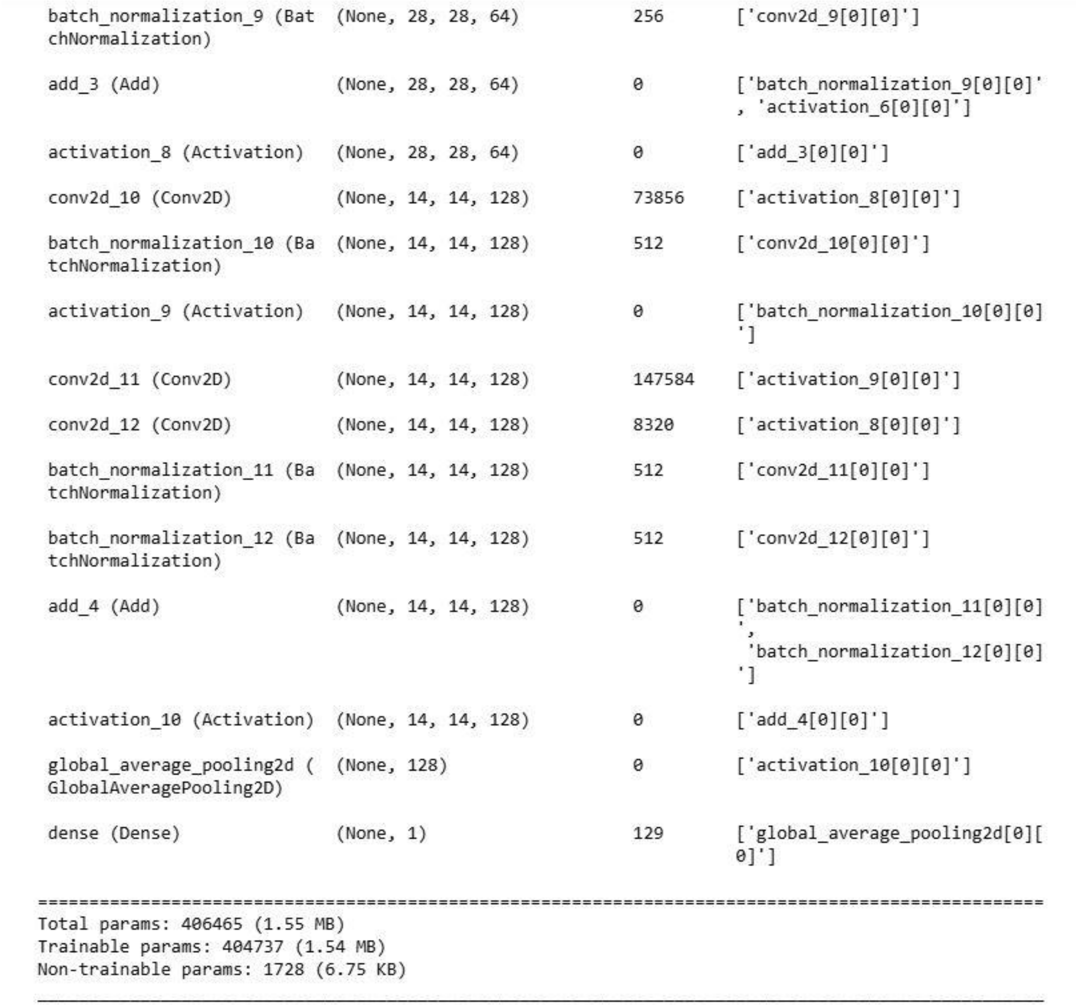
In the suggested approach, ResNet is a deep convNet created for image classification purposes. The architecture integrates several key components, such as convolutional layers, residual blocks, batch normalization, and pooling, creating an efficient and powerful model for image-based tasks. The shape of this input layer is (None, 224, 224, 3), where "None" allows for variable batch sizes, offering flexibility when training with datasets of different sizes.

The output shape after this operation is (112, 112, 32) due to down sampling from the convolution operation. This layer includes 4,736 parameters, which encompass the weights and biases of the filters. Following the convolution operation, a batch normalization layer is introduced to normalize activations and improve training stability. The Rectified Linear Unit (ReLU) activation function is then applied to introduce non-linearity into the model. Finally, max pooling is used to reduce the spatial dimensions by half, resulting in a shape of (56, 56, 32). The next section of the model consists of residual blocks, which are inspired by ResNet architectures. These blocks utilize skip connections that allow the network to bypass certain layers, which helps alleviate the vanishing gradient problem, a frequent challenge in deep network. Within this block, multiple convolutional layers with 32 filters are applied, each followed by ReLU activation and batch normalization. The Add layer is a key component, as it merges the input with the output of subsequent convolutions, preserving information from earlier layers. This residual structure is repeated multiple times at the 56x56 resolution level, ensuring that feature maps retain crucial information through the network’s depth.

After the residual blocks, the model downscales the resolution to 28x28. In this block, 64 filters are applied, and the process of convolution, batch normalization, and ReLU activation continues. Again, skip connections are incorporated to further assist in preserving information as the network deepens. The output at this stage becomes more detailed, as more filters allow the model to capture higher-level features from the images. The third convolutional block operates at a reduced resolution of 14x14, with 128 filters applied at this stage. This block follows the same structure as the previous ones, with convolutions, batch normalization, activations, and skip connections. However, the complexity of the skip connections increases as the number of filters grows, allowing the model to capture more complex and abstract features from the input images. This progression towards more filters enables the model to handle more intricate patterns in the data. A key feature of this model is the Global

Average Pooling 2D layer, which reduces the entire feature map into a single 128-element vector by averaging across all spatial dimensions. This technique efficiently condenses the information extracted by the convolutional layers, while avoiding the need for fully connected layers with many parameters. Global average pooling is particularly useful for image classification tasks because it helps prevent overfitting and reduces the overall model size.

The final layer of the model is a dense (fully connected) layer that maps the 128-element vector produced by the global average pooling layer to a single output neuron. This output is likely used for binary classification, where a single value is produced that can be interpreted as a probability (if paired with a sigmoid activation function). The entire dense layer contains 129 parameters, suggesting that the classification decision is made with minimal overhead after the complex feature extraction stages



```
 batch_normalization_9 (Bat  (None, 28, 28, 64)          256       ['conv2d_9[0][0]']
 chNormalization)

 add_3 (Add)                 (None, 28, 28, 64)          0         ['batch_normalization_9[0][0]'
                                                                    , 'activation_6[0][0]']

 activation_8 (Activation)   (None, 28, 28, 64)          0         ['add_3[0][0]']

 conv2d_10 (Conv2D)          (None, 14, 14, 128)         73856     ['activation_8[0][0]']

 batch_normalization_10 (Ba  (None, 14, 14, 128)         512       ['conv2d_10[0][0]']
 tchNormalization)

 activation_9 (Activation)   (None, 14, 14, 128)         0         ['batch_normalization_10[0][0]
                                                                   ']

 conv2d_11 (Conv2D)          (None, 14, 14, 128)         147584    ['activation_9[0][0]']

 conv2d_12 (Conv2D)          (None, 14, 14, 128)         8320      ['activation_8[0][0]']

 batch_normalization_11 (Ba  (None, 14, 14, 128)         512       ['conv2d_11[0][0]']
 tchNormalization)

 batch_normalization_12 (Ba  (None, 14, 14, 128)         512       ['conv2d_12[0][0]']
 tchNormalization)

 add_4 (Add)                 (None, 14, 14, 128)         0         ['batch_normalization_11[0][0]
                                                                   ',
                                                                    'batch_normalization_12[0][0]
                                                                   ']

 activation_10 (Activation)  (None, 14, 14, 128)         0         ['add_4[0][0]']

 global_average_pooling2d (  (None, 128)                 0         ['activation_10[0][0]']
 GlobalAveragePooling2D)

 dense (Dense)               (None, 1)                   129       ['global_average_pooling2d[0][
                                                                   0]']

==================================================================================================
Total params: 406465 (1.55 MB)
Trainable params: 404737 (1.54 MB)
Non-trainable params: 1728 (6.75 KB)
__________________________________________________________________________________________________
```

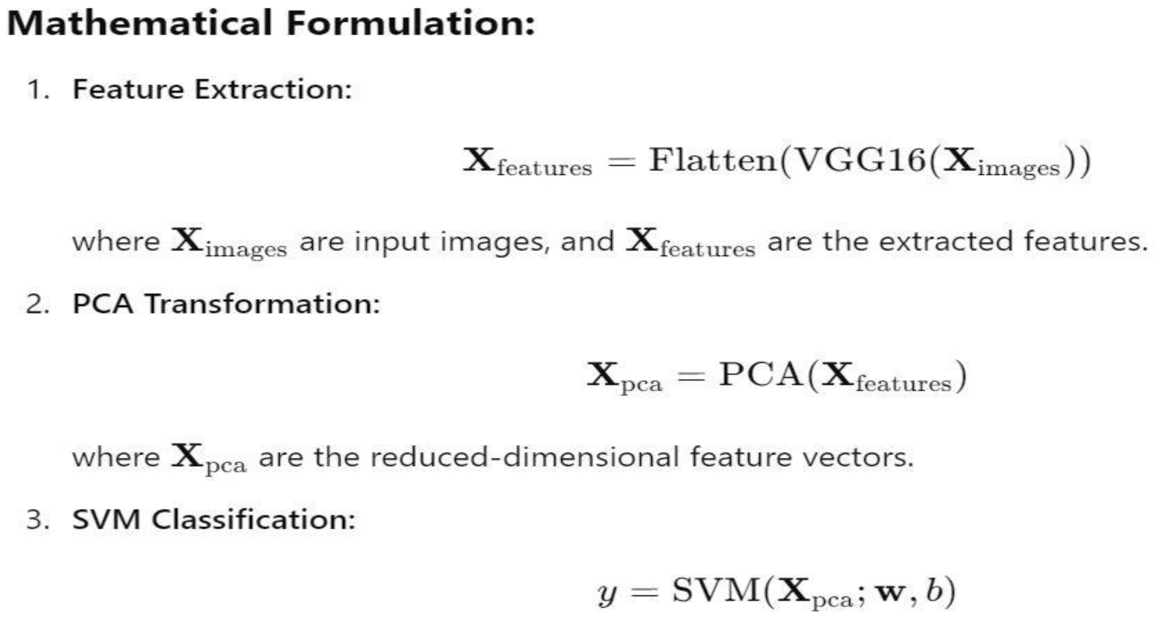
Figure 3.7: The architecture of the ResNet mode.

In total, the model consists of approximately 406,465 parameters, of which 404,737 are trainable, while 1,728 are non-trainable. This efficient parameterization allows the model to balance complexity with performance.

3.4.3 VGG16-SVM Hybrid Model

The VGG16-SVM combination, was pre-trained on ImageNet and utilized a linear kernel with a regularization parameter. It employed 5-fold cross-validation to determine the best-performing model. The model formulation involves three main components: Feature Extraction using VGG16, a deep learning model (CNN) pre-trained on ImageNet; Dimensionality Reduction using PCA, a statistical method that reduces the feature space; and Classification using SVM, a traditional machine learning algorithm optimized through hyperparameter tuning. Feature extraction with VGG16 is used in the model, which consists of 16 layers including convolutional layers, pooling layers, and fully connected layers. The model utilizes weights trained on ImageNet, enabling it to act as a powerful feature extractor for images by capturing complex patterns such as edges, textures, and shapes. This feature extraction process converts raw images into high-dimensional feature vectors that represent the content of each image in a form suitable for classification.

Initially, I ran the model without integrating PCA, but the computer struggled due to its limited memory capacity, as the RAM was insufficient to handle the size of the dataset. To address this, Principal Component Analysis (PCA) was employed to reduce the dimensionality of the high-dimensional feature vectors extracted by VGG16. The feature vectors generated by VGG16 are large, making SVM training computationally expensive and prone to overfitting. PCA reduces the number of features while retaining most of the variance in the data. It transforms the feature vectors into a lower-dimensional space with 12,000 components, thereby reducing computational complexity while preserving critical information. SVM model is a supervised learning algorithm that identifies the optimal hyperplane separating data points of different classes in a transformed feature space. It utilizes kernel functions (linear and RBF in this case) to map input data into a higher-dimensional space, where a separating hyperplane can be more easily found. Hyperparameter Tuning with GridSearchCV performs cross-validation across different parameter values to find the best combination. The SVM model learns to classify images based on PCA-reduced feature vectors, aiming to minimize classification error on unseen data. The trained SVM model is evaluated on validation and test datasets using metrics such as accuracy, confusion matrix, classification report, and ROC curve analysis.



Mathematical Formulation:

1. **Feature Extraction:**

$$\mathbf{X}_{\text{features}} = \text{Flatten}(\text{VGG16}(\mathbf{X}_{\text{images}}))$$

   where $\mathbf{X}_{\text{images}}$ are input images, and $\mathbf{X}_{\text{features}}$ are the extracted features.

2. **PCA Transformation:**

$$\mathbf{X}_{\text{pca}} = \text{PCA}(\mathbf{X}_{\text{features}})$$

   where $\mathbf{X}_{\text{pca}}$ are the reduced-dimensional feature vectors.

3. **SVM Classification:**

$$y = \text{SVM}(\mathbf{X}_{\text{pca}}; \mathbf{w}, b)$$

Figure 3.8: The mathematical architecture of the VGG16-SVM mode.

where w and b are the weights and bias parameters of the SVM that define the decision boundary. The formulation involves extracting meaningful features using VGG16, compressing these features using PCA, and classifying them using an optimized SVM. This hybrid model leverages deep learning for feature extraction and SVM's robustness for classification, providing a powerful approach to solving the image classification problem.

