IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 10, October 2024

DOI: 10.17148/IARJSET.2024.111028

CERVICAL SPINE FRACTURE DETECTION USING DEEP LEARNING

Mutyam Bhargav Reddy[1], Sanjai srinath[2]

UG Student, VIT Bhopal Institute of Technology, Bhopal, India[1]

UG Student, VIT Bhopal Institute of Technology, Bhopal, India [2]

Abstract: Cervical spine fractures represent a critical issue in healthcare, especially among older adults, where identifying these injuries can be complicated by existing degenerative conditions. In light of this, a recent study proposes an innovative method that harnesses deep neural networks (DNNs), particularly the U-Net architecture, to streamline the detection of fractures in computed tomography (CT) images. The focus of this methodology is to accurately identify and localize the cervical vertebrae, which is essential for a reliable assessment of fractures. By utilizing the U-Net's strengths in semantic segmentation, the model can effectively outline the boundaries of cervical vertebrae, capturing detailed features and spatial relations within the images.

Additionally, the framework enhances the U-Net's fracture detection capabilities by adding multi-class classification layers, allowing it to differentiate between fractured and intact areas in the segmented vertebrae. This advancement significantly improves the diagnostic accuracy of the approach. Trained on a wide-ranging dataset of cervical spine injuries, this methodology presents substantial clinical benefits, such as facilitating real-time fracture assessments that lead to quicker diagnoses and timely interventions, ultimately enhancing patient outcomes. By tapping into the potential of deep learning, this approach promises to boost both the efficiency and accuracy of cervical spine fracture detection, contributing positively to patient care and treatment results.

Keywords: Key-words: U-Net architecture, computed tomography (CT) images, cervical vertebrae, semantic segmentation.

I. INTRODUCTION

Spinal fractures have become a pressing health issue worldwide, with over 1.5 million new cases reported each year. Cervical spine fractures pose serious challenges because of their critical location and the complications they may cause. The elderly population is especially vulnerable, often facing a higher risk due to conditions like osteoporosis and other degenerative diseases. Thus, the quick and precise detection of cervical spine fractures is essential for timely medical intervention and better patient outcomes. Recent advancements in medical image analysis, particularly the advancements in deep learning, have paved the way for new solutions. This project aims to utilize deep neural networks (DNNs) to create an automated method for detecting cervical spine fractures through computed tomography (CT) imaging. By leveraging the strengths of deep convolutional neural network (DCNN) architecture, this approach seeks to improve the accuracy and efficiency of fracture detection significantly. Traditionally, identifying cervical spine fractures has relied heavily on skilled healthcare professionals' manual interpretation of radiographic images. This process is not only time-consuming but also prone to human error, especially when the fractures are subtle or masked by degenerative changes. Additionally, the presence of coexisting degenerative diseases and osteoporosis complicates detection in older patients. To tackle these challenges, our project introduces an innovative strategy focused on deep neural networks, particularly the U-Net architecture. U-Net is well-known for its effectiveness in semantic segmentation tasks, making it ideal for accurately outlining cervical vertebrae in medical images. By utilizing U-Net's capabilities, this framework aims to deliver outstanding performance in segmenting the intricate structures of the cervical spine. Moreover, we enhance U-Net's functionality for fracture detection by incorporating multi-class classification layers. This enhancement allows the model to differentiate between fractured and intact areas within the segmented cervical vertebrae, thus improving diagnostic accuracy. With extensive training in diverse cervical spine injury radiographic images, this framework learns to identify and locate fractures with remarkable precision. Evaluation of the U-Net model shows it outperforms traditional methods and other deep learning architectures in segmentation accuracy and fracture detection sensitivity, offering significant potential for improving clinical practices in spinal fracture diagnostics. The methodology provides a key benefit by enabling real-time evaluation of fractures, which supports quick diagnosis and timely interventions to improve patient care.

DOI: 10.17148/IARJSET.2024.111028

II. LITERATUREREVIEW

This writing survey will investigate later thinks about that examine the application of profound learning strategies such as CNNs, repetitive neural systems (RNNs), and advanced division models like U Net and Quicker R-CNN for spinal break discovery and classification. By synthesizing the discoveries of these ponders, this audit points to recognize patterns, challenges, and future inquire about headings in the field of computerized spinal harm conclusion. Germann et al. (2023) illustrated the potential of profound convolutional neural networks (DCNNs) in precisely measuring vertebral bodies and recognizing lacking breaks in lumbar spine MRI. Their consider underscores the utility of DCNNs in achieving exact analyze, advertising promising prospects for moved forward quiet care [1]. Bhavya et al. (2022) utilized DCNNs to separate between traumatic and non-traumatic causes of cervical spine fractures from CT looks. Their work pointed to accomplish exact classification, highlighting the part of profound learning in improving symptomatic precision and directing appropriate treatment procedures [2]. Little et al. (2021) proposed a profound learning show that coordinating convolutional neural systems (CNNs) with bidirectional long short term memory (BLSTM) for computerized location of cervical spine breaks in CT pivotal pictures. Their approach guarantees proficient determination, possibly streamlining clinical workflows and speeding up persistent care [3]. Vong and Dinh (2021) illustrated tall exactness in therapeutic picture examination by utilizing UNet++ with EfficientNet for pneumothorax division in chest X-ray pictures. Their think about exhibits the viability of progressed profound learning designs in accomplishing exact division, fundamental for accurate conclusion [4]. Ahmad et al. (2021) presented MH UNet, a novel engineering that accomplished state-of-the-art execution in restorative picture division. Their work signifies noteworthy progressions in profound learning models, clearing the way for more precise and proficient determination of different therapeutic conditions [5]. Salehinejad et al. (2021) proposed a profound successive learning show for cervical spine break location on CT imaging. Their approach upgrades demonstrative capabilities in spinal injury evaluation, advertising potential advancements in persistent results and treatment arranging [6]. Zhao et al. (2021) upgraded restorative picture division exactness through the joining of attention components in CBAM-Unet++. Their work exhibits the significance of consideration components in progressing division precision, vital for exact conclusion and treatment arranging [7]. Saini and Sood (2021) presented a altered UNet++ demonstrate for lung division in chest X-ray pictures, accomplishing tall exactness in picture examination assignments. Their approach illustrates the adequacy of customized designs in tending to specific demonstrative challenges [8]. Sha et al. (2020) illustrated promising comes about in injury discovery utilizing an moved forward form of the YOLOv2 calculation for identifying spinal fracture injuries. Their consider highlights the potential of progressed protest location methods in moving forward symptomatic capabilities [9]. Sha et al. (2020) emphasized the significance of exact localization in spinal break location by utilizing an moved forward Speedier R-CNN demonstrate. Their work underscores the noteworthiness of exact localization for guiding suitable treatment techniques and moving forward quiet results [10]. Liu et al. (2020) proposed a multi-receptive-field CNN for semantic division of restorative images, contributing to moved forward division precision and productivity. Their approach offers potential progressions in computerized picture investigation, encouraging more precise analyze [11].Ahammad et al. (2019) created a quick and precise division system for spinal line damage seriousness classification. Their work advances demonstrative capabilities in damage appraisal, possibly supporting clinicians in making educated treatment choices [12]. Tomita et al. (2018) showcased the effectiveness of DCNNs in early location of osteoporotic vertebral breaks in CT examinations. Their consider underscores the utility of profound learning in break discovery, offering promising prospects for moving forward quiet care and treatment results [13]. Generally, these ponders collectively highlight the transformative potential of profound learning in revolutionizing the determination and administration of spinal wounds and breaks. By leveraging progressed profound learning procedures and structures, clinicians can upgrade demonstrative exactness, streamline clinical workflows and eventually make strides in persistent results.

III. IMPLEMENTATION & METHODOLOGY

This deep learning-based system for cervical spine fracture detection integrates various interconnected modules, each addressing a specific stage in the workflow. The methodology is designed to deliver high accuracy and efficiency, suitable for clinical applications. Below is a detailed breakdown of the workflow:

System Architecture (Fig-1)The architecture consists of ten primary modules, which ensure seamless data processing, model training, and deployment.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⋇ Peer-reviewed / Refereed journal ⋇ Vol. 11, Issue 10, October 2024

DOI: 10.17148/IARJSET.2024.111028

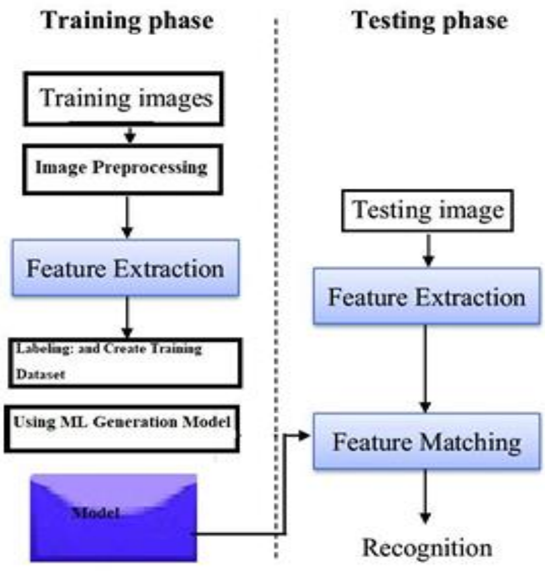


Fig 1: System Architecture

III.I Load Dataset Module

Purpose: This module manages the acquisition of cervical spine image datasets, which typically include medical imaging modalities such as X-rays, CT scans, or MRIs.

Process:

Reads image files (e.g., DICOM, JPEG, PNG) from specified directories. Organizes the dataset into a structured format, including metadata (e.g., patient ID, imaging modality). Ensures the dataset contains diverse samples to improve model generalization.

III.II Data Cleaning and Preprocessing

Purpose:

Prepares raw data for use in the machine learning pipeline by removing artifacts and inconsistencies.

Tasks: Artifact Removal: Eliminates unwanted elements like scanning artifacts or patient identification markers in images.

Standardization: Converts all images to a uniform format (e.g., grayscale or RGB).

Normalization: Adjusts pixel values to a consistent range (e.g., 0 to 1 or -1 to 1) to enhance model convergence during training.

III.III Image Resizing

Purpose: Resizes images to a uniform resolution to meet the input size requirements of the neural network.

Process:

Applies resizing algorithms (e.g., bilinear or nearest-neighbor interpolation).

Maintains aspect ratios when possible to preserve anatomical integrity.

Standard dimensions (e.g., 224x224 or 256x256 pixels) are chosen for compatibility with pre-trained models.

III.IV Feature Extraction

Purpose: Identifies critical features in preprocessed images, such as edges, textures, and patterns, that distinguish between fractured and intact vertebrae.

Process:

Employs techniques like edge detection, histogram analysis, or gradient computation.

DOI: 10.17148/IARJSET.2024.111028

Prepares feature maps that highlight vertebral contours and anomalies, aiding the model in learning spatial and structural differences.

III.V Labeling

Purpose: Annotates images with ground truth labels to indicate the presence or absence of fractures and the vertebral class (C1 to C7).

Process:

Uses manual annotation by radiologists or automated labeling tools.

Labels include binary classification (fractured/not fractured) and multi-class labels for vertebrae identification.

III.VI Create Training and Testing Datasets

Purpose: Splits the annotated dataset into subsets for model training and evaluation.

Process:

Ensures an appropriate ratio for training and testing datasets (e.g., 80:20 or 70:30).

Balances the dataset to prevent class imbalance (e.g., using oversampling or data augmentation for underrepresented classes).

III.VII U-Net Model Generation

Purpose: Implements the U-Net architecture for precise segmentation and feature extraction of cervical vertebrae and fractures.

Key Features:

Encoder-Decoder Structure: Captures contextual information and refines image details.

Skip Connections: Preserve spatial information by linking encoder and decoder layers.

Training: Optimizes the model using labeled training data and a loss function such as cross-entropy or Dice loss.

III.VIII Transfer Learning with DenseNet

Purpose: Enhances fracture classification by leveraging pre-trained DenseNet models, fine-tuned with output features from the U-Net.

Process:

Feature Integration: Combines U-Net outputs with DenseNet's pre-trained knowledge.

Fine-Tuning: Updates model weights using cervical spine images to adapt DenseNet for fracture classification.

Advantages: Reduces training time and improves performance by reusing robust feature representations.

III.IX Deploy Model

Purpose: Translates the trained model into a deployable application for real-time fracture detection in clinical settings.

Process:

Converts the model into an optimized format (e.g., ONNX, TensorRT) for faster inference.

Integrates with hardware accelerators (e.g., GPUs or TPUs) for improved performance.

III.X Flask Framework Integration

Purpose: Provides a user-friendly interface for interacting with the fracture detection system.

Implementation:

Endpoints: Creates routes for uploading images, retrieving predictions, and managing user sessions.

User Interaction: Allows clinicians to upload cervical spine images and receive immediate predictions on fracture presence and vertebra class.

Scalability: Enables integration with hospital information systems (HIS) for seamless workflows.

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 10, October 2024

DOI: 10.17148/IARJSET.2024.111028

End-to-End Workflow

The system combines data preprocessing, neural network training, and real-time deployment:

Data Handling: Ensures high-quality input for training and evaluation.

Deep Learning Models: Exploits U-Net for segmentation and DenseNet for classification, ensuring robust fracture detection.

Deployment: Delivers an interactive solution that aids clinical decision-making, ensuring accuracy and efficiency.

IV.I COMPONENT - BASED ARCHITECTURE:

➢ **Image Processing Component:**

This component takes charge of essential image preprocessing tasks such as normalization, resizing, and noise reduction, all aimed at preparing CT scan images for effective fracture detection.

➢ **Deep Learning Component:**

This part implements a neural network architecture dedicated to fracture detection and classification. It features modules for model training, inference, and optimization to enhance performance.

➢ **Data Management Component:**

This component oversees the storage and retrieval of critical data, including patient metadata, CT scan images, and model parameters. It plays a vital role in ensuring data consistency, integrity, and security throughout the entire system.

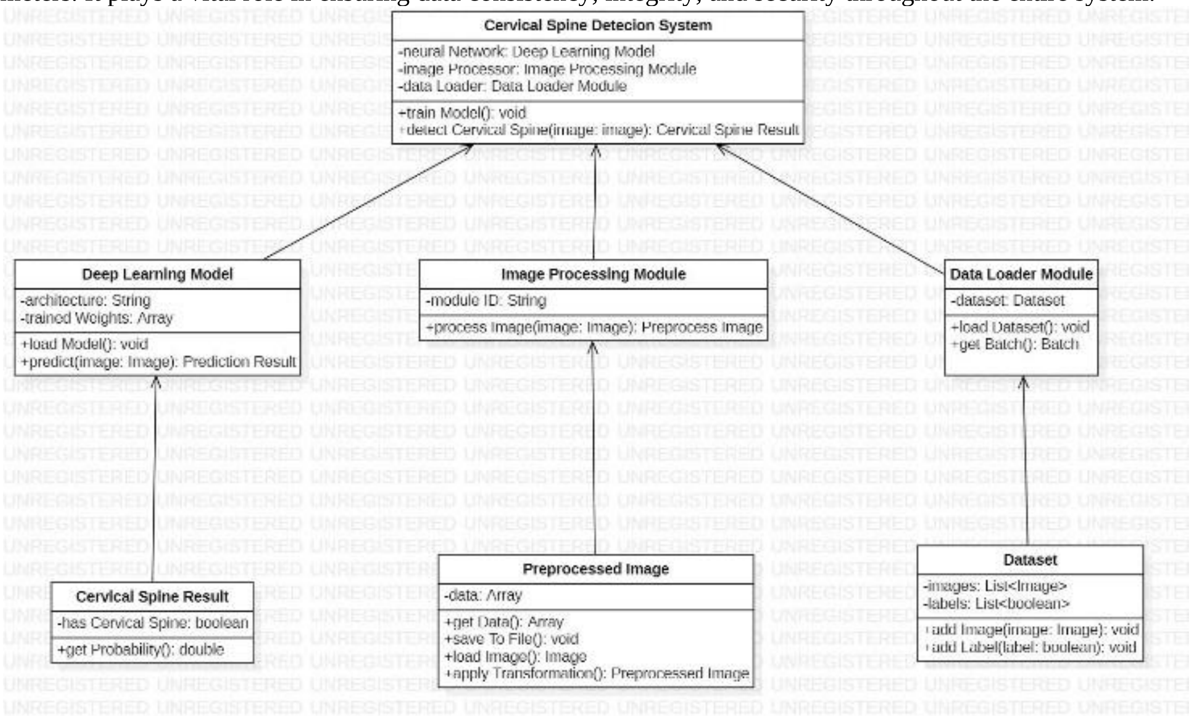


Fig 2: Component-Based Architecture

IV.II USE CASE ARCHITECTURE

The Use Case Diagram illustrates the various use cases or functionalities of the system from the perspective of users. It shows how users interact with the system to accomplish specific tasks or goals.

DOI: 10.17148/IARJSET.2024.111028

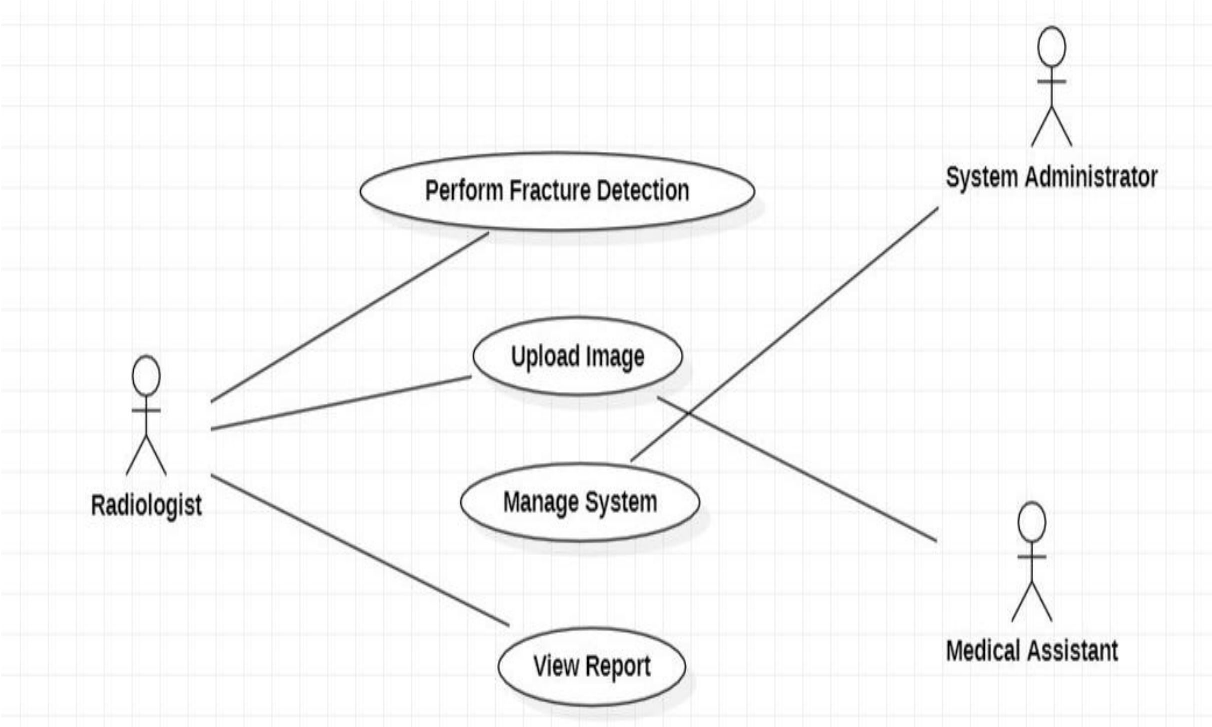
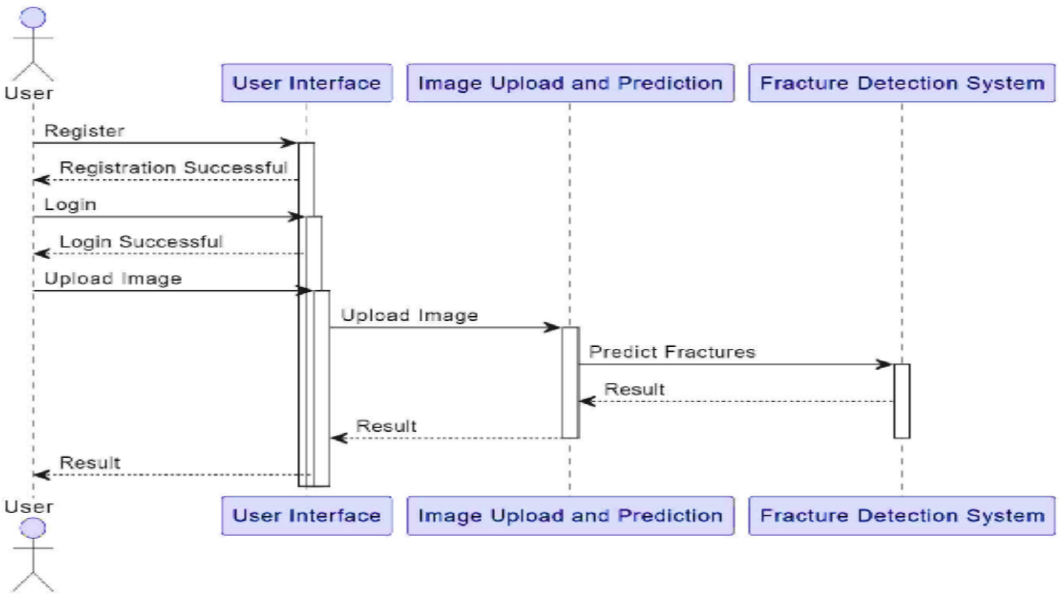


Fig 3: Use Case Architecture

IV.III SEQUENCE ARCHITECTURE

The Sequence Diagram illustrates the interactions between different components or objects in a sequential manner. It shows the sequence of messages exchanged between objects over time, depicting the flow of control or communication within the system.



IV.IV STATE CHART ARCHITECTURE

The State Chart Diagram models the behavior of individual objects or components by depicting their states and transitions. It shows how objects transition from one state to another in response to events or stimuli.

DOI: 10.17148/IARJSET.2024.111028

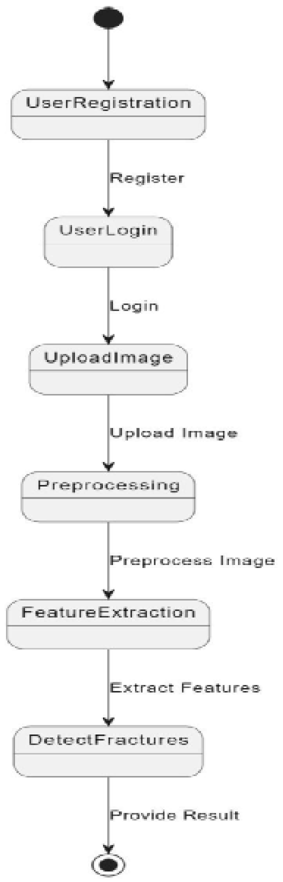


Fig 5: State Chart Architecture

IV.V DEPLOYMENT ARCHITECTURE

The Deployment Diagram illustrates the physical deployment of software components across different hardware nodes. It shows how software components are distributed across servers, computers, or other hardware devices.

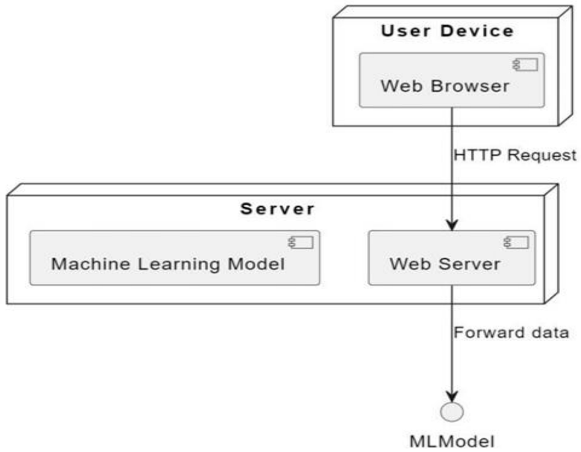


Fig 6: Deployment Architecture

IV.VI ACTIVITY ARCHITECTURE

The Activity Diagram represents the flow of control or workflow within the system, depicting the sequence of activities or actions performed to achieve a particular task or functionality.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Impact Factor 8.066 ⵋ Peer-reviewed / Refereed journal ⵋ Vol. 11, Issue 10, October 2024
DOI: 10.17148/IARJSET.2024.111028

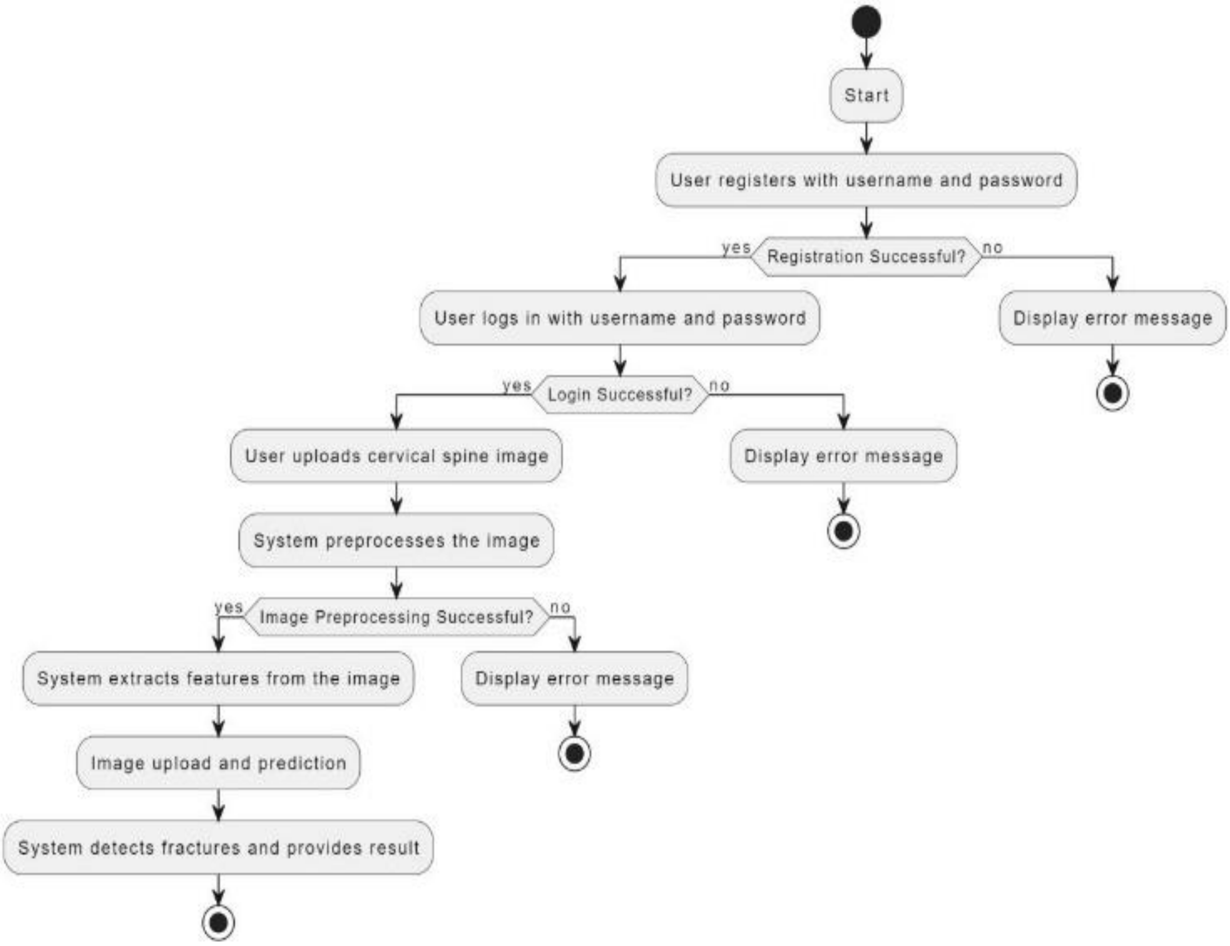


Fig 7: Activity Architecture

IV.VII. COMPONENT ARCHITECTURE

The Component Diagram depicts the physical components or modules of the system and their dependencies. It shows how the system is decomposed into smaller components, illustrating the relationships between them.

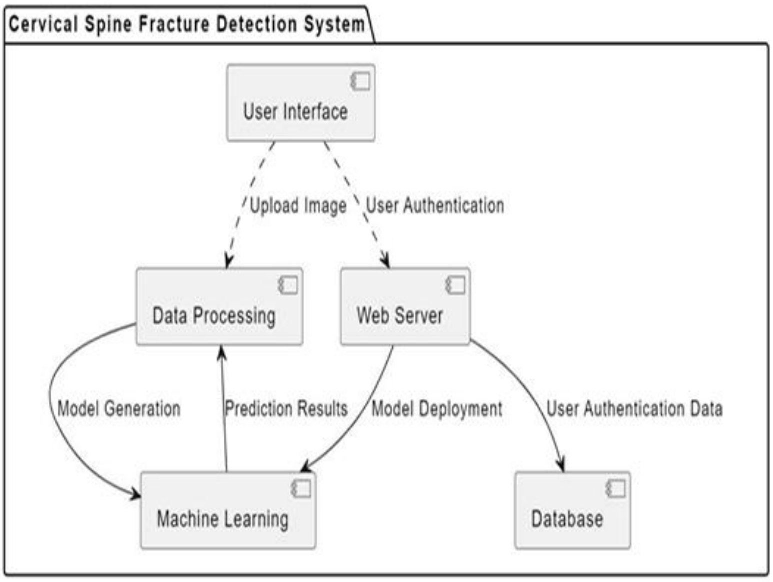


Fig 8: Component Architecture

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 10, October 2024

DOI: 10.17148/IARJSET.2024.111028

V. USED ALGORITHMS

V.I U-Net Architecture: The U-Net architecture is a convolutional neural network (CNN) designed for semantic segmentation tasks, known for its effectiveness in medical image analysis. UNet consists of an encoder-decoder structure with skip connections that allow the network to capture both high-level features and detailed spatial information. The encoder downsamples the input image to extract features, while the decoder upsamples the features to generate the segmentation mask. Skip connections concatenate feature maps from the encoder to the decoder, localization.

ENCODER:

$Zi = \mathrm{Conv}i(\mathrm{ReLU}(\mathrm{Conv}i-1(zi-1)))$

DECODER:

$Yi=\mathrm{Conv}i(\mathrm{ReLU}(\mathrm{Conv}i-1(\mathrm{Concat}(zi-1,yi-1)))$

V.II DenseNet for Transfer Learning: DenseNet is a CNN architecture characterized by dense connections between layers, allowing each layer to directly receive input from all its preceding layers. In transfer learning, a pre-trained DenseNet model, trained on a large dataset like ImageNet, is fine-tuned on the specific task of cervical spine fracture detection. The dense connections facilitate feature reuse and enable the model to learn discriminative features from the limited medical imaging dataset.

Dense Block: x_{l+1} = H_l([x_0, x_1, ..., x_l]), where (x_{l+1}\) represents the output feature maps of the (l+1)th layer, (H_l\) denotes the layer function, and ([x_0, x_1, ..., x_l]\) denotes the concatenation of feature maps from all preceding layers.

V.III Binary Cross-Entropy Loss Function:

Description: Binary cross-entropy is a loss function commonly used for binary classification tasks, such as fracture detection, where each pixel is classified as fractured or intact.

Explanation: Binary cross-entropy measures the dissimilarity between the predicted probability distribution and the ground truth labels. It penalizes deviations from the true labels, encouraging the model to produce accurate predictions.

L(y,y^)=−N1∑i=1N[yilog(y^i)+(1−yi)log(1−y^i)]

where (y) represents the ground truth labels, (hat{y}) represents the predicted probability distribution, and (N) denotes the number of pixels.

V.IV Stochastic Gradient Descent (SGD) Optimization: SGD is an optimization algorithm commonly used to minimize the loss function during model training. SGD updates the model parameters iteratively by computing the gradient of the loss function with respect to the parameters and adjusting the parameters in the opposite direction of the gradient. It aims to find the optimal set of parameters that minimize the loss. **Mathematical Equation:** θt+1=θt−η∇L(θt), where θtθt represents the model parameters at iteration tt, ηη denotes the learning rate, and $\nabla L(\theta t)$∇L(θt) represents the gradient of the loss function with respect to the parameters.

VI. TECHNICAL IMPLEMENTATION & ANALYSIS

MODULE DESCRIPTION:

➢ **Create Training and Testing Dataset:**

Description: This module splits the labeled dataset into training and testing subsets for model evaluation. **Functionality:** It partitions the dataset into separate sets for training the model (to learn patterns) and testing the model's performance (to assess accuracy).

➢ **U-Net Model Generation:**

Description: This module implements the U-Net architecture to train a deep neural network for fracture detection. **Functionality:** It constructs the U-Net model using convolutional layers and performs training using the labeled training dataset to learn to identify fractures and vertebrae classes.

➢ **Transfer Learning:**

Description: After U-Net model generation, transfer learning with the DenseNet architecture is applied.

DOI: 10.17148/IARJSET.2024.111028

- **Functionality:** Fine-tune a pre-trained DenseNet model using the given dataset to extract features which are then feeded as input to the decoder of U-Net model. This enhances the model's ability to classify fractures by leveraging knowledge from the pre-trained network, thereby improving performance, especially when data is limited.

- **Deploy Model:**

Description: This module deploys the trained model for real-time fracture detection. **Functionality:** It integrates the trained model into a deployable application environment, ready for user interaction.

- **Flask Framework Integration:**

Description: This module integrates the trained model with the Flask web framework to create a user-friendly interface. **Functionality:** It sets up endpoints for user authentication, image uploading, and prediction result retrieval using Flask routes.

- **User Authentication Module:**

Description: This module manages user authentication, allowing users to sign up and log in to the application securely. **Functionality:** It verifies user credentials, handles user sessions, and provides access control to protected resources.

VII. TESTING

VII.I FUNCTIONAL TESTING: Functional tests give orderly shows that capacities tried are accessible as specified by the trade and specialized prerequisites, framework documentation, and client manuals. Functional testing is centered on taking after things.

Valid Input: recognized classes of substantial input must be accepted**.** **Invalid Input:** distinguished classes of invalid input must be rejected.

Functions: distinguished capacities must be worked out **Output:** recognized classes of application yields must be exercised.

Systems/Procedures: meddle frameworks or methods must be conjured. Organization and planning of useful tests is centered on prerequisites, key capacities, or special test cases. In expansion, orderly scope relating to recognize Commerce handle flows; information areas, predefined forms, and progressive forms must be considered for testing. Sometime recently utilitarian testing is total, extra tests are distinguished and the viable value of current tests is decided.

VII.II SYSTEM TEST

System testing guarantees that the whole coordinates program framework meets prerequisites. It tests a configuration to guarantee known and unsurprising come about. An illustration of framework testing is the configuration situated framework integration test. Framework testing is based on handle portrayals and streams, emphasizing pre-driven handle joins and integration focuses.

VII.III WHITE BOX TESTING

White Box Testing is a testing in which in which the program analyzer has information of the inward workings, structure and dialect of the computer program, or at slightest its reason. It is reason. It is utilized to test zones that cannot be come to from a dark box level**.**

VII.IV BLACK BOX TESTING

Black Box Testing is testing the computer program without any information of the internal workings, structure, or dialect of the module being tried. Dark box tests, like mot other sorts of tests, must be composed of authoritative source archive, such as detail or prerequisites document, such as detail or necessities record. It is a testing in which the program under test is treated, as a dark box .you cannot "see" into it. The test gives inputs and responds to yields without considering how the computer program works.

DOI: 10.17148/IARJSET.2024.111028

VII.V UNIT TESTING

Unit testing includes the plan of test cases that approve that the inside program rationale is functioning appropriately, and that program inputs deliver substantial yields. All choice branches and internal code stream ought to be approved. It is the testing of person program units of the application .it is done after the completion of an person unit some time recently integration. This is a structural testing, that depends on information of its development and is obtrusive. Unