Deep Learning Approaches to Cervical Spine Fracture Detection: A Comparative Study of Mobile-Net, Res-Net, and ConvNeXt-Tiny

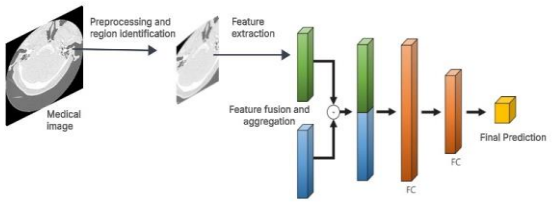
Sagar Joshi[1], Sanika Bhalerao[2], Manali Sali[2], Nikita Asawale[2]
[1] *Department of Electronics and Telecommunication, Nutan Maharashtra Institute of Engineering and Technology, Talegaon Dabhadhe, Pune, 410507, Maharashtra, India, Corresponding author*
[2] *Department of Electronics and Telecommunication, Nutan Maharashtra Institute of Engineering and Technology, Talegaon Dabhadhe, Pune, 410507, Maharashtra, India*

***Abstract*— Cervical spine fractures represent significant injuries that necessitate timely and precise diagnosis to avert serious neurological impairments. Recent progress in deep learning technologies has enabled the creation of automated systems designed for the analysis of medical images. This study conducted a comparative analysis of three prominent convolutional neural network architectures—Mobile-net, Res-net, and ConvNeXt—in detecting cervical spine fractures from computed tomography (CT) images. Our findings reveal that Res-Net and Mobile-net attained an accuracy of 97.94% and 97.50%, respectively, while ConvNeXt achieved an accuracy of 84%. These results underscore the superior performance of Res-Net and Mobile-net in this specific diagnostic task.**

***Index Terms*— Cervical Spine Fracture Detection, Deep Learning, Convolutional Neural Networks, Medical Imaging, Model Comparison, Automated Diagnosis, Performance Evaluation**

I. INTRODUCTION

Cervical spine fractures present considerable dangers, including the potential for irreversible paralysis or death if not diagnosed and treated in a timely manner. Dependence on radiologists for the interpretation of CT scans can be inefficient and susceptible to human error. Deep learning techniques, especially convolutional neural networks (CNNs), represent a promising avenue for enhancing diagnostic accuracy and minimizing processing times. These models excel in the realm of image classification, rendering them highly effective for the automated identification of fractures.. The issues of misdiagnosis and delayed detection are significant, particularly in emergency contexts where swift decision-making is essential. Automating the detection process can assist radiologists in prioritizing cases that are high-risk, alleviating their workload, and ultimately improving patient outcomes. The capacity of deep learning models to extract intricate features from medical images allows them to surpass traditional computer-aided diagnostic methods. Furthermore, pre-trained models that are fine-tuned on specialized datasets can significantly boost predictive performance. This research investigates and contrasts the efficacy of three CNN architectures—MobileNet, ResNet, and ConvNeXt—in the detection of cervical spine fractures from CT images. Through a comprehensive analysis of their performance, this study seeks to determine the most effective model for automated fracture detection, thereby providing critical insights for enhancing clinical reliability and diagnostic accuracy.



[Figure 1] CT Image processing pipeline

II. LITERATURE REVIEW

The advancement of deep learning methodologies, particularly convolutional neural networks (CNNs), has markedly improved the detection of cervical spine fractures. Numerous studies have illustrated the effectiveness of these approaches in enhancing diagnostic precision and minimizing processing durations. Nejad et al. (2023) presented an advanced system for detecting cervical spine fractures that leverages deep learning techniques, highlighting the significant role of CNNs in the realm of medical imaging [1][9]. Further validation of CNNs was

provided by Bansal et al. (2021) and a related study published in the American Journal of Neuroradiology (2021), which confirmed their high diagnostic accuracy in identifying CT cervical spine fractures [2][4]. In addition, Small et al. (2024) assessed the diagnostic performance of an artificial intelligence algorithm for cervical spine fracture detection in comparison to radiologists, emphasizing the potential of AI to enhance clinical decision-making processes [3]. Various neural network architectures, such as MobileNet and ResNet, have been investigated, demonstrating exceptional capabilities in image classification tasks [5][6]. The integration of deep learning with cloud computing has also been explored, revealing its scalability and robustness in systems designed for cervical spine fracture detection [7][19]. Comparative analyses, including those conducted by ResearchGate (2023) and GitHub (2022), have shed light on the effectiveness of diverse data representations and model architectures, underscoring the necessity of selecting the most suitable model for particular applications [8][11]. The creation of specialized models, such as NeckFrac by the University of California, Berkeley (2022), along with a comprehensive survey of deep learning techniques published in the International Journal of Advanced Research (2023), has enriched the understanding of model optimization and practical implementation [12][13]. Furthermore, supporting research on the anatomy and physiology of the cervical spine, as well as the incidence and prevalence of vertebral fractures, has provided crucial context for the application of deep learning in this field.

III. MATH

A. Commonly used equations in CNN models

Convolutional Neural Networks (CNNs) are employed in image-processing tasks because of their ability to extract both spatial and hierarchical features. The preprocessing phase, which includes feature extraction, optimization, and evaluation, constitutes the workflow of a CNN. Distinct mathematical methods and techniques are necessary at each step for the network to function. A crucial phase in image pre-processing is the normalization step, which transforms the raw pixel values into a standardized range. Normalization for the input images involves scaling the pixel values to a range of [0,1] by employing the following method:

$$y_{normalized} = \frac{y}{255} \qquad (1)$$

Here y are the raw pixel intensity values range from 0 to 255, and to promote faster convergence and numerical stability, the input data is kept on a fixed scale throughout the training process.

CNNs employ a fundamental convolutional method in their initial feature extraction process to detect local characteristics such as textures, edges, and shapes. The following equation illustrates the convolution process at work

$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n).K(m,n) \quad (2)$$

I is the input image and K represents the kernel. While (m, n) iterates over the kernel dimensions. The indices (i, j) define the output feature map's spatial coordinates.

Following convolution, non-linear activation functions are utilized to enhance the network's expressiveness. This explanation concerns the Rectified Linear Unit (ReLU), a widely used activation function in Convolutional Neural Networks (CNNs)

$$a(x) = max\ (0, x) \qquad (3)$$

The output of the activation function is denoted by a(x), with x being the input. The ReLU activation function maintains constant positive values and resets negative values to zero, thereby introducing non-linearity into the network. It enables CNNs to capture intricate connections within the data.

Consistent direction during training is facilitated by momentum, which prevents the model from becoming stuck in trivial variations. The model's effectiveness is enhanced by recalculating based on past gradients in addition to the current one. The formula that defines momentum is given by:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \qquad (4)$$

In Equation, the exponentially weighted moving average of gradients is denoted by m_t, with β_1 representing the decay rate, and g_t representing the current gradient.

The model's weights are updated according to the following rule:

$$\emptyset_t = \emptyset_{t-1} - \frac{\delta}{v_t} m_t \qquad (5)$$

The updated parameter in the equation, $\emptyset_t$ is influenced by the learning rate η, while the step size is adjusted adaptively by v_t and m_t.

The CNN's performance is typically done using metrics that are customized to particular tasks.
Accuracy refers to the ratio of correctly classified samples to the overall number of samples evaluated..

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Precision refers to the ratio of accurate predictions to the total number of instances classified as positive. It serves as a measure of the reliability of predictions identified as positive.

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

Recall: The proportion of actual positives that are accurately recognized.

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

F1 Score: This metric represents the harmonic mean of precision and recall, effectively balancing the two measures.

$$F1\ Score = 2x\frac{PrecisionxRecall}{Precision+Recall} \quad (9)$$

False positive Rate: Proportion of actual negatives incorrectly identified as positive. Indicates the rate of false positives.

$$FPR = \frac{FP}{TN+FP} \quad (10)$$

B. Equations used in MobileNetV2

Mobile-Net achieves enhanced computational efficiency through the use of depthwise separable convolutions without compromising model accuracy. In the final classification layer, the extracted feature map X from the last convolutional block undergoes a global average pooling (GAP) operation, represented as $f_{global\_avg}(X)$, which reduces the spatial dimensions by averaging the activations across each feature channel. This pooled output is then passed through a fully connected layer with learned weight parameters W and bias terms b, enabling the network to transform high-dimensional features into class scores. A softmax activation function is ultimately employed to generate probability distributions across the potential outcomes.

$$\hat{y} = softmax(W.f_{global\_avg}(X) + b) \quad (11)$$

C. Equations used in ResNet50

In ResNet-50, the last two layers are pivotal in transforming extracted features into relevant class probabilities. The penultimate layer is a fully connected (FC) layer, which accepts the high-dimensional feature vector generated by the preceding convolutional layer and converts it into class-specific logits via a linear transformation. This transformation is given by

$$Z = W_{fc}.F + b_{fc} \quad (12)$$

where W_{fc} represents the weight matrix, F is the input feature vector, and b_{fc} is the bias term. This operation helps map the learned features to a lower-dimensional space corresponding to the number of output classes. The resulting logits, Z, serve as raw, unnormalized scores that indicate the confidence of the network in predicting each class. The concluding layer employs the softmax activation function to convert the raw logits into probability distributions.The softmax function is defined as

$$P(y_i) = \frac{e^{Z_i}}{\sum_j e^{Z_j}} \quad (13)$$

where Z_i represents the logit for a specific class i, and the denominator ensures that the sum of all output values equals 1. By exponentiating the logits and normalizing them, softmax ensures that the highest logit corresponds to the most probable class while the others are proportionally scaled. This phase is essential in multi-class classification, as it allows the model to generate reliable predictions by identifying the class that exhibits the highest probability. These two layers collectively empower ResNet-50 to translate unprocessed image characteristics into understandable class likelihoods, thereby facilitating precise classification operations.

D. Equations used in ConvNeXt Tiny

The raw pixel intensity values range from 0 to 255, and to promote faster convergence and numerical stability, the input data is kept on a fixed scale throughout the training process. The ConvNeXt-Tiny model features a fully connected Dense layer, situated immediately before the output layer, comprising 256 neurons and utilizing ReLU activation. This layer reduces the high-dimensional feature maps produced by the ConvNeXt backbone to a lower-dimensional feature representation. The equation governing this layer is

$$h = max\,(0, W'_x + b') \qquad (14)$$

The input features from the Global Average Pooling2D layer are represented by x, while the learnable weight and bias parameters are denoted as W' and b′ respectively, with ReLU activation preventing negative values by setting them to zero. This layer is pivotal in detecting intricate patterns and resolving the vanishing gradient issue.

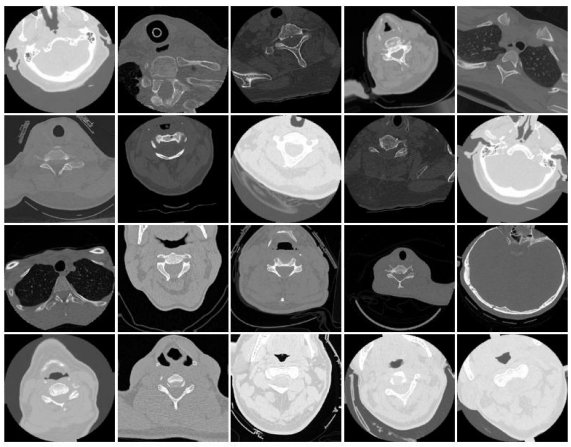
The output layer is comprised of a single Dense neuron utilizing Sigmoid activation, which generates a probability score for binary classification, specifically distinguishing between fracture and normal instances. The equation for this layer is expressed as

$$y = \sigma(W_x + b) \qquad (15)$$

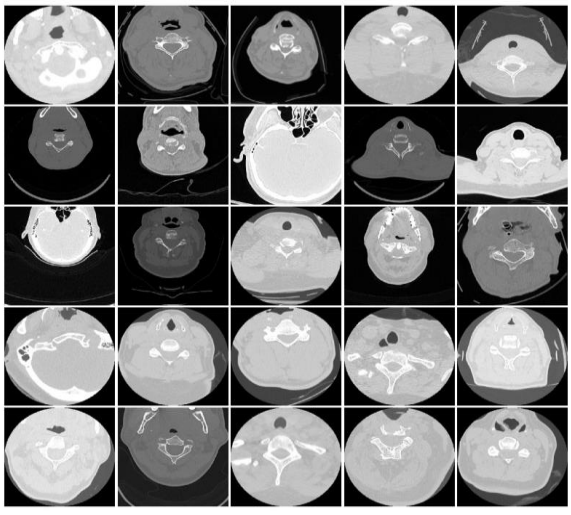
with W and b being trainable parameters, and σ denoting the Sigmoid function, which maps the output to a probability value between 0 and 1. The given probability signifies the chance of a cervical spine fracture existing. The layered structure enables the model to efficiently utilize the extracted hierarchical features and retain its ability to make dependable decisions.

IV. DATASET

The dataset employed in this research originates from Kaggle and comprises CT scans presented in PNG format, specifically assembled for the purpose of classifying cervical spine fractures. It includes a total of 4,200 images, which are divided into a training set[Figure 2] of 3,800 images and a validation set[Figure 3] of 400 images. Each of these subsets is categorized into two separate classes: fracture and normal. The fracture class contains CT images that illustrate cervical spine fractures, while the normal class consists of images depicting healthy cervical spines. The dataset is meticulously organized into designated train/ and val/ directories for the respective training and validation images. To improve the performance of the model, preprocessing methods such as histogram equalization and resizing were applied to standardize the images, thereby addressing variations in resolution and contrast. This dataset, compiled from a selection of images from the RSNA cervical spine fracture detection challenge, aims to develop classifiers that can accurately identify fractures based on spine x-rays. Slices identified as containing fractures were marked as positive, while those without fractures were designated as negative, ensuring a well-structured dataset for both training and validation. The incorporation of well-balanced classes along with diverse imaging conditions strengthens the dataset's robustness, making it appropriate for training deep learning models.. This extensive dataset serves as a strong foundation for assessing the performance and generalization capabilities of the proposed CNN architectures in practical applications



[Figure 2] Train Directory Sample Images: The CT images were designed for training and assessing the performance of deep learning models in automating the detection of cervical spine fractures from medical imaging



[Figure 3]Validation Directory sample images: CT scan images in the validation directory were classified as either fracture or normal and were used to evaluate a model's accuracy.

V. METHODOLOGY

A. Data Preprocessing

The initial stage of data pre-processing involved loading grayscale CT images of the cervical spine

from directories that were labeled as "train" and "val," with each of these directories further subdivided into sub-directories for cases involving "fracture" and "normal" conditions. All images were resized to a consistent 224x224 pixel resolution to guarantee uniform input for the deep learning models. The pre-processing pipeline's critical initial step involved normalizing data (Equation 1), whereby pixel values were adjusted to a particular range in order to enhance both model convergence and performance. To boost the training dataset and increase the model's adaptability, data augmentation methods were implemented. Potential edge cases and anomalies within the data set were identified and rectified to guarantee the quality and reliability of the data (Equation 2)(Equation 4).

B. Feature extraction and Model architecture

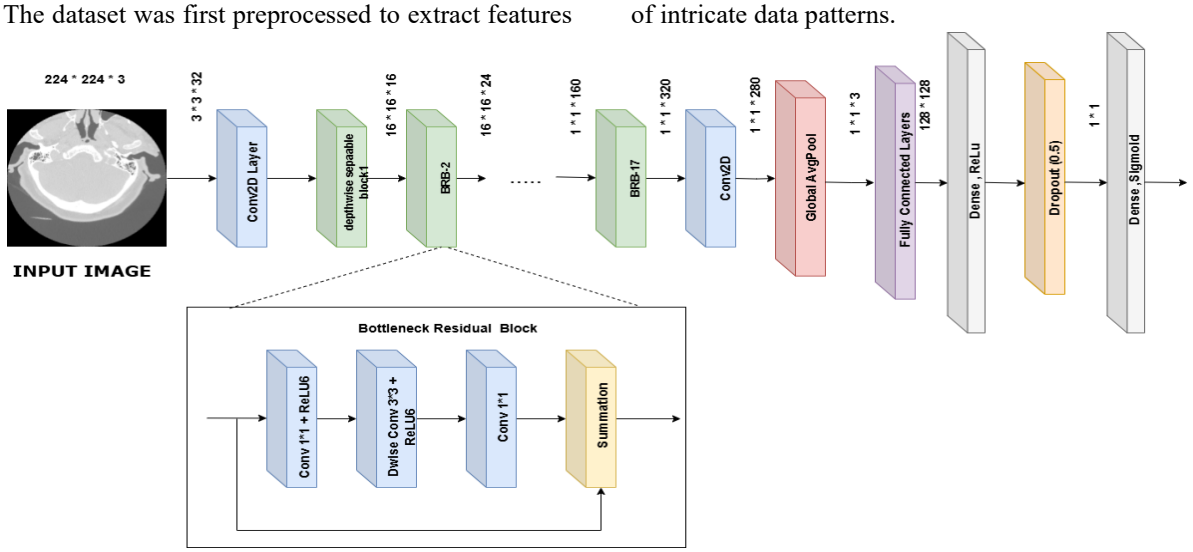
MobileNetV2 is developed for resource-efficient mobile and embedded vision applications, employing depthwise separable convolutions to decrease computational complexity and model size. The architecture [Figure 2] MobileNetV2 Architecture Diagram is centered on the inverted residual structure, which features thin bottleneck layers functioning as input and output, with lightweight depthwise convolutions applied in the intermediate layers. MobileNetV2 boasts a design that not only reduces memory usage but also preserves high performance, making it a flexible option for environments with limited resources.

The dataset was first preprocessed to extract features by resizing the images to a constant size for consistency. Pixel values were subsequently standardized to a range of (0, 1), thereby improving the efficacy of the training process. To enhance the dataset's diversity, several data augmentation methods were implemented, including rotation, flipping, and scaling. These steps were instrumental in developing a robust model that can effectively manage variations in real-world data.

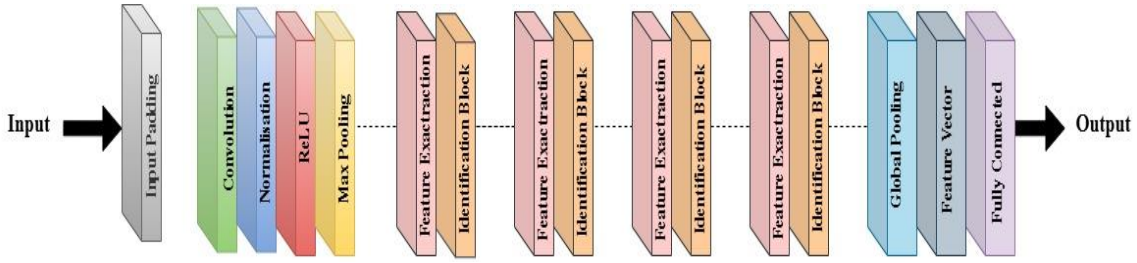
ResNet50 [Figure 3] is a distinguished deep convolutional neural network characterized by its 50-layer architecture and its innovative residual learning framework. The proposed architecture includes bypass pathways that omit one or more layers, enabling gradients to be transmitted directly through these connections. This approach helps resolve the vanishing gradient issue, thereby allowing for the training of very deep networks.

Model performance is evaluated through a range of metrics, as specified by the equation. The model's efficient convergence was facilitated by a scheduler that dynamically adjusted the learning rate, thereby enhancing training performance.

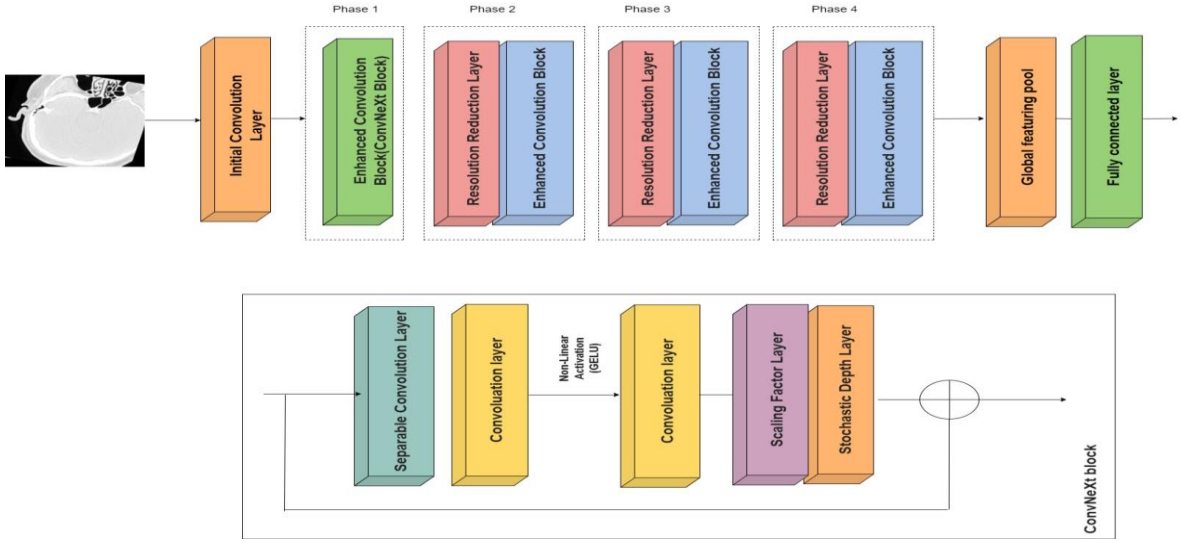
ConvNeXt Tiny [Figure 4] represents an innovative architectural model that combines the strengths of convolutional networks with those of transformers. It boasts a compact design with reduced complexity, yet still delivers high-level performance. ConvNeXt Tiny utilizes contemporary design principles including layer normalization and multi-head self-attention mechanisms, thereby facilitating efficient processing of intricate data patterns.



[Figure 2] MobileNetV2 Architecture Diagram



[Figure 3]ResNet50 Architecture Diagram



[Figure 4]ConvNeXt Tiny Architecture Diagram

C. Training and evaluation

To address the ongoing difficulties associated with the detection of cervical spine fractures, we implemented a thorough training and evaluation methodology. Although the primary assessment was centered on precision and the confusion matrix. The training duration for the MobileNetV2, ResNet50, and ConvNeXt Tiny architectures ranged from 10 to 20 epochs, utilizing early stopping methods to reduce the likelihood of overfitting. This early stopping mechanism monitored the validation loss and halted training when no significant improvement was detected over a specified number of epochs, thereby enhancing the models' ability to generalize. The evaluation metrics included accuracy and the confusion matrix, which provided a detailed analysis of the models' performance. The confusion matrix presented the tallies of true positives, true negatives, false positives, and false negatives, thereby clarifying the characteristics and prevalence of errors.. This detailed examination facilitated the identification of specific areas needing enhancement, such as minimizing false positive rates and rectifying dataset imbalances. Alongside accuracy, we evaluated additional metrics including precision, recall, and the F1 score to achieve a more thorough insight into the performance of the model. Precision measures the proportion of true positives among all predicted positives, whereas recall assesses the proportion of true positives relative to the total number of actual positives. Precision measures the proportion of true positives among all predicted positives, whereas recall assesses the proportion of true positives relative to the total number of actual positives. Reliability and robustness were critical components of the evaluation process, ensuring that the models maintained consistent performance across diverse conditions and datasets. By integrating these metrics, we developed a robustness evaluation framework aimed at assessing the effectiveness of classification, ultimately enhancing model performance, interpretability, and clinical reliability for improved outcomes in fracture detection. This thorough assessment approach ensures that the models attain not only elevated levels of accuracy but also demonstrate practical relevance and reliability in real-world situations.

VI. RESULT

Upon evaluating the models using our dataset, MobileNetV2 was identified as the leading model, attaining a remarkable accuracy of 97.50%. ResNet50 was closely followed, recording an accuracy of 97.45%. Although ConvNeXt Tiny ranked lower, it

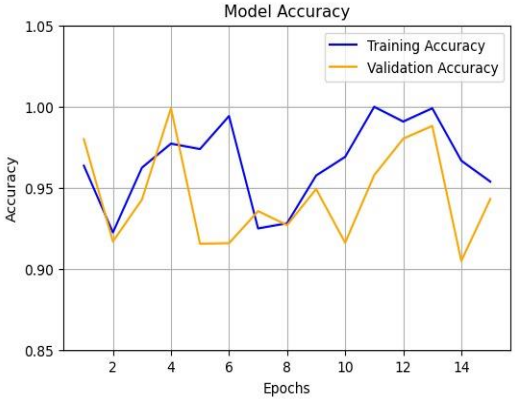
still exhibited commendable performance with an accuracy of 84%. A detailed comparison of the models' effectiveness is presented in the accompanying table[Table 1]

Model	Accuracy	recall	F1 score	Precision
MobileNet	97.50	0.47	0.49	0.49
ResNet	97.45	0.48	0.48	0.53
ConvNeXt	84.00	0.72	0.68	0.55

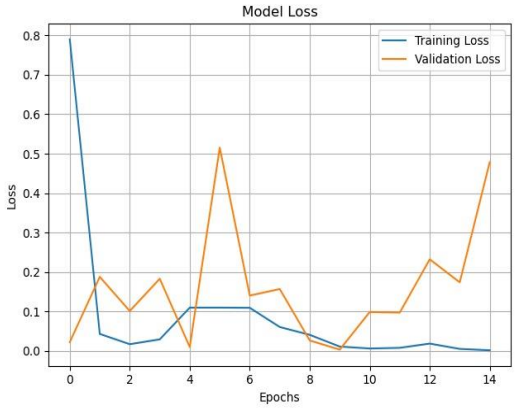
[Table 1] Model Evaluation table

Accuracy in percentage(%)

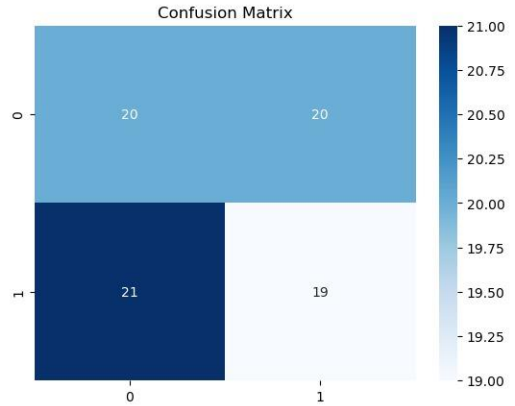
Furthermore, the findings are illustrated through graphs and confusion matrices, which provide enhanced insights into the classification performance and error analysis for each model.



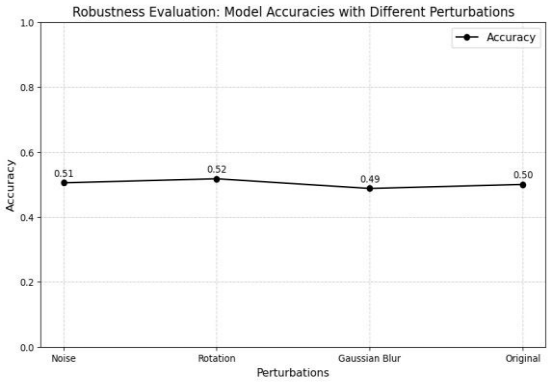
[Figure 5] Model Accuracy Graph of MobileNetV2



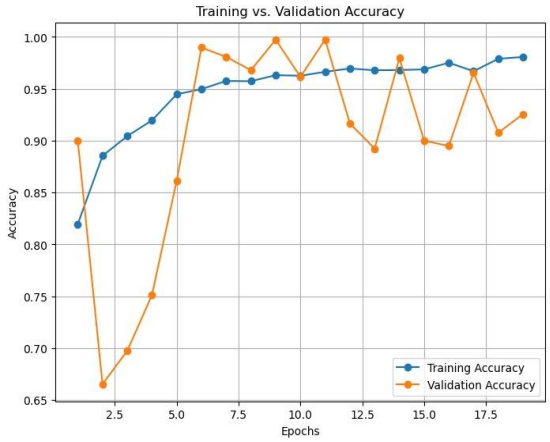
[Figure 6] Model Loss Graph of MobileNetV2



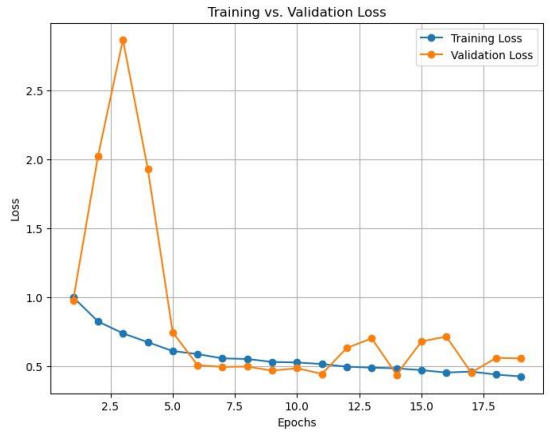
[Figure 7] Confusion Matrix of MobileNetV2 Model



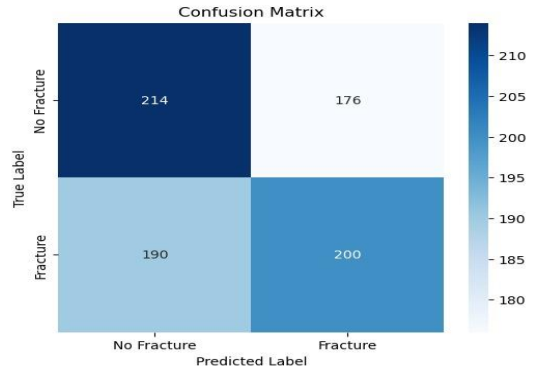
[Figure 8]Robustness accuracy curve for MobileNet50 model



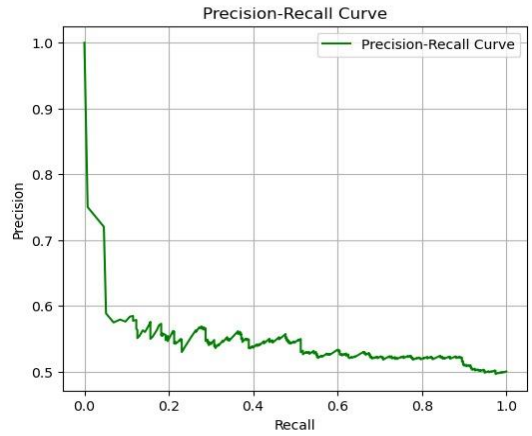
[Figure 9] Model Accuracy Graph of ResNet50



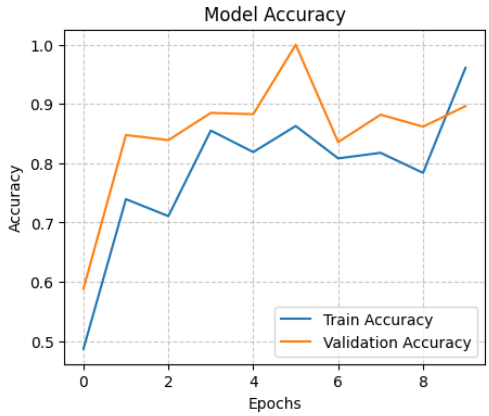
[Figure 10] Model Loss Graph of ResNet50



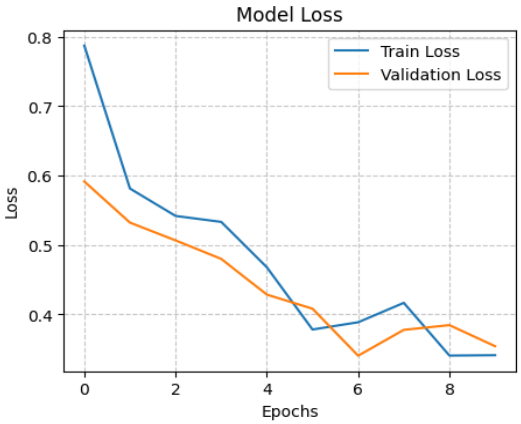
[Figure 11] Confusion Matrix of ResNet50 Model



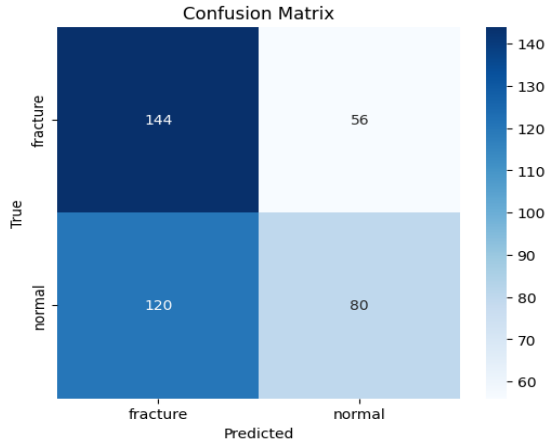
[Figure 12]Precision-Recall graph for Res-Net50 model



[Figure 13] Model Accuracy Graph of ConvNeXt Model



[Figure 14] Model Loss Graph of ConvNeXt



[Figure 15] Confusion Matrix of ConvNeXt Model

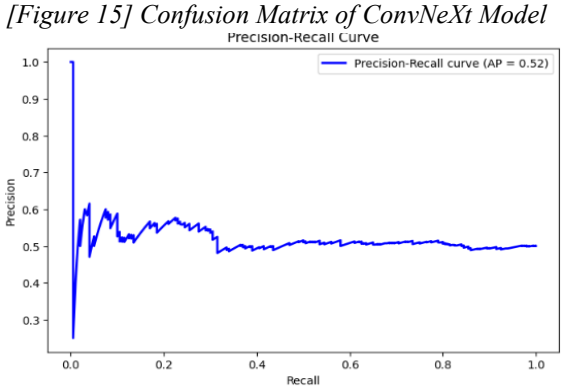
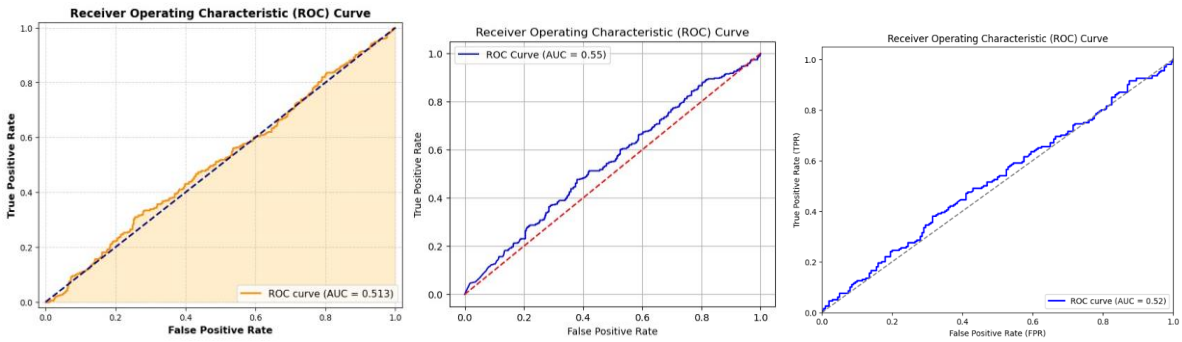


Figure 16]Precision recall graph for ConvNeXt model



[Figure 17]ROC(Receiver Operating Curve) curve for MobileNetV2, ResNet50 and ConvNeXt Tiny Models respectively.

VII. CONCLUSION

This study underscores the efficacy of deep learning techniques, particularly the MobileNetV2, ResNet50, and ConvNeXt Tiny architectures, in the identification of cervical spine fractures

through CT imaging. The findings illustrate how these models can significantly enhance diagnostic accuracy and decrease processing times, thereby contributing to improved clinical reliability and patient outcomes. Nonetheless, it is crucial to acknowledge that the results are derived from a dataset that, although substantial, would benefit from additional validation and testing in a wider array of clinical environments to ensure their generalizability. Furthermore, the performance of these models highlights the necessity for ongoing refinement and optimization to mitigate the risks associated with overfitting, especially when dealing with relatively limited datasets. In light of these factors, the knowledge acquired from this comparative study lays a solid groundwork for future developments in the automated detection of cervical spine fractures.

REFERENCES

[1] R. Behbahani Nejad et al., "Intelligent Cervical Spine Fracture Detection Using Deep Learning Methods," in arXiv:2311.05708 [cs.CV], 2023.

[2] A. Bansal et al., "CT Cervical Spine Fracture Detection Using a Convolutional Neural Network," in American Journal of Neuroradiology, 2021.

[3] J. Small et al., "Diagnostic accuracy of an artificial intelligence algorithm for cervical spine fracture detection on CT to attending radiologists," in European Radiology, 2024.

[4] "CT Cervical Spine Fracture Detection Using a Convolutional Neural Network," in American Journal of Neuroradiology, 2021.

[5] "Cervical Spine Fracture Detection and Classification Using," in IEEE, 2024.

[6] "Artificial Intelligence Detection of Cervical Spine Fractures," in e-Neurospine, 2023.

[7] "Deep Learning and Cloud-Based Computation for Cervical," in MDPI, 2023.

[8] "Comparison of three different data representations for," in ResearchGate, 2023.

[9] R. Behbahani Nejad et al., "Intelligent Cervical Spine Fracture Detection Using Deep Learning Methods," in arXiv:2311.05708, 2023.

[10] M. Gayathri et al., "A deep learning approach on cervical spine fracture detection," in International Journal of Novel Research and Development, 2020.

[11] Umang1103, "Cervical-Spine-Fracture-Detection: CNN based cervical spine fracture detection for vertebrae CT scan images," in GitHub, 2022.

[12] "NeckFrac: Deep Learning Powered Cervical Spine Fracture Detector," in University of California, Berkeley, School of Information, 2022.

[13] "A Survey on Deep Learning Techniques for Cervical Spine Fracture Detection," in International Journal of Advanced Research in Computer and Communication Engineering, 2023.

[14] Y. Dong et al., "Global incidence, prevalence, and disability of vertebral fractures: a systematic analysis of the global burden of disease study 2019," in The Spine Journal, 2022.

[15] D. Dreizin et al., "Multidetector CT of blunt cervical spine trauma in adults," in Radiographics, 2014.

[16] M.B.S. Bhavya, M.V. Pujitha, and G.L. Supraja, "Cervical Spine Fracture Detection Using Pytorch," in IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC), 2022.

[17] M.F.I. Aguirre, A.I. Tsirikos, and A. Clarke, "Spinal injuries in the elderly population," in Orthopaedics and Trauma, 2020.

[18] J.H. Bland, and D.R. Boushey, "Anatomy and physiology of the cervical spine," in Seminars in Arthritis and Rheumatism, 1990.

[19] N. Sugandhavesa et al., "Deep Learning and Cloud-Based Computation for Cervical Spine Fracture Detection System," in Electronics, 2023.

[20] T. Kanwal, M. Rabbia, A. Syed, I. Usama, K. Romail, S. Faisal, and Butt, "SPINE-EDL NET: ENSEMBLE APPROACH FOR CERVICAL SPINAL FRACTURE DETECTION," in Journal of Population Therapeutics and Clinical Pharmacology, 2024.