3.4.4 Deep ConvNet Integrated with Canny Edge Detection

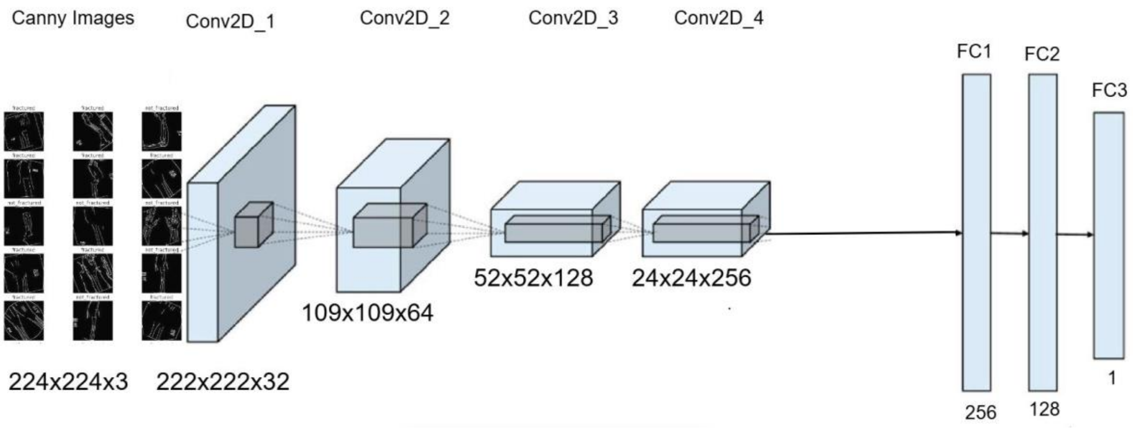
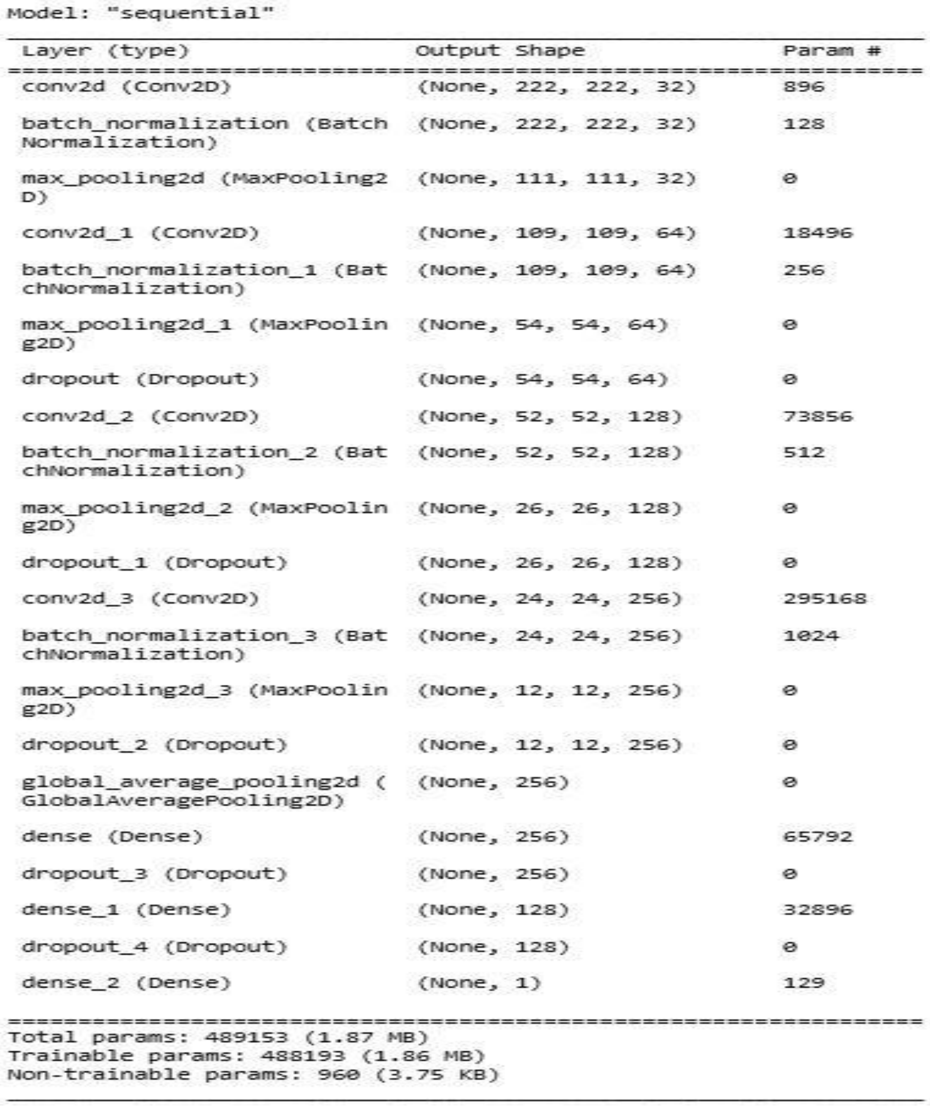


Figure 3.9: Deep ConvNet with Canny Edge Detection Architecture

The presented model is a Deep ConvNet with canny edge detection designed for image analysis tasks such as classification. The model takes images pre-processed with canny edge detection as input, emphasizing key structural features like edges and boundaries. It reduces irrelevant noise, improving generalization, particularly with limited or noisy data and edge-focused inputs help the model learn from identifiable features, making predictions more understandable and explainable. The preprocessing function reads an image, converts it to grayscale, resizes it to 224x224 pixels, and applies canny edge detection. This technique highlights prominent edges in the image, converting them to an RGB format for consistency with the input expectations of CNNs. The edge detection step simplifies the input by focusing on structural features, helping the model concentrate on relevant patterns.

The model receives a 224x224x3 input containing the edge-detected features. As a result, the dimensions reduce by 1 pixel on each side, leading to an output shape of 222 x 222 pixels with 32 feature maps. This preprocessing step allows the model to focus on the critical contours and boundaries within the image, facilitating improved feature extraction. The first layer uses 32 filters of size (3x3) to extract basic patterns from the image, such as lines and simple shapes. The edge-enhanced input allows this layer to effectively capture the most relevant details right from the start. Batch normalization helps stabilize training and ensures that the model learns more effectively from the simplified, edge-focused input. Max pooling reduces the spatial dimensions of the feature maps (from 222x222 to 111x111), which helps in reducing computational complexity and focuses the model on prominent features. Dropout layers randomly drop some neurons during training, preventing overfitting and helping the model generalize better, which is especially important when dealing with highlighted edge features.

As the model advances to deeper layers, the number of filters increases (64 and 128). allowing it to capture more complex patterns and relationships in the edge-detected images. The sequential nature ensures that only the most significant features are preserved and refined at each stage. Flatten layer transforms the 3D feature maps into a 1D vector, organizing it for the dense layers which will classify or regress the features extracted from the edge-enhanced inputs. Dense layers perform high-level reasoning based on the abstracted features from earlier layers. The presence of dropout between these dense layers further mitigates overfitting and enhances robustness. The final dense layer outputs the prediction, which could be a single value for regression or a probability score for binary classification. The preceding architecture, coupled with edge-focused preprocessing, enables the model to make more precise predictions by leveraging structural information effectively. Integrating canny edge detection in the preprocessing pipeline significantly enhances the model’s ability to learn, generalize, and interpret image features, resulting in more accurate and reliable performance. The model progressively extracts features from the canny edge input image, followed by batch normalization to stabilize and accelerate training, max pooling layers to reduce spatial dimensions, and dropout layers to prevent overfitting. After feature extraction, the model uses global average pooling to reduce the data size without losing important information. It then includes two fully connected (dense) layers with L2 regularization, the first dense layer contains 256 units, while the second dense layer contains 128 units. These fully connected layers allow the model to combine the advanced features captured by the convolutional layers and make decisions based on them. Dropout is again applied after each dense layer to further reduce the risk of overfitting. The final layer is a single neuron with a sigmoid activation function, which outputs the probability for binary classification. The model is compiled using the Adam optimizer with binary cross-entropy loss and tracks accuracy during training.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 batch_normalization (Batch  (None, 222, 222, 32)      128
 Normalization)

 max_pooling2d (MaxPooling2  (None, 111, 111, 32)      0
 D)

 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496

 batch_normalization_1 (Bat  (None, 109, 109, 64)      256
 chNormalization)

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0
 g2D)

 dropout (Dropout)           (None, 54, 54, 64)        0

 conv2d_2 (Conv2D)           (None, 52, 52, 128)       73856

 batch_normalization_2 (Bat  (None, 52, 52, 128)       512
 chNormalization)

 max_pooling2d_2 (MaxPoolin  (None, 26, 26, 128)       0
 g2D)

 dropout_1 (Dropout)         (None, 26, 26, 128)       0

 conv2d_3 (Conv2D)           (None, 24, 24, 256)       295168

 batch_normalization_3 (Bat  (None, 24, 24, 256)       1024
 chNormalization)

 max_pooling2d_3 (MaxPoolin  (None, 12, 12, 256)       0
 g2D)

 dropout_2 (Dropout)         (None, 12, 12, 256)       0

 global_average_pooling2d (  (None, 256)               0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 256)               65792

 dropout_3 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 128)               32896

 dropout_4 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 1)                 129

=================================================================
Total params: 489153 (1.87 MB)
Trainable params: 488193 (1.86 MB)
Non-trainable params: 960 (3.75 KB)
_________________________________________________________________
```

Figure 3.10: The architecture of the Deep ConvNet with Canny Edge detection mode.

The model has a total of 489,153 parameters, with most concentrated in the dense layers, highlighting its high learning capacity. Out of these, 488,193 are trainable parameters that are adjusted during training to minimize the loss, while 960 are non-trainable parameters, likely from batch normalization layers, and remain fixed during training.

3.5 Model Testing

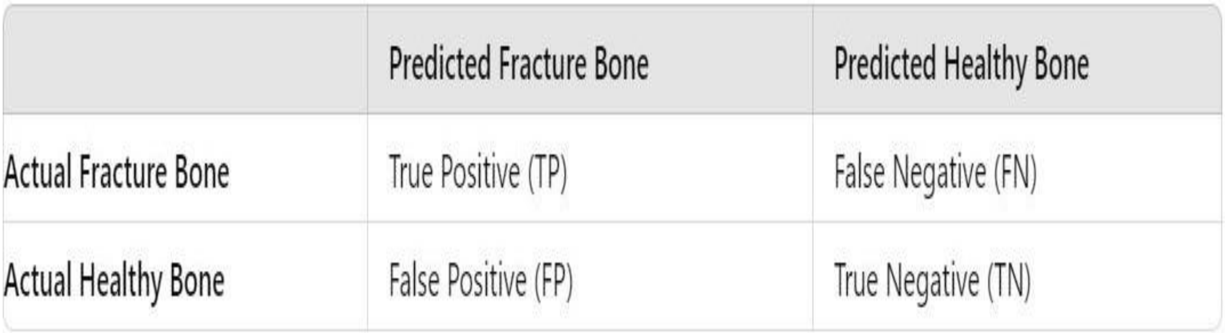
The model will undergo testing using a distinct test dataset that was not involved in the training phase. Typically, 5% of the entire dataset is allocated for this purpose. This test dataset is meticulously prepared to emulate real-world scenarios, encompassing two key conditions: healthy bone states and fracture conditions. It is crucial that the test data is comprehensive and diverse, encompassing various conditions and potential fracture scenarios. This approach ensures that the model's performance can generalized effectively to new and unseen cases, thereby validating its reliability and accuracy in practical applications. Through rigorous testing with this independent dataset, the model's ability to accurately predict and classify healthy bones and fracture conditions will be thoroughly evaluated, confirming its suitability for use in medical diagnostics and fracture assessment.

3.6 Model Evaluation

Model evaluation is an essential and pivotal stage in assessing the efficiency and reliability of the machine learning model designed for detecting and diagnosing bone fractures in medical applications. This essential process involves quantifying the model’s performance through a variety of metrics and methodologies, ensuring that it meets the required standards for accurate diagnosis because precise and reliable predictions are critical for effective fracture management and patient care.

3.6.1 Confusion Matrix

The confusion matrix is a critical model evaluation tool adopted in this research, essential for evaluating the performance of machine learning models, especially in classification tasks like diagnosing bone fractures. This tool provides a clear and concise visualization of the performance of the model’s classification accuracy by showing the accurate and inaccurate predictions in a matrix format. This helps in comprehending not only the model's accuracy but also the nature of the errors it generates. In the context of this research, the confusion matrix used is constructed for a binary classification problem by categorizing bone conditions into two categories: "healthy" and "fracture." The resulting matrix is a 2 x 2 grid where each row represents the actual condition of the bones, and each column represents the predicted condition. The diagonal elements of the matrix indicate accurate predictions, whereas the off-diagonal elements signify misclassifications. This structured representation allows for detailed performance analysis, such as identifying whether healthy bones or fractures are most often misclassified. For example, if the model frequently misclassifies a "fracture" as "healthy," it suggests a need for improved feature extraction or model tuning specific to these conditions. By providing such granular insights, the confusion matrix not only helps in evaluating the overall accuracy but also aids in fine-tuning the model to enhance its predictive capabilities, ultimately contributing to more effective and reliable diagnostic strategies for bone health assessment and fracture management.



	Predicted Fracture Bone	Predicted Healthy Bone
Actual Fracture Bone	True Positive (TP)	False Negative (FN)
Actual Healthy Bone	False Positive (FP)	True Negative (TN)

Figure 3.1: An illustrative example of the Confusion Matrix for a 2-class

True Positive (TP): The model correctly predicts a fracture when it is actually a fracture.

False Positive (FP): The model incorrectly predicts a fracture when the bone is actually healthy.

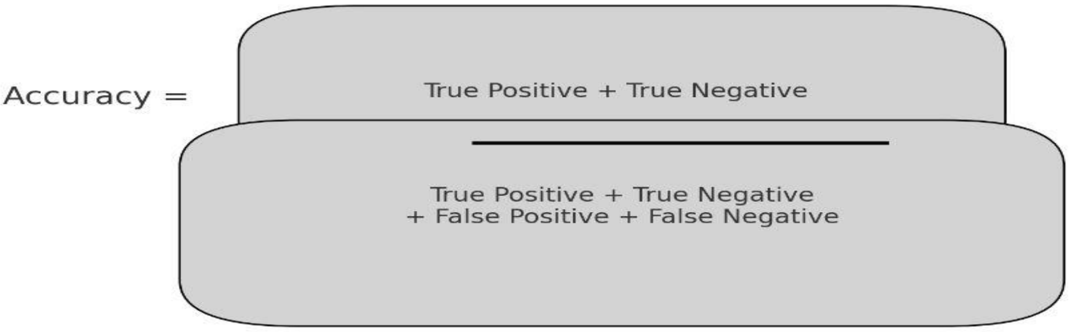
False Negative (FN): The model incorrectly predicts the bone is healthy when it is actually fractured.

True Negative (TN): The model correctly predicts the bone is healthy when it is indeed healthy.

Several performance metrics can be derived from the confusion matrix to provide a comprehensive evaluation of our classification model.

3.6.2 Accuracy

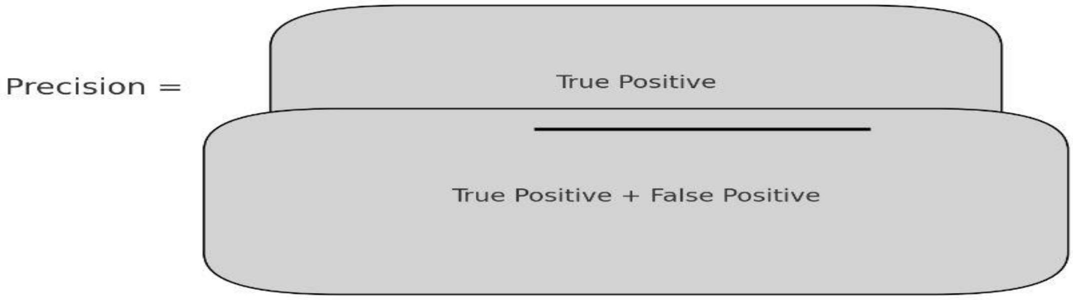
Accuracy is a key metric for assessing the performance of a machine learning model, especially in classification tasks. It indicates the ratio of correctly predicted instances to the total instances in the dataset. It is simple and straightforward and hence a good choice for model evaluation. However, it comes with a limitation of giving a misleading result for an unbalanced dataset. Mathematically, accuracy is defined as:



$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

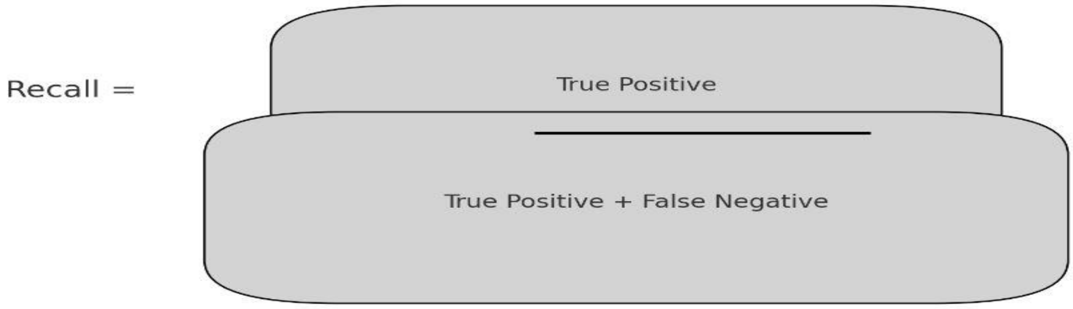
3.6.3 Precision

Precision measures the accuracy of the model's positive predictions. For fractures (class 0), 99% of the predicted instances are correctly classified as fractures. For healthy cases (class 1), 95% of the predicted instances are correctly identified as healthy.



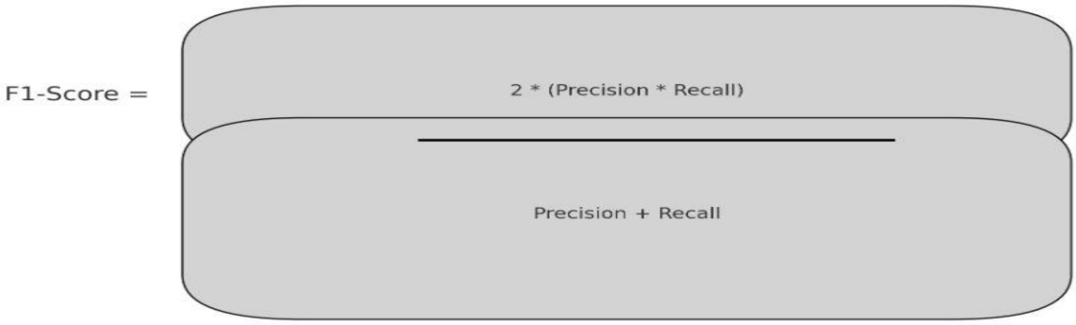
3.6.4 Recall

Recall measures the model's ability to correctly identify positive instances. For fractures (class 0), 96% of actual fractures were correctly classified, while for healthy cases (class 1), 99% were accurately identified.



3.6.5 F1-Score

The F1-score balances precision and recall, providing a useful metric for imbalanced class distributions. Both class 0 (fracture) and class 1 (healthy) have an F1-score of 0.97, indicating balanced performance for both classes.



3.7 Model Optimization

This is an essential stage in model development that involves optimizing the parameter of the model and the datasets to obtain the best possible performance. It involves various techniques and strategies aimed at improving the performance of the model. It collectively enhances the model's training, evaluation, and overall predictive performance, making it robust and efficient for classifying bone fractures in images. The various strategies adopted in this study include image data generator, principal component analysis, canny edge detection, batch normalization, early stopping, and check pointing as a hyperparameter technique.

Chapter 4: Experimentation and Analysis

4.1 Experimental Setup

The image illustrates the key components and process of an X-ray machine, including the electron beam generated by a cathode, striking a tungsten anode to produce X-rays. These X-rays pass through a filter and are directed toward the patient for diagnostic imaging. A lead case ensures radiation safety, and the result is an X-ray image showing internal structures like bones.

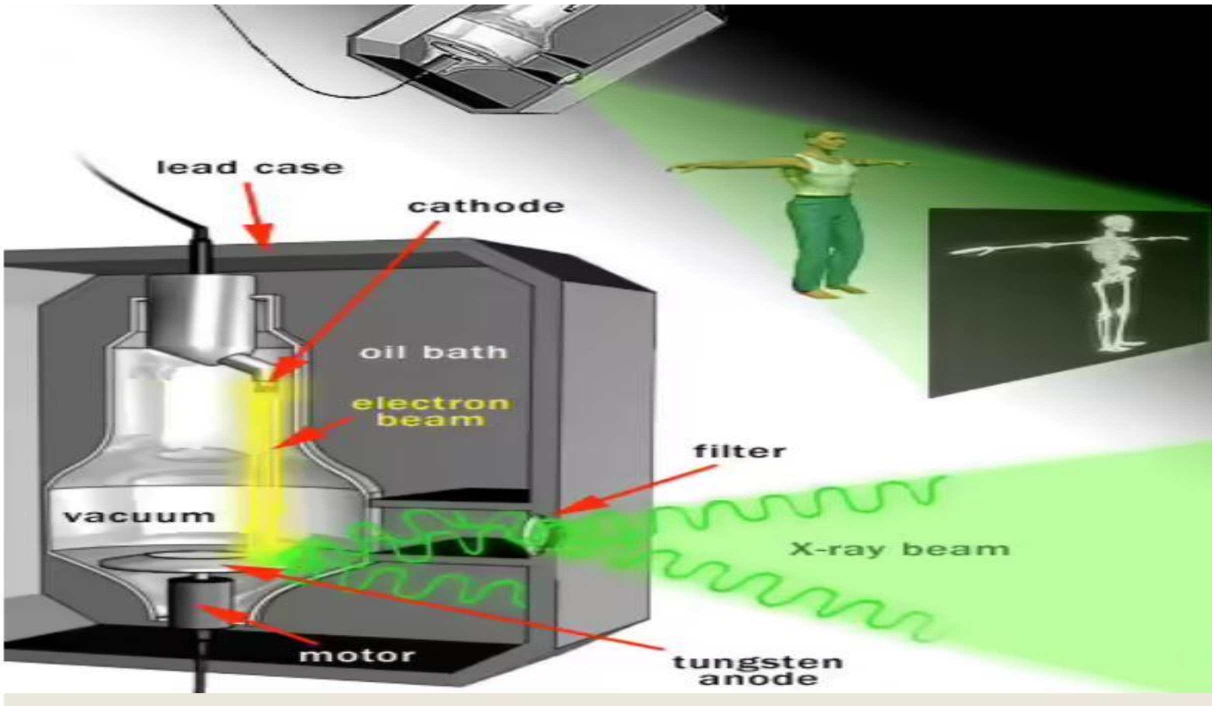


Figure 4.1: Experimental Setup for X-Ray and Imaging in Medical Diagnostics (Tom Harris 2024).

4.2 Model Training Results

This section discusses the results obtained using various models, including the Deep ConvNet model, ResNet model, VGG16-SVM model and the proposed hybrid Deep ConvNet integrated with canny edge detection. The evaluation also includes performance matrics and assessment methods, were presented in tabular and graphical formats including the confusion matrices. The best performing and worst performing models were highlighted. Here, the model accuracy and loss graph of the Deep ConvNet is shown. The confusion matrix of the best performing and the least performing before the improved models with their accuracy was also shown.

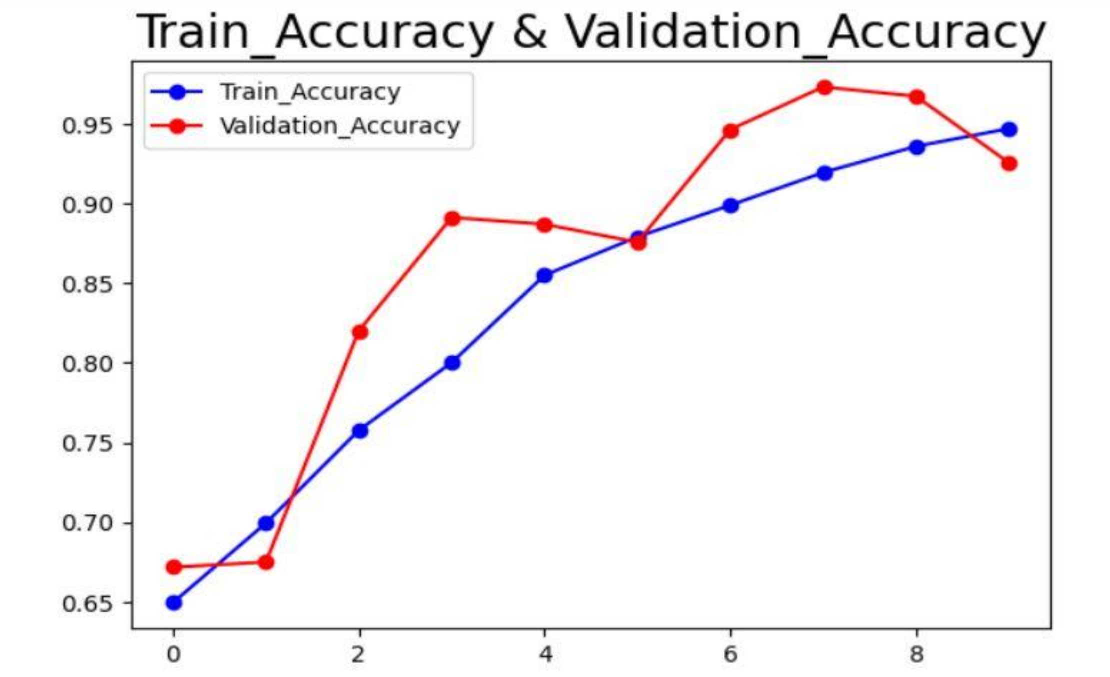


Figure 4.2: **Deep ConvNet Classifier Accuracy Curve.**

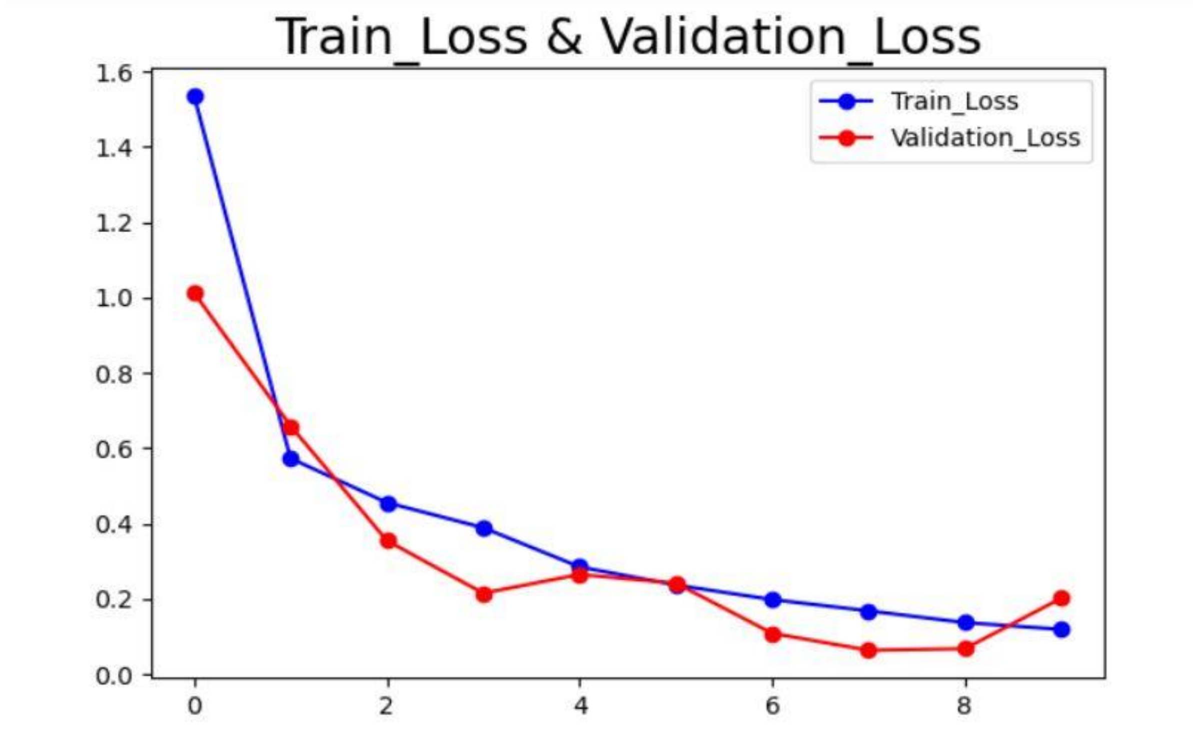


Figure 4.3: Deep ConvNet Classifier Loss Curve.

Figures (4.2 & 4.3) shows the training and validation curves representing the accuracy and loss of the Deep ConvNet classifier model developed. As clearly seen, there is a good convergence of the training dataset and the test or validation dataset. The validation loss slightly lower than the train loss indicates that the model test dataset performed well and hence the model generalization is high.

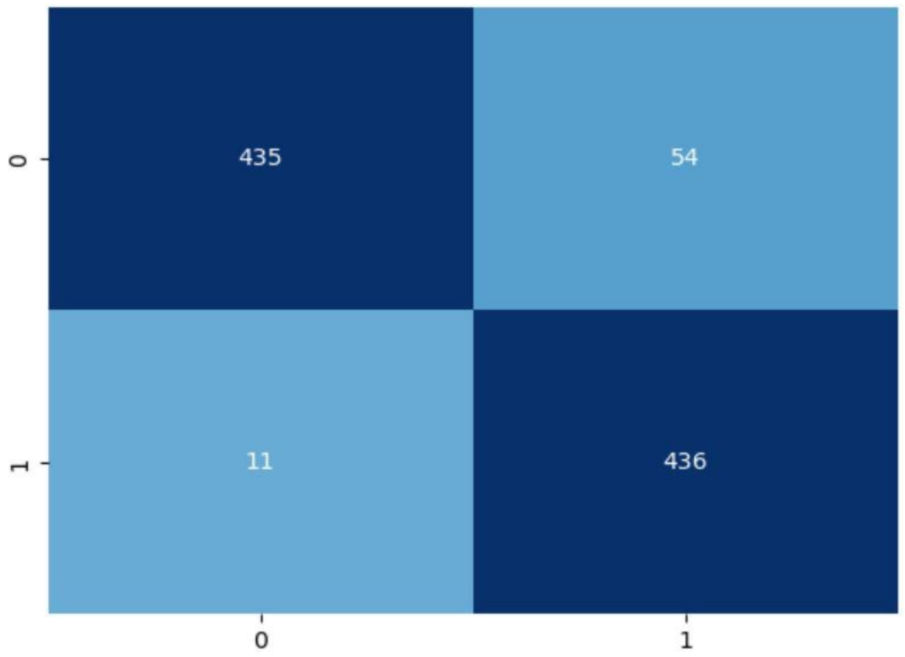


Figure 4.4: Deep ConvNet classifier confusion matrix.

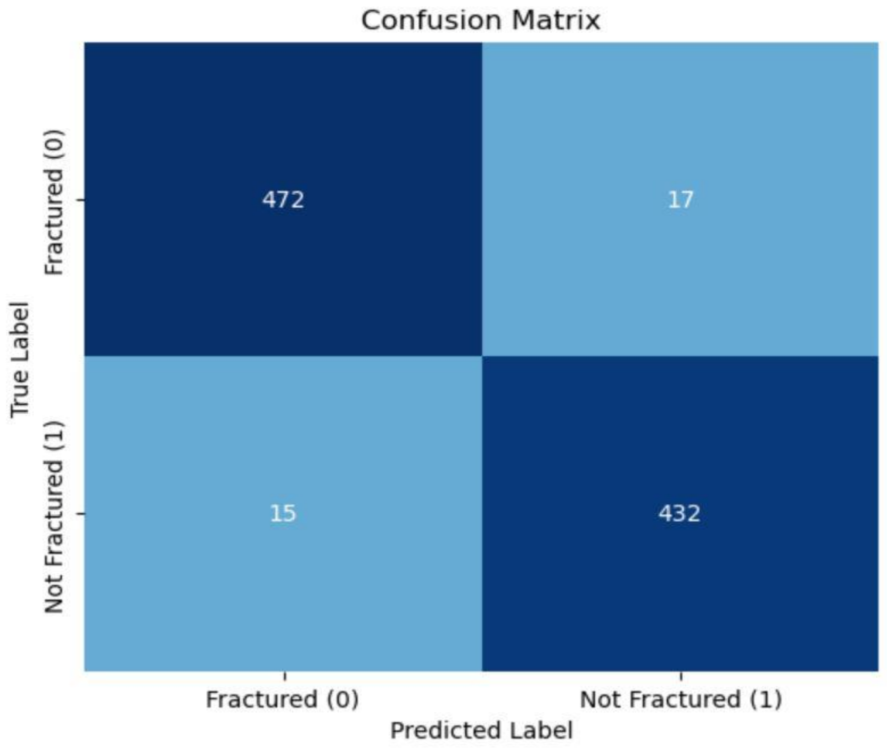


Figure 4.5: ResNet classifier confusion matrix.

Figures (4.4 & 4.5) present the binary class confusion matrix for the highest performing model and lowest-performing model before model improvement. As with every deep learning model classification problem, the objective is to increase True Positives (TP) and True Negatives (TN) while reducing False Positives (FP) and False Negatives (FN), ensuring the model accurately classifies different statuses. From the figures, the True Positives (TP) and True Negatives (TN) of the ResNet classifier is 904 while that of the Deep ConvNet classifier is 871 indicating that the ResNet classifier performed better than Deep ConvNet classifier. Further optimization tasks have been implemented to improve both True Positives and True Negatives. This led to the development of a Deep ConvNet integrated with a canny edge detection model.

4.3 Optimization

Optimization is an essential phase in developing machine learning models, as it includes key steps and processes that enhance the model's performance. Many optimization tasks were performed and an expected improvement on the performance metrics was observed. As already mentioned, the first optimization task done on the model was efficient data handling to manage memory, resizing input images for consistency, and splitting data into training, validation, and test sets for proper evaluation. These steps ensure robust, efficient preprocessing and enhance the model's overall effectiveness.

The second optimization task is model training and evaluation optimizations include batch normalization and dropout layers to enhance learning and reduce overfitting, while a well-structured CNN architecture with pre-trained models like VGG16 boosts training speed and performance. Early stopping, model checkpoint callbacks, and grid search for hyperparameter tuning further refine the training process and optimize model accuracy.

The third optimization involves using Grid Search CV for hyperparameter tuning and applying PCA to reduce the dimensionality of extracted features to 12,000, which decreases computational complexity and overfitting, thereby enhancing the performance of downstream classifiers like SVM. The last optimization is the use of canny edge detection, which enhances essential image features by focusing on edges and contours while reducing noise, helping the model learn distinguishing characteristics. This approach improves the model's robustness against variations in lighting, color, and background, ultimately leading to improved performance when fed into the Deep ConvNet, as shown below.

4.4 Improved Results

The results of all the models trained are presented here for the selected dataset. The models selected are Deep ConvNet model, ResNet model, VGG16-SVM and proposed hybrid Deep ConvNet integrated canny edge detection model. Each model was trained over 10 epochs with a batch size of 32 images expect the VGG16-SVM model. The accuracy and loss for both training and validation were calculated for each model to assess performance.

First, the results of the Deep ConvNet are presented, followed by the improved version, hybrid Deep ConvNet integrated canny edge detection, evaluated based on the accuracy and loss of the developed model. Next, the performance of the ResNet is detailed, and finally, the results of the improved hybrid Deep ConvNet integrated canny edge detection are presented using the confusion matrix.

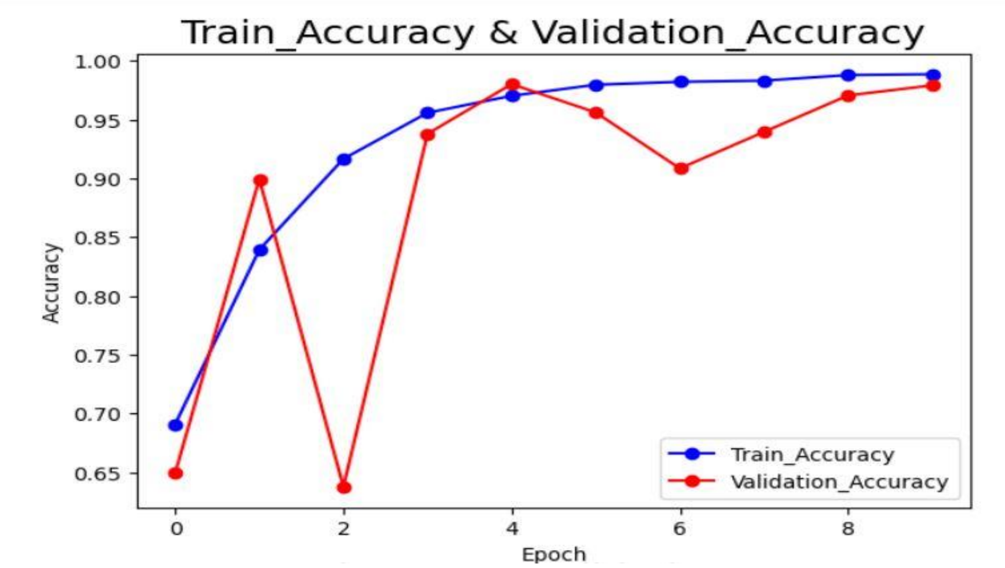


Figure 4.6: Hybrid Deep ConvNet with canny edge detection classifier Accuracy Curve.

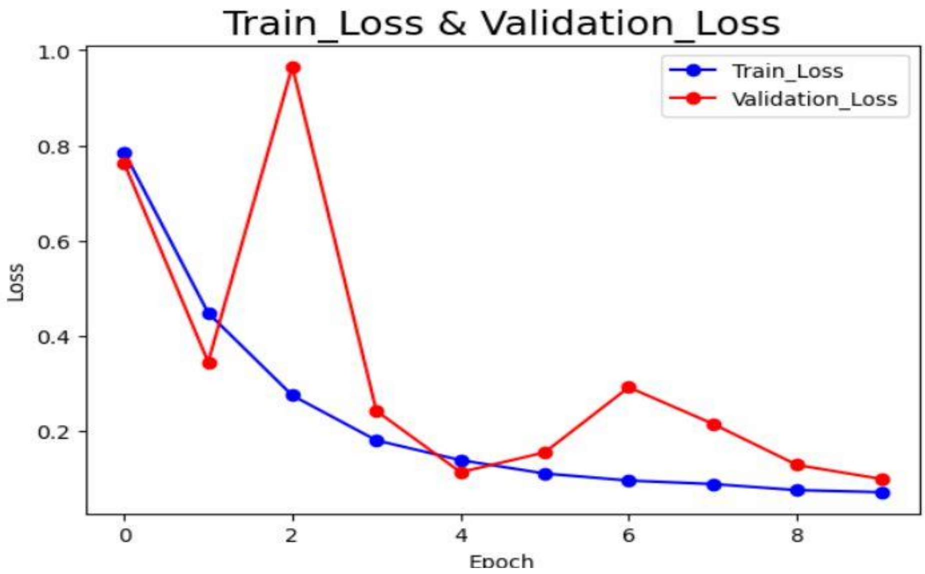


Figure 4.7: Hybrid Deep ConvNet with canny edge detection classifier Loss Curve.

Figures (4.6 & 4.7) present proposed hybrid Deep ConvNet with canny edge detection classifier the training and validation loss and accuracy curves.

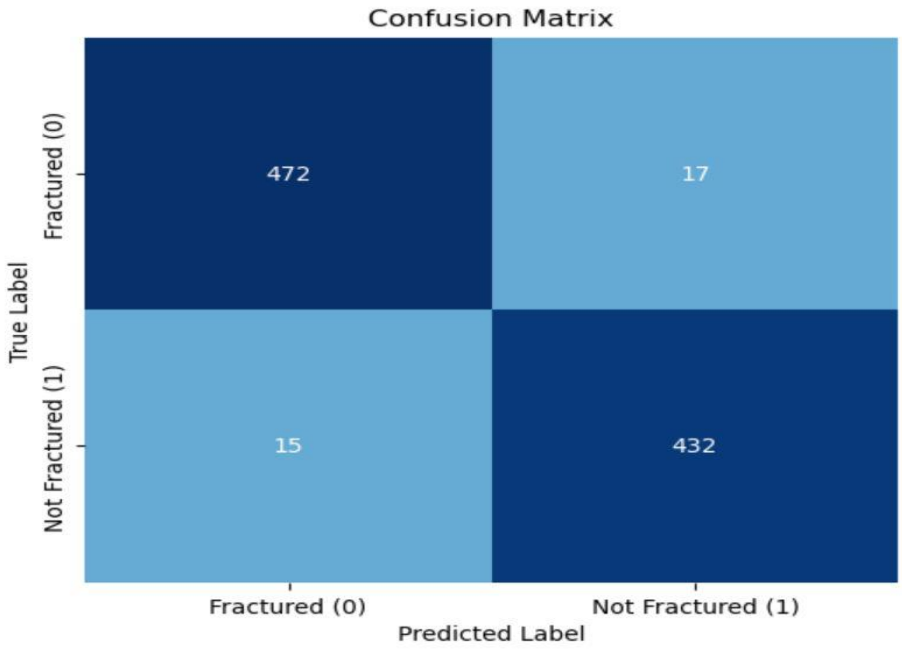


Figure 4.8: ResNet Confusion matrix

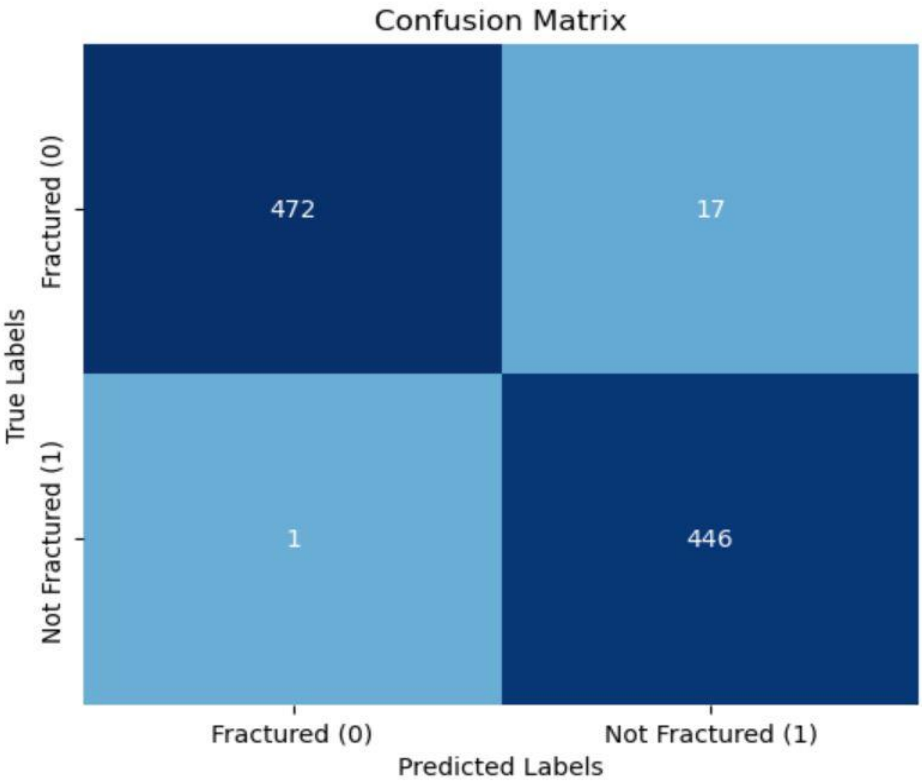


Figure 4.9: Proposed Deep ConvNet with canny edge confusion matrix.

Figures (4.8 & 4.9) show the proposed Deep ConvNet integrated with canny edge detection confusion matrix for binary class. The Deep ConvNet with canny edge detection outperforms the ResNet classifier, as shown by an increase in True Positives (TP) and True Negatives (TN) from 904 for the

ResNet Classifier to 918 for the proposed hybrid Deep ConvNet integrated canny edge detection classifier, demonstrating a clear improvement in performance.

In Figures 4.10 and 4.11, it is noticed that both the training and testing losses for Deep ConvNet are very low. Additionally, the training accuracy reaches nearly 95% after 9 epochs, while the testing accuracy begins to decline after 7 epochs, eventually reaching 97%.

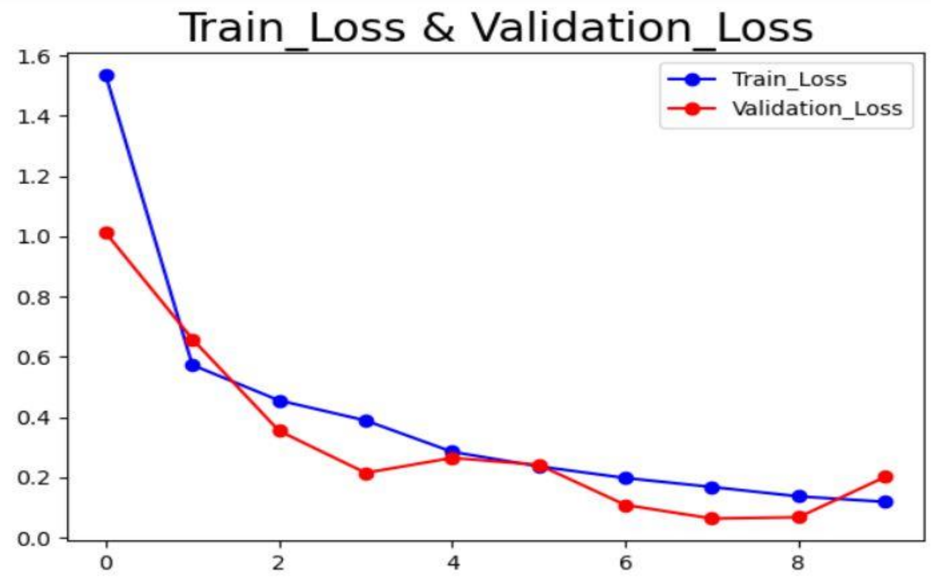


Figure 4.10: DeepConvNet Classifier Loss Curve.

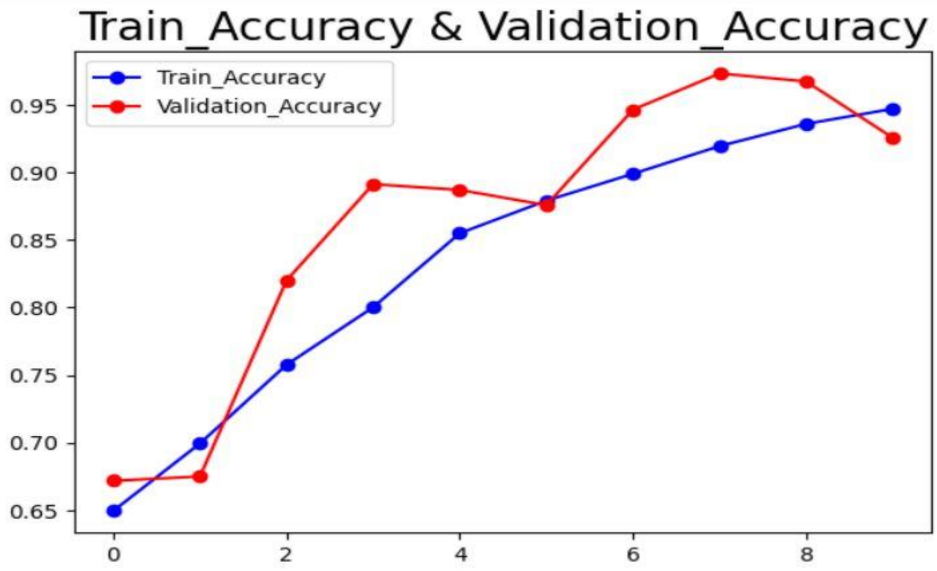


Figure 4.11: DeepConvNet Classifier Accuracy Curve.

A similar trend is seen in Figure 4.12 and 4.13 for the ResNet model. The training loss approaches zero, while the testing loss drops after 8 epochs, reaching 0.2. The training accuracy achieves 98% after 8 epochs, with the testing accuracy staying around 97% after 9 epochs.

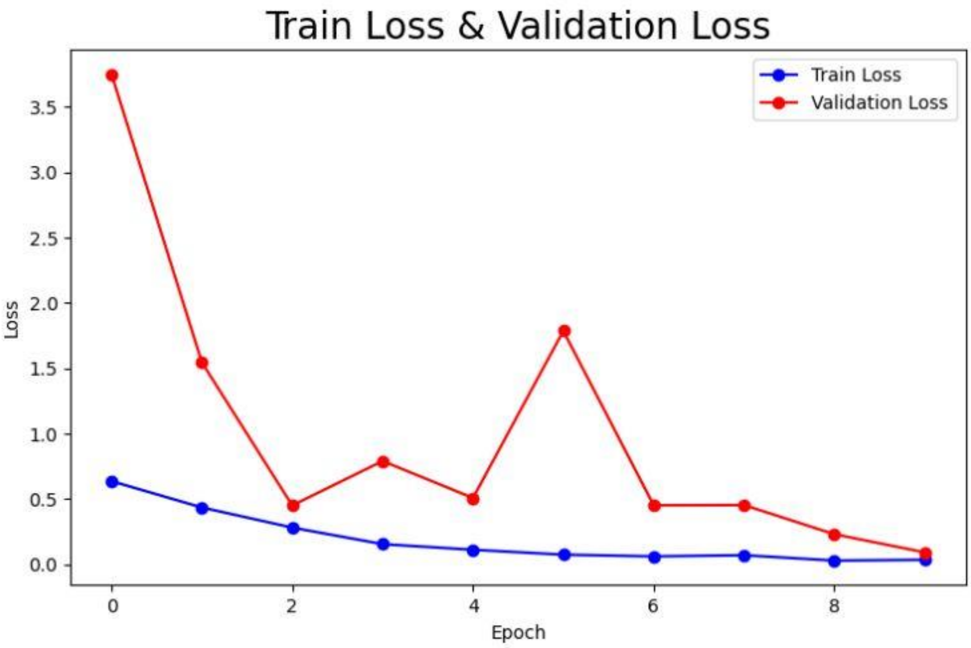


Figure 4.12: ResNet Classifier Loss Curve.

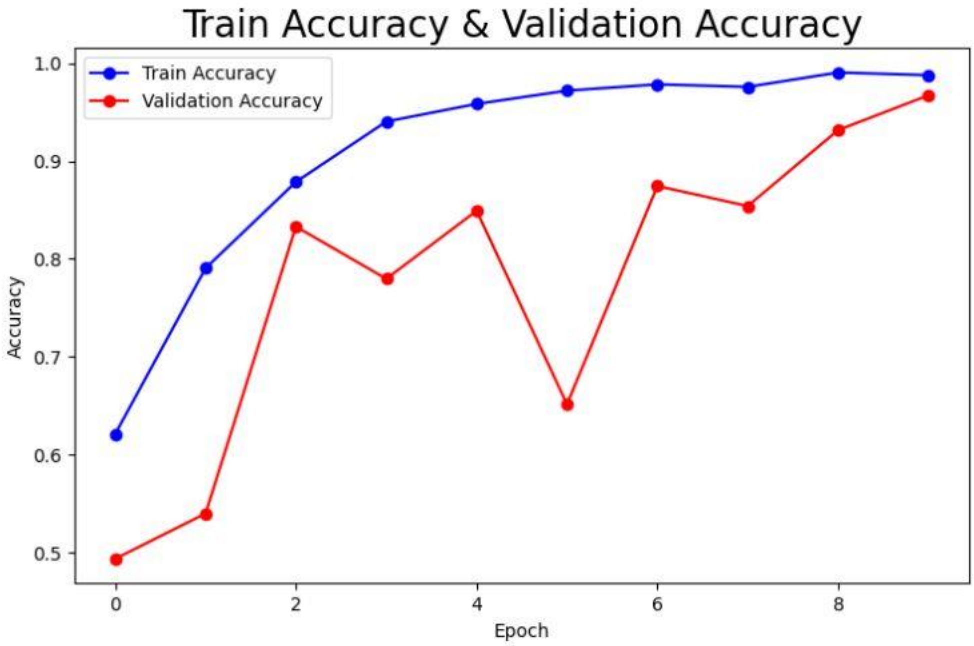


Figure 4.13: ResNet Classifier Accuracy Curve.

In figures 4.14 below, it is observed that both the training and testing accuracy for VGG16-SVM hits 83% and 81% respectively.

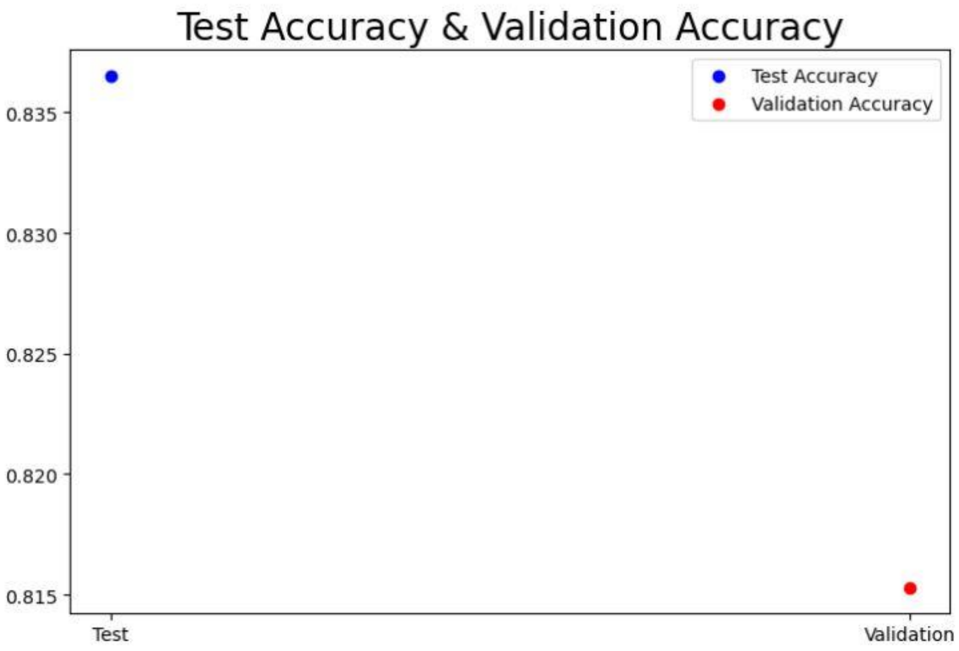


Figure 4.14: VGG16-SVM Classifier Accuracy Curve.

Finally, Figures 4.15 and 4.16 illustrate that with the proposed hybrid Deep ConvNet integrated with the canny edge detection model, both the training and testing loss nearly reached zero. Training accuracy reaches 98% after 9 epochs, but the testing accuracy slightly drops from 99% to 98% after 8 epochs.

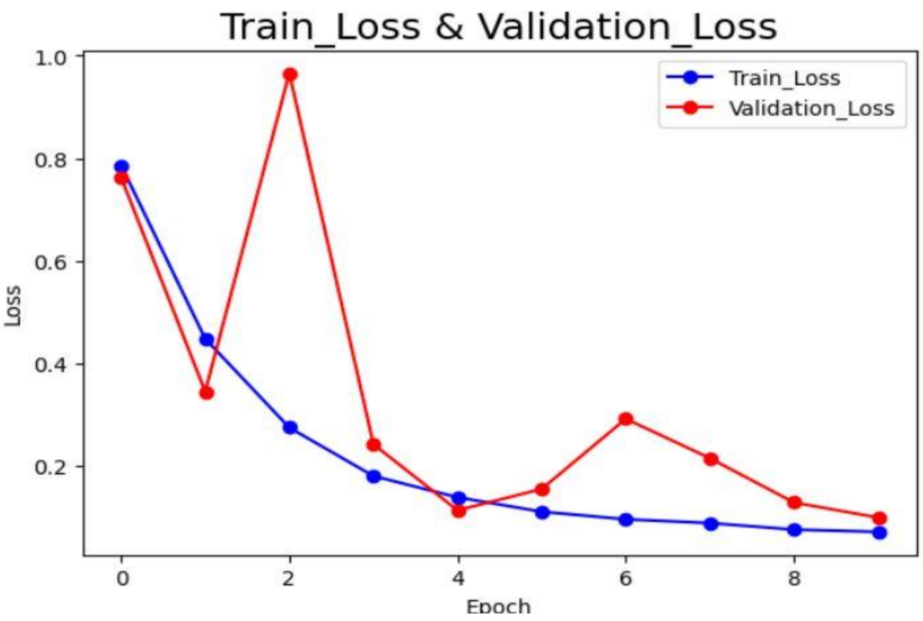


Figure 4.15: Proposed Deep ConvNet with Canny Edge Detection Loss Curve.

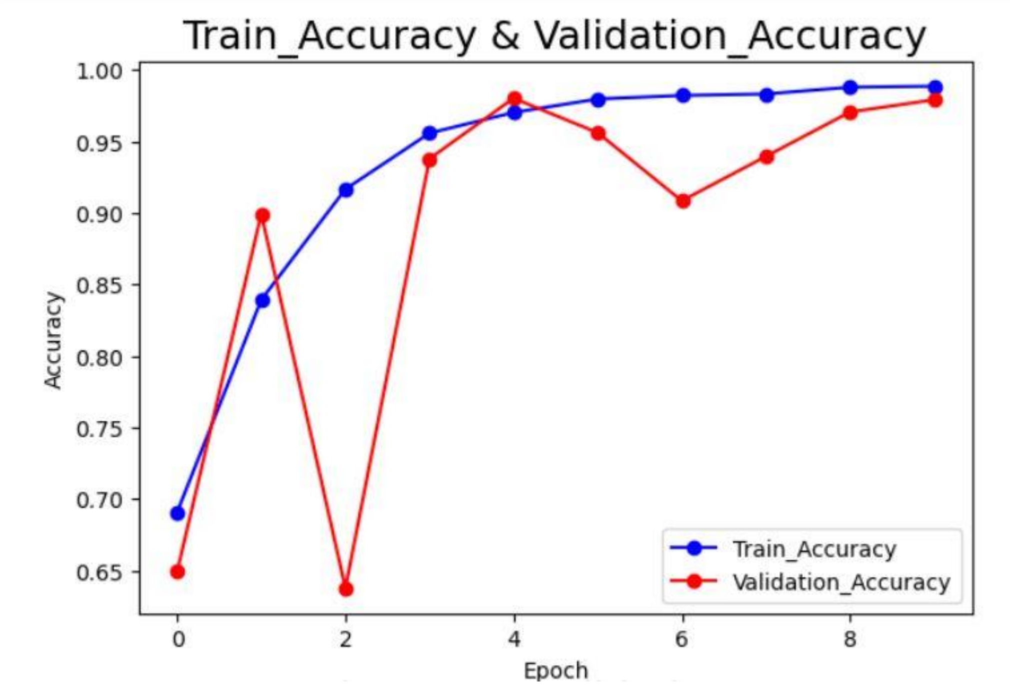


Figure 4.16: Proposed Deep ConvNet with Canny Edge Detection Accuracy Curve.

The receiver operating characteristic (ROC) for VGG16-SVM, Deep ConvNet, ResNet, and the proposed Deep ConvNet integrated canny edge detection are show in Figures 4.17, 4.18, 4.19, and 4.20 respectively.

The VGG16-SVM model demonstrates reasonably high performance with an AUC of 0.91. However, it has the lowest AUC compared to other models, indicating that it is less effective at distinguishing between fracture and non-fracture cases. While it performs well, it may struggle with more complex classifications compared to models with higher AUC values.

Deep ConvNet achieves an impressive AUC of 0.99, demonstrating excellent classification performance. The ROC curve closely approaches the upper-left corner, indicating a very high true positive rate (TPR) and a very low false positive rate (FPR). This model significantly outperforms the VGG16-SVM model and effectively distinguishes between fracture and non-fracture cases. Similarly, ResNet also demonstrates near-perfect performance, achieving an AUC of 0.9956. Its ROC curve indicates outstanding classification ability with minimal false positives.

The proposed hybrid model achieves a perfect AUC of 1.00, indicating flawless classification performance in this context. The ROC curve reaches the upper-left corner, suggesting no false positives and perfect recall. It outperforms all other models, due to the integration of canny edge detection, which enhances its ability to accurately identify fractures.

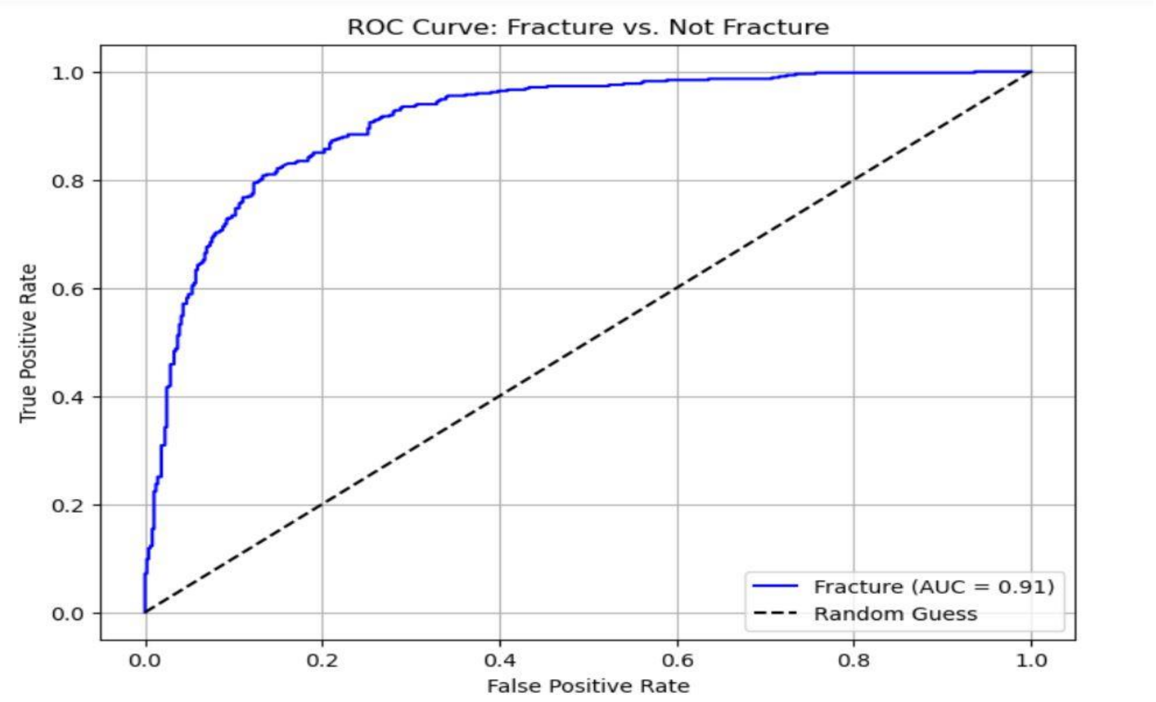
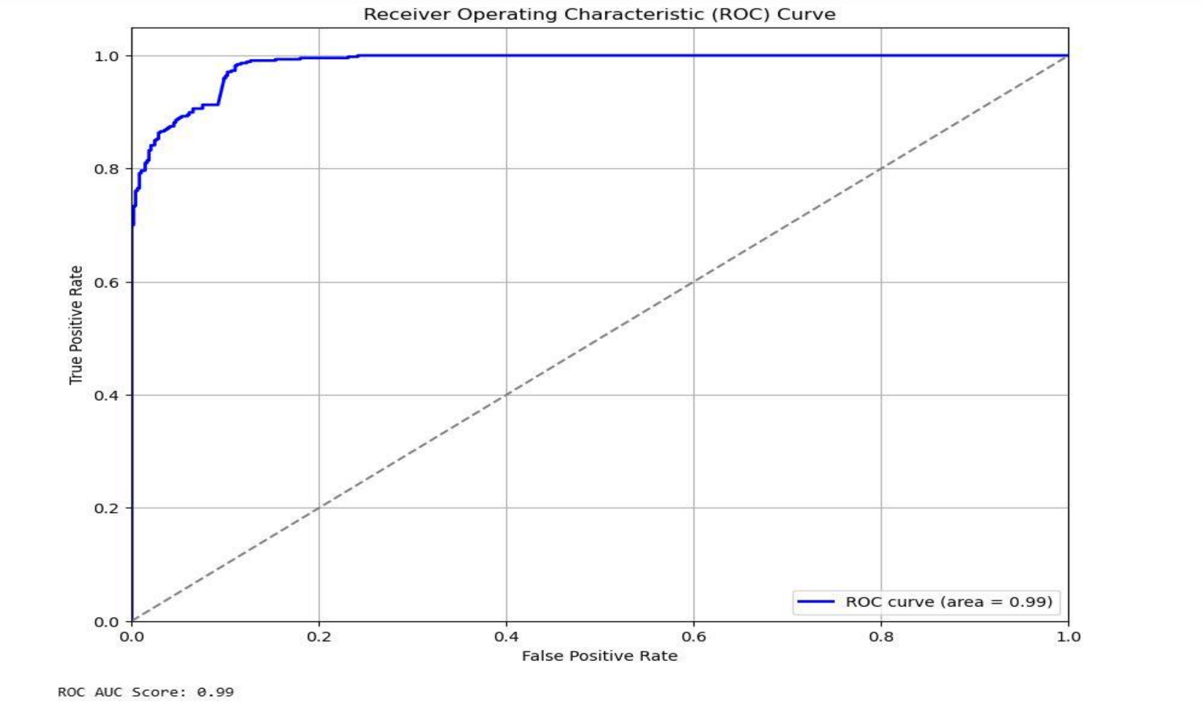


Figure 4.17: VGG16-SVM ROC Curve.



```
ROC AUC Score: 0.99
```

Figure 4.18: Deep ConvNet ROC Curve.

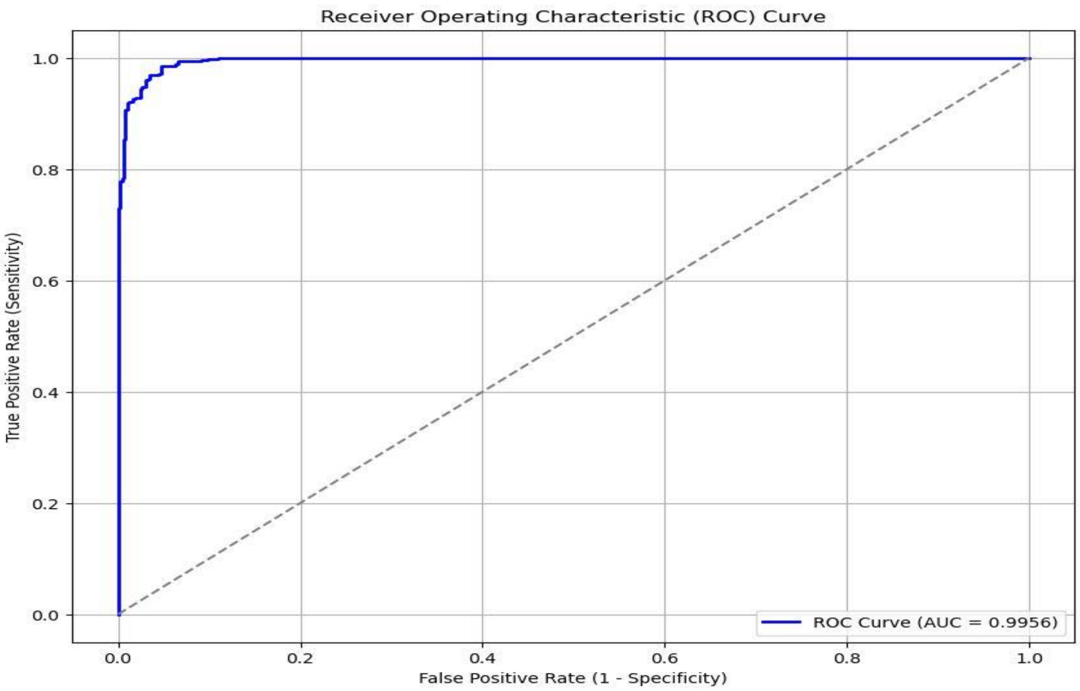
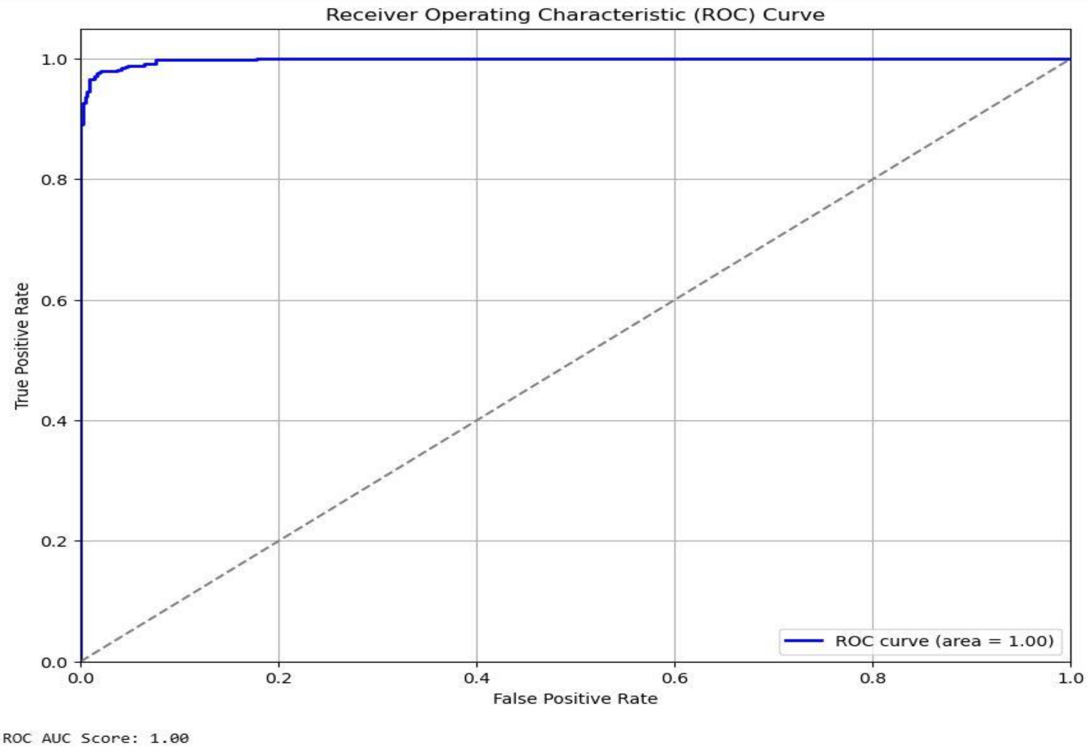


Figure 4.19: ResNet model ROC Curve.



```
ROC AUC Score: 1.00
```

Figure 4.20: Proposed hybrid Deep ConvNet with Canny Edge Detection ROC Curve.

The proposed hybrid deep ConvNet with canny edge detection is the top performer and offers a more reliable solution, followed by ResNet and deep ConvNet, while VGG16-SVM demonstrates relatively lower performance.

The confusion matrices for Deep ConvNet, ResNet, VGG16-SVM and the proposed Deep ConvNet integrated canny edge detection are displayed in Figures 4.21, 4.22, 4.23, and 4.24 respectively. The false positive count for Deep ConvNet, ResNet, VGG16-SVM and Proposed Deep ConvNet with integrated canny edge detection is 54, 17, 77 and 17, respectively. However, the false negative counts for these models are 11, 15, 76 and 1, respectively. In this work, the proposed model has the fewest false negatives, with just 1. The model's performance is evaluated using precision, recall, F1-score, and accuracy.

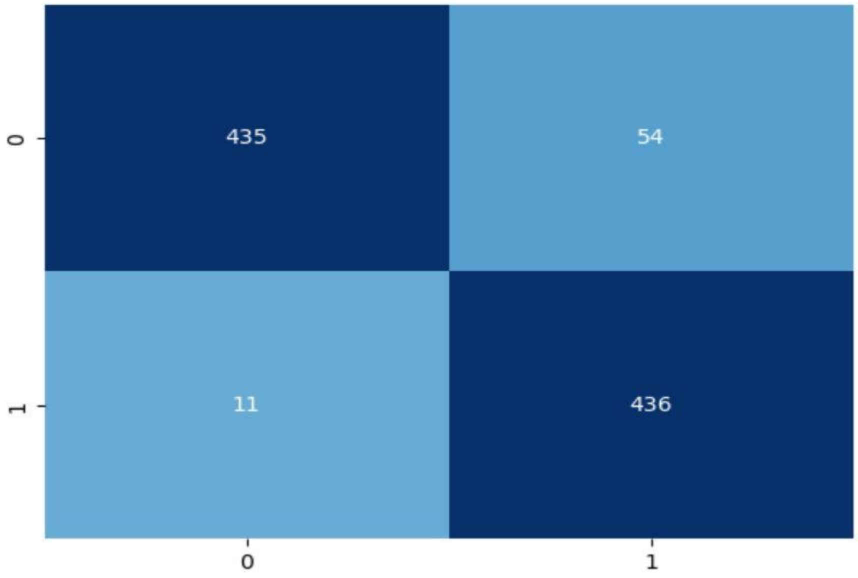


Figure 4.21: Deep ConvNet model confusion matrices.

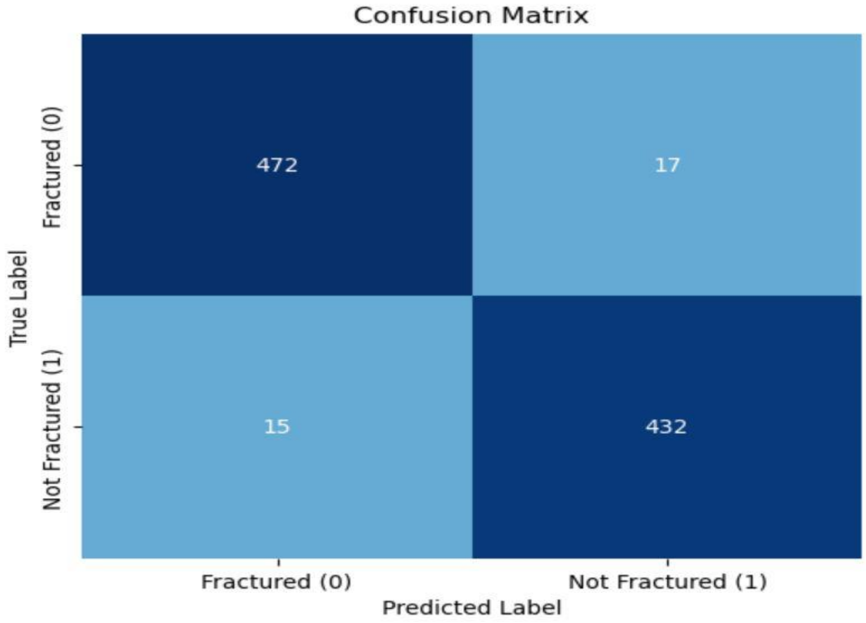


Figure 4.22: ResNet model confusion matrices.

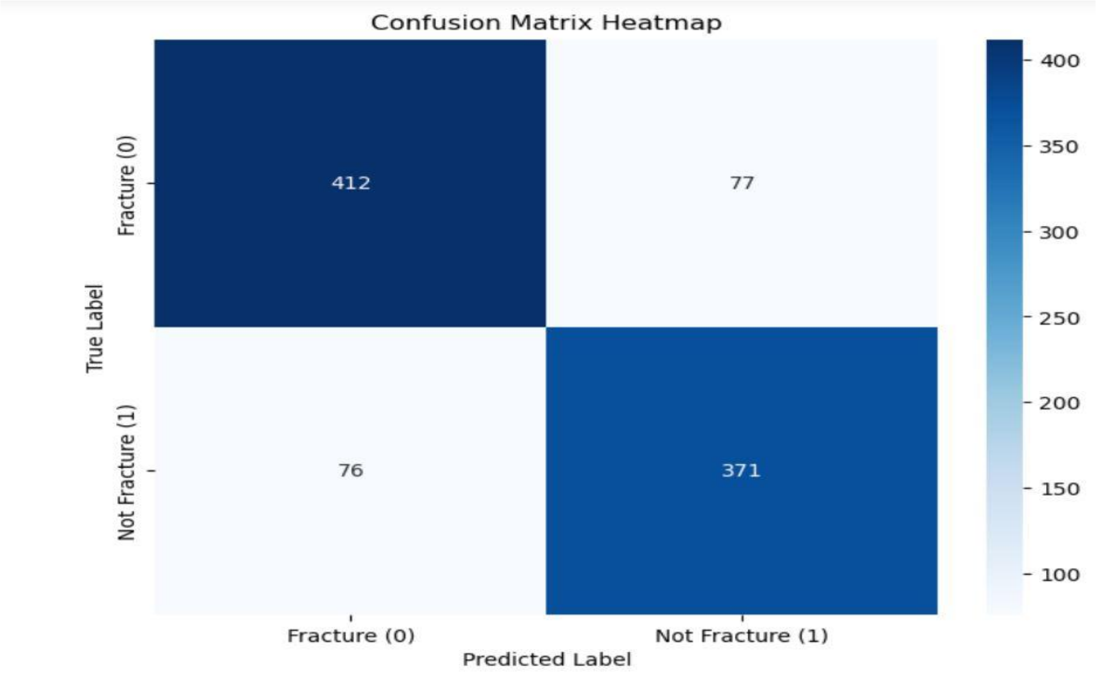


Figure 4.23: VVG16-SVM model confusion matrices.

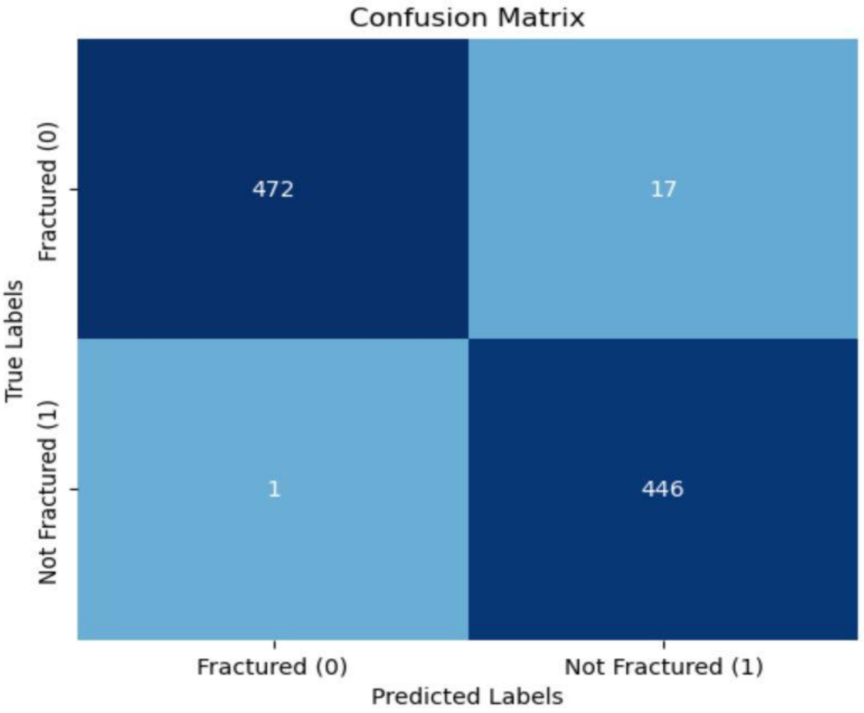


Figure 4.24: Proposed deep ConvNet with canny edge detection confusion matrices.

Above figures, presents the confusion matrices for the deep ConvNet, ResNeXt, VGG16-SVM and the proposed model, as depicted in figures 4.21, 4.22, 4.23, and 4.24, respectively.

4.5 Discussion

Table 4.1 presents the performance metrics, while the confusion matrices are displayed above. In Fig 4.21, the CM for deep ConvNet shows 435 true positives (TP), 436 true negatives (TN), 54 false positives (FP), and 11 false negatives (FN). Fig 4.22 illustrates the confusion matrix for ResNet, with 472 TP, 432 TN, 17 FP, and 15 FN. In Fig 4.23, the VGG16-SVM model displays 412 true positives (TP), 371 true negatives (TN), 77 false positives (FP), and 76 false negatives (FN). Finally, Fig 4.24 displays the confusion matrix for the proposed hybrid deep ConvNet integrated with canny edge detection, showing 472 TP, 446 TN, 17 FP, and 1 FN. Table 4.1 reveals that VGG16-SVM recorded the lowest precision at 83%, followed by deep ConvNet with 89% precision. While ResNet and the proposed hybrid Deep ConvNet integrated with canny edge detection, both achieved 96% precision for healthy bone classification. Similarly, for fractured bone, deep ConvNet, ResNet, and VGG16-SVM reached precision at 98%, 97%, and 84%, respectively, while the proposed hybrid deep ConvNet reached the highest at 100%. For recall, both ResNet and the proposed hybrid deep ConvNet integrated with canny edge detection achieved 97% for fractured bones, while Deep ConvNet reached 89%, and VGG16 had the lowest recall at 84%. Similarly, for non-fractured bones, the recommended model achieved 100%. Additionally, the Deep ConvNet integrated with canny edge detection achieved an F1 score of 98%, compared to the deep ConvNet and ResNet models, which reached approximately 93% and 97%, respectively, for both fractured and non-fractured bones. In contrast, VGG16-SVM recorded the lowest F1 scores, with 84% for fractured bones and 83% for healthy bones. These results suggest that deep ConvNet, ResNet, VGG16-SVM exhibit strong performance in bone image classification, though they require significant computation time and could benefit from further refinement. The proposed deep ConvNet integrated with canny edge detection stands out with exceptional performance and the shortest computation time.

Table 4.1: A comparative evaluation of effectiveness.

Model	Type of Bone	Precision	Recall	F1-Score	Accuracy
Deep ConvNet Model	Fracture Not Fracture	0.98 0.89	0.89 0.98	0.93 0.93	0.93
ResNet Model	Fracture Not Fracture	0.97 0.96	0.97 0.97	0.97 0.97	0.96
VGG16-SVM Model	Fracture Not Fracture	0.84 0.83	0.84 0.83	0.84 0.83	0.81
Proposed Hybrid Model with Canny Edge	Fracture Not Fracture	1.00 0.96	0.97 1.00	0.98 0.98	0.98

Furthermore, the recall for the proposed model is 100% for healthy bone and highest precision for fractured bone, at 100%, is achieved by the proposed model. Moreover, the proposed deep ConvNet integrated with canny edge detection model attains a classification accuracy of 98%, while VGG16-SVM, Deep ConvNet and ResNet, achieve accuracies of 81%, 93% and 96% respectively.

4.6 Evaluation Analysis

The evaluation analysis of the model accuracies is another vital step in machine learning and deep learning model development as it gives opportunity to access the model performance, compare their results, guide for model improvement, identify overfitting & underfitting and ensures robustness & reliability for choosing the best models that best suits the dataset. Here, the results of the models are evaluated and compared, the best performing models are selected for further optimization tasks and least performing models are dropped.

4.6.1 Comparison with other studies

This work differs from previous research by demonstrating that high accuracy was achieved in the detection and classification of bone fractures, making it suitable for real-time applications. In this research, we assessed the performance of advanced CNN models, including Deep ConvNet, ResNet, and hybrid model that integrates both ML and DL approaches. which VGG16-SVM using an openaccess dataset for evaluation. Table 4.2 provides a favourable comparison between this study and recent research in the field. An overall classification accuracy of 98% was achieved for bone fracture detection using deep ConvNet integrated with canny edge detection as the base model. This study introduces an approach to classifying and detecting bone fractures, and demonstrated, the accuracy is higher than in several recent papers.

Tables 4.2: Comparison with other studies

Study	Data source	Method/Model	Performance
This study	X-Ray	Hybrid deep ConvNet with Canny Edge Detection	98%
[30] Amani Al-Ghraibah et al, (2024)	X-Ray	Principal Component Analysis (PCA) Support Vector Machine (SVM).	80%
[33] Luo et al, (2021)	X-Ray	Deep Learning Model integrated with curriculum learning guided	86.57%
[24] Shuzhen L et al, (2022)	X-Ray	Modified Ada-ResNeSt backbone network with AC-BiFPN	68.4%

The precision performance for each model is illustrated in Figure 4.25. The proposed Deep ConvNet with canny edge achieves a maximum precision of 100% for detecting fractured bones. Deep ConvNet, ResNet and VGG16-SVM demonstrate comparable precision values of 98%, 97%, and 84% respectively. The lowest precision, at 83%, is produced by the VGG16-SVM for detecting healthy bone.

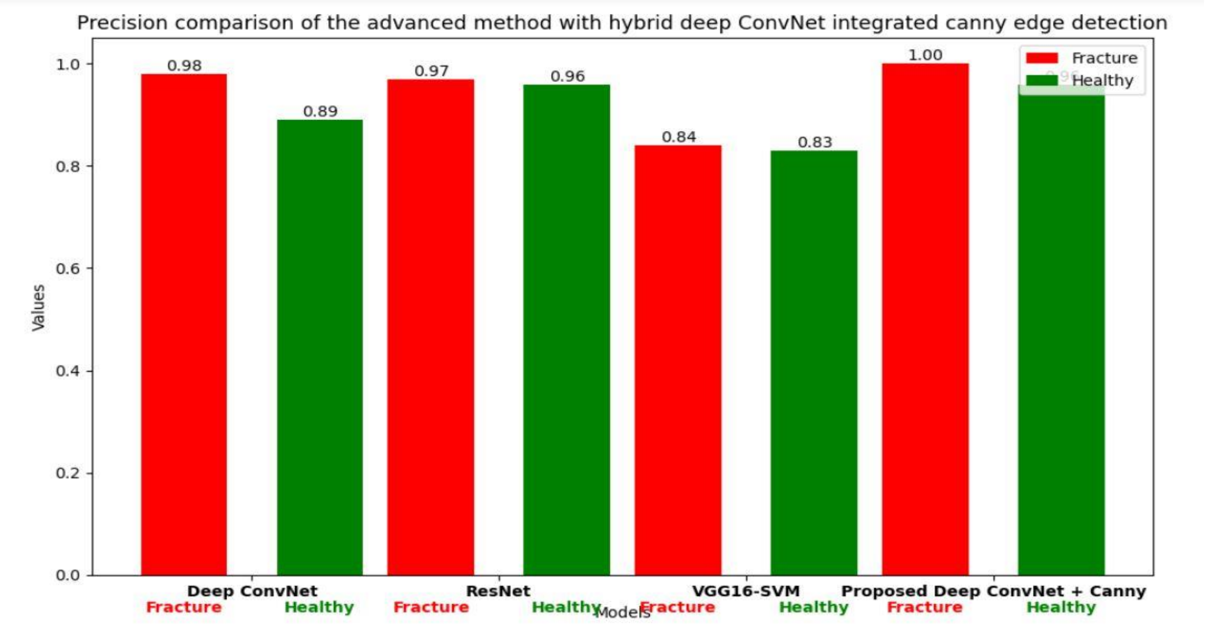


Figure 4.25: Precision Comparison of the Advanced method with Proposed Hybrid Model.

Figure 4.26 displays the recall values for each model. Both ResNet and the proposed model achieved a recall of 97% for detecting fractured bones, while VGG16-SVM and Deep ConvNet attained 84% and 89% respectively. For healthy bone detection, the proposed model achieved a recall of 100%, with the lowest model achieved recall values exceeding 83% for VGG16-SVM model.

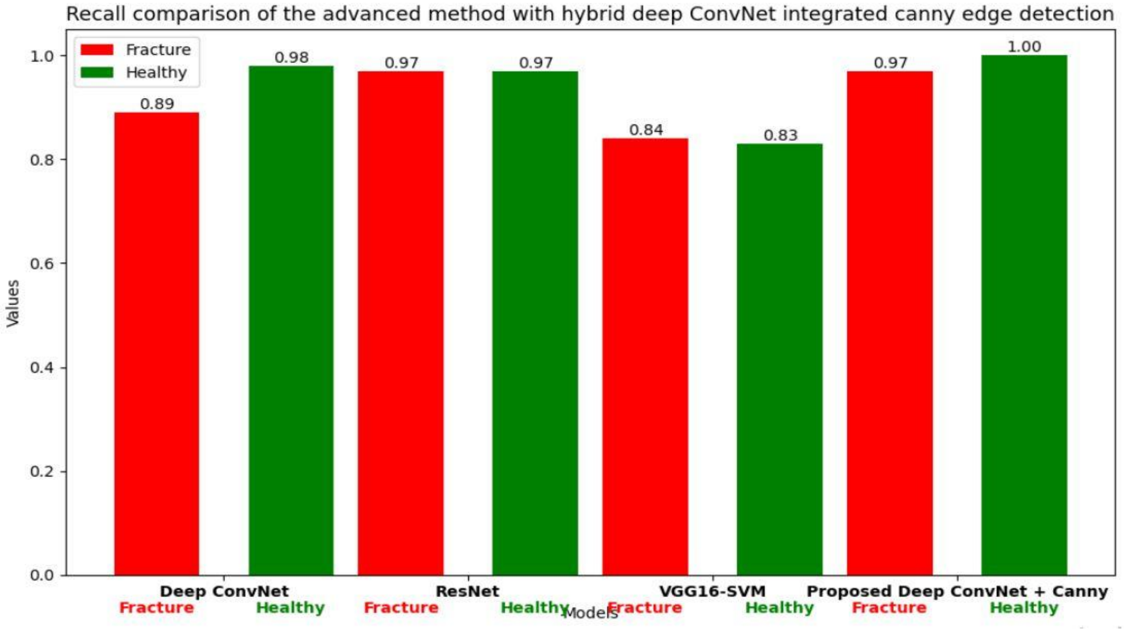


Figure 4.26: Recall Comparison of the Advanced method with Proposed Hybrid Model.

Additionally, figure 4.27 presents the F-score of all models. The proposed Deep ConeNet hits a 98% F1-score for both fractured and healthy bone. In comparison, Deep CovNet and ResNet reached a 93% and 97% F1-score respectively, while VGG16-SVM achieved 84% for fractured and 83% for healthy bone.

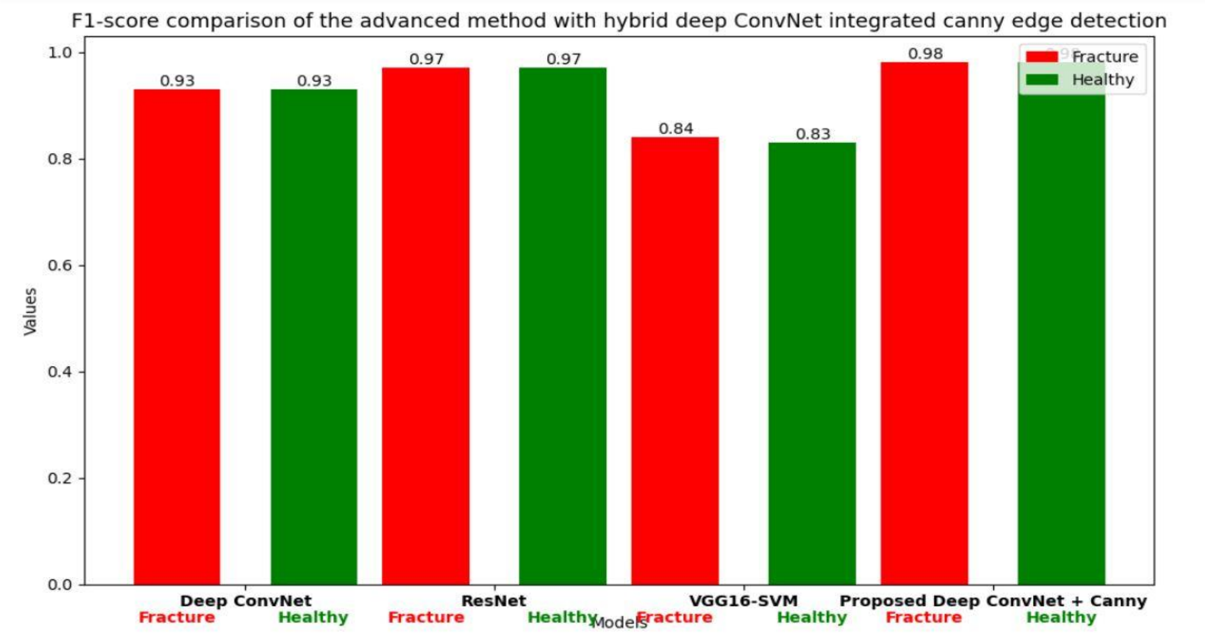


Figure 4.27: F-score Comparison of the Advanced method with Proposed Hybrid Model.

Figure 4.28 illustrates the classification accuracy of each model. The proposed Deep ConvNet with Canny edge detection achieved the highest accuracy, surpassing 98%. While VGG16-SVM, Deep ConvNet and ResNet also showed solid performance, with accuracies of 81%, 93% and 96%, respectively, though they did not match the performance of the proposed model.

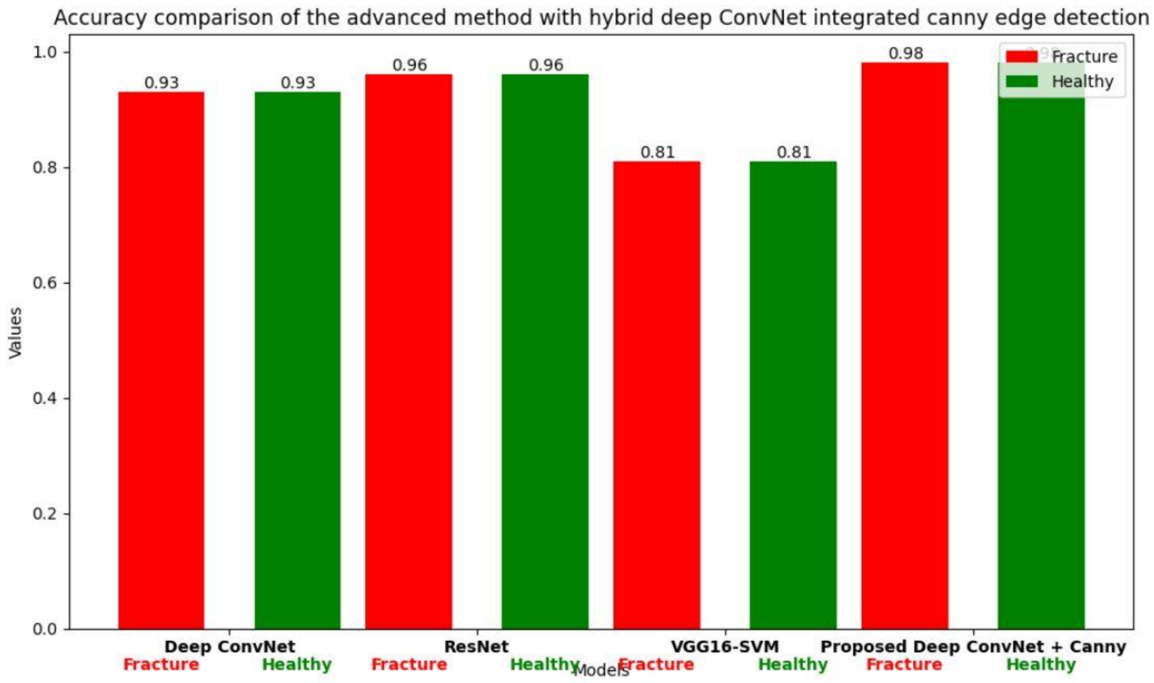


Figure 4.28: Accuracy Comparison of the Advanced method with Proposed Hybrid Model.

Figures 4.25, 4.26, 4.27, and 4.28 demonstrate that the proposed deep ConvNet with canny edge detection surpasses the performance of advanced deep ConvNet models. This shows that the proposed hybrid deep ConvNet with canny edge detection can effectively assist doctors by providing a reliable second opinion in diagnosing both fractured and not fractured bones.

Chapter 5: Conclusions and Future Steps

In this chapter, the conclusion of the research will be presented by highlighting the key research findings to address the research aims and objectives as indicated in the introduction chapter. The chapter will also summarize the values and contributions of the key research findings and additionally mention some of the limitations encountered during the research process while proposing some possible future research considerations.

5.1 Concussions and Limitations

This study aims to analyze medical imaging data and classify bone fractures in patients. The findings indicate that extracting features with a high degree of accuracy classify the condition of bone health. Many studies have explored the application of machine learning and deep learning in the medical field, particularly in bone fracture detection classification using medical imaging data.

In this study, we propose a deep ConvNet integrated with canny edge detection. The proposed model has fewer parameters compared to other and requires less computation time per epoch. The canny-processed images are used as input to the ConvNet for deep feature extraction. The extracted features are then passed to a classification module, where a sigmoid layer is used to classify images as either healthy or fractured. The proposed deep ConvNet achieves an accuracy of 98% and a precision of 100%, indicating that the use of canny images enhances performance. Moreover, zero is nearly reached for both the validation losses and training losses, confirming the model's high sensitivity.

As bone fracture detection is the main application area of this research, the developed model will contribute significantly to both researchers and healthcare practitioners. The key findings of this study will support ongoing research on the application of deep learning and artificial intelligence to the diagnosis and classification of bone fractures. In real-world applications, healthcare practitioners will find this study valuable when applied to the detection of bone fractures. This will assist in optimizing treatment plans and supporting clinical decision-making. Fracture detection not only helps in preventing severe complications, but it can also reduce healthcare costs by avoiding the need for more invasive treatments due to delayed diagnosis. This ultimately improves patient care by assisting in treatment prioritization and management strategies. Adopting this model as a baseline for continuous improvement, where more patient data is collected to train the model and feedback from performance analysis is used for model refinement, is another practical consideration. Additionally, integrating the model into existing healthcare systems as part of a routine diagnostic strategy can be beneficial for medical practitioners. Again, this approach will improve the generalization of model and encourage wider acceptance based on user’s expertise and choices.

In the findings and analysis, modern challenges associated with these innovations and approaches have been thoroughly explored. The significant progress in medical science, driven by the widespread application of machine learning, is evident from these findings. It is frequently observed that machine learning's contribution to such humanitarian efforts is invaluable in building a more robust healthcare system within the country. However, the use of machine learning has been overshadowed by deep learning's initiation for bone fracture detection, as deep learning has demonstrated remarkable accuracy and practical applications. To maximize the benefits of this technology, its integration into the healthcare sector should be supported and encouraged at a national level, ensuring it covers rural regions and inspiring new researchers to advance these kinds of initiatives.

5.2 Future Work

One area for future research could involve using datasets from real-world clinical environments, which may include interference from surrounding physiological conditions. In such cases, noise reduction techniques would need to be implemented to improve the accuracy and reliability of fracture detection models. Another area of future interest would be consideration to identify the size of the fracture area, which will assist medical professionals in more informed decisions regarding bone fractures. Also, can be improved further by expanding the classification to include different types of fractures, such as avulsion fractures, segmental fractures, spiral fractures, complete fractures, transverse fractures, greenstick fractures, and more. Another approach researchers could explore is integrating their developed model with X-ray machines. Furthermore, optimization approaches will be employed to minimize the processing complexity of the algorithms. Virtual server computing and machine learning are vital in identifying health issues, particularly for people in remote regions with restricted access to healthcare services. Machine learning driven diagnostic tools serve as supplementary aids, assisting radiologists make precise diagnoses. This is of huge benefit to medical practitioners in the effective diagnosis of bone fractures, enabling proper treatment planning and reducing delays in diagnosis, especially in remote areas, and ultimately saving costs for healthcare facilities. Research outputs like this study will help in advancing the world and making things easier for humans.

Chapter 6: Reference

1. International Osteoporosis Foundation. "Broken Bones, Broken Lives: A Roadmap to Solve the Fragility Fracture Crisis in Europe, 2018. Available online: https://www.osteoporosis.foundation/

2. Amirkolaee, H.A.; Bokov, D.O.; Sharma, H. Development of a GAN architecture based on integrating global and local information for paired and unpaired medical image translation. Expert Syst. Appl. 2022, 203, 117421.

3. T. Anu and R. Raman, Detection of bone fracture using image processing methods, Int J Comput Appl, 975 (2015), p. 8887

4. Amodeo M, Abbate V, Arpaia P, Cuocolo R, Orabona GD, Murero M, et al. Transfer Learning for an Automated Detection System of Fractures in Patients with Maxillofacial Trauma. Applied Sciences. 2021;11(14):6293. Available from: https://www.mdpi.com/1180840.

5. Varoquaux G, Cheplygina V. Machine learning for medical imaging: methodological failures and recommendations for the future. NPJ Digital Medicine. 2022;5(1):1–8. Available from: https://www.nature.com/articles/s41746-022-00592-y.pdf

6. Bengio Y, Lecun Y, Hinton G. Deep learning for AI. Communications of the ACM. 2021;64(7):58–65. Available from: https://doi.org/10.1145/3448250.

7. Yadav DP, Rathor S. Bone Fracture Detection and Classification using Deep Learning Approach. 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC). 2020;p. 282–285. Available from: https://doi.org/10.1109/PARC49193.2020.236611.

8. Zhang X, Wang Y, Cheng CT, Lu L, Harrison AP, Xiao J, et al. A new window loss function for bone fracture detection and localization in X-ray images with point-based annotation. 2020. Available from: https://doi.org/10.48550/arXiv.2012.04066.

9. Hardalaç F, Uysal F, Peker O, Çiçeklidağ M, Tolunay T, Tokgöz N, et al. Fracture Detection in Wrist X-ray Images Using Deep Learning-Based Object Detection Models. Sensors. 2022;22(3):1285–1285. Available from: https://doi.org/10.3390/s22031285.

10. Lapeña JF, David JN, Pauig ANA, Maglaya JG, Donato EM, Roasa F, et al. Management of Isolated Mandibular Body Fractures in Adults. Philippine Journal of Otolaryngology Head and Neck Surgery. 2021; p. 1–43. Available from: https://doi.org/10.32412/pjohns.vi.1857.

11. Ma Y, Luo Y. Bone fracture detection through the two-stage system of Crack-Sensitive Convolutional Neural Network. Informatics in Medicine Unlocked. 2021;22:100452. Available from: https://doi.org/10.1016/j.imu.2020.100452.

12. Singh, L.K.; Garg, H.; Khanna, M. Deep learning system applicability for rapid glaucoma prediction from fundus images across various data sets. Evol. Syst. 2022,1–30.

13. Randy Cahya et al, Improving the Classification of Unexposed Potsherd Cavities by Means of Preprocessing Wihandika, ; Lee, Yoonji; Data, Mahendra; Aritsugi, Masayoshi; Obata, Hiroki Preview author details.

14. Lakkimsetti M; Devella, Swati G; Patel, Keval B; Sarvani, Dhandibhotla; Kaur J ; etal. Cureus; Palo Alto Optimizing the Clinical Direction of Artificial Intelligence with Health Policy: A Narrative Review of the Literature Vol. 16, Iss. 4, (2024). DOI:10.7759/cureus.58400

15. Knud N, NairzIngrid B, Barbieri. Enhancing patient value efficiently: Medical history interviews create patient satisfaction and contribute to an improved quality of radiologic examinations September 2018PLOS ONE 13(9):e0203807 DOI: 10.1371/journal.

16. Ali T, Emre G, Ibraheem S. A Survey on IoT Smart Healthcare: EmergingTechnologies, Applications, Challenges, and FutureTrendsM.

17. Leonardo Tanzi, Enrico Vezzetti, Rodrigo Moreno and Sandro Moos X-Ray Department of Management and Production Engineering, Politecnico di Torino, 10129 Torino, Italy Bone Fracture Classification Using Deep Learning: A Baseline for Designing a Reliable Approach Appl. Sci. 2020, 10(4), 1507; https://doi.org/10.3390/app10041507

18. Shuzhen Lu, Shengsheng Wang & Guangyao Wang Automated universal fractures detection in X-ray images based on deep learning approach. Published: 04 June 2022.

19. Yang A.Y.,&Cheng L.(2019),Long-Bone Fracture Detection Using Artificial Neural Networks Based on Contour Features of X-Ray Images, arXiv: 1902.07897v1, 21

20. Y. Cao, H. Wang, M. Moradi, P. Prasanna, T.F. Syeda-Mahmood Fracture detection in x-ray images through stacked random forests feature fusion Proc. - int. symp. biomed. imaging, IEEE (2015), pp. 801-805, 10.1109/ISBI.2015.7163993

21. D.H. Kim, T. MacKinnon Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks Clin Radiol, 73 (2018), pp. 439-445, 10.1016/j.crad.2017.11.015

22. Thomas Davenport and Ravi Kalakota The potential for artificial intelligence in healthcare, https://www.sciencedirect.com/science/article/pii/S2514664524010592?via%3Dihub (2019)

23. Dhirendra Prasad Yadav, Sandeep Rathor. Bone Fracture Detection and Classification using Deep Learning Approach.February 2020 DOI: 10.1109/PARC49193.2020.236611

24. Shuzhen L, WangGuangyao W. Automated universal fractures detection in X-ray images based on deep learning approach June 2022Multimedia Tools and Applications 81(1) DOI: 10.1007/s11042-022-13287-z

25. Chijioke O, Maiss R. Application of Artificial Intelligence and Machine Learning in Diagnosing Scaphoid Fractures: A Systematic Review October 2023Cureus 15(10) DOI: 10.7759/cureus.47732 LicenseCC BY 3.0

26. Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, ―Hidden TwoStream Convolutional Networks for Action Recognition. ‖ arXiv, Oct. 30, 2018. Accessed: Dec. 20, 2023. [Online]. Available: http://arxiv.org/abs/1704.00389

27. Dustakar Surendra Rao, L. Koteswara Rao, Vipparthi Bhagyaraju, P. Rohini. Advancing Human Action Recognition and Medical Image Segmentation using GRU Networks with V-Net Architecture. Vol. 15, No. 2, 2024.

28. Y. D. Pranata, K.-C. Wang, J.-C. Wang, I. Idram, J.-Y. Lai, J.-W. Liu, and I.-H. Hsieh, Deep learning and surf for automated classification and detection of calcaneus fractures in ct images, Computer methods and programs in biomedicine, 171 (2019), pp. 27–37

29. Chao-Lung YangChao-Lung Yang. A hybrid approach of simultaneous segmentation and classification for medical image analysis May 2024 Multimedia Tools and Applications DOI: 10.1007/s11042-024-19310-9

30. MachineAmani A, Mohammad A, Waseem A, Muneera A. Classification of Long Bone X-ray Images using New features and Support Vector.

31. Vishnu V, Prakash J, Rengasamy Swathika, Sharmila T et al (2015). Detection and classification of long bone fractures. International Journal of Applied Engineering Research 10:18315–18320

32. M. Varan, J. Azimjonov, and B. Maçal," Enhancing Prostate Cancer Classification by Leveraging Key Radiomics Features and Using the Fine-Tuned Linear SVM Algorithm," IEEE Access, vol. 11, pp. 88025–88039, 2023. [Online]. Available:https://doi.org/10.1109/access.2023.3306515

33. Luo, J.; Kitamura, G.; Doganay, E.; Arefan, D.; Wu, S. Medical knowledge-guided deep curriculum learning for elbow fracture diagnosis from X-ray images.

34. Ankur Mani Tripathi, Abhishek Upadhyay, Akanksha Singh Rajput, Ajay Pal Singh and Brajesh Kumar. "Automatic Detection of fracture in femur bones using Image Processing". Department of Computer Science & IT, M.J.P Rohilkhand University, Bareilly, India

35. Vedika BenganiVedika Bengani. Hybrid Learning Systems: Integrating Traditional Machine Learning with Deep learning Techniques. May 2024 DOI: 10.13140/RG.2.2.10461.22244/1

36. G. Kitamura, C.Y. Chung, and B.E. Moore. Ankle fracture detection utilizing a convolutional neural network ensemble implemented with a small sample, de novo training, and multiview incorporation, J. Digit. Imaging, vol. 32, pp. 672-677, 2019.

37. Sai Charan M, Detection of Hand Bone Fractures in X-ray Images using Hybrid YOLO NAS. January 2024IEEE Access PP(99):1-1PP(99):1-1 DOI:10.1109/ACCESS.2024.3379760 LicenseCC BY 4.0

38. L. Tanzi et al., "Hierarchical fracture classification of proximal femur X-ray images using a multistage deep learning approach.

39. Mohamed et al, "Explainable TransferLearning-Based Deep Learning Model for Pelvis Fracture Detection", In-ternational Journal of Intelligent Systems, vol. 2023

40. Beyaz, S.; Açıcı, K.; Sümer, E. Femoral neck fracture detection in X-ray images using deep learning and genetic algorithm approaches. Jt. Dis. Relat. Surg. 2020, 31, 175.

41. Jones, R.M.; Sharma, A.; Hotchkiss, R.; Sperling, J.W.; Hamburger, J.; Ledig, C.; Lindsey, R.V. Assessment of a deep-learning system for fracture detection in musculoskeletal radiographs. NPJ Digit. Med. 2020, 3, 1–6.

42. Y. Ma and Y. Luo, “Bone fracture detection through the two-stage system of crack-sensitive convolutional neural network,” Informatics in Medicine Unlocked

43. G. Kitamura, C.Y. Chung, and B.E. Moore, "Ankle fracture detection utilizing a convolutional neural network ensemble implemented with a small sample, de novo training, and multiview incorporation," J. Digit. Imaging, vol. 32, pp. 672-677, 2019

44. Dupuis, M.; Delbos, L.; Veil, R.; Adamsbaum, C. External validation of a commercially available deep learning algorithm for fracture detection in children.

45. Shahab S B, Atefeh Yarahmadi, Chung-Chian Hsu ,Meghdad Biyari,Mehdi Sookhak,Rasoul Ameri,Iman Dehzangi, Anthony Theodore Chronopoulos,Huey-Wen Liang. Application of explainable artificial intelligence in medical health.

46. Guang Yang,Qinghao Ye, Jun Xia. Unbox the black-box for the medical explainable AI via multi-modal and multi-centre data fusion.

47. Laura Marie Fayad, CT Scan Versus MRI Versus X-Ray. Link:https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/ct-vs-mri-vs-xray.

48. William H. McIlhagga, The Canny Edge Detector Revisited. DOI: 10.1007/s11263-010-0392-0.

49. Khaled Alomar, Halil Ibrahim Aysel,and Xiaohao Cai. Data Augmentation in Classification and Segmentation. doi: 10.3390/jimaging9020046.

50. Tom Harris, https://science.howstuffworks.com/x-ray.htm2024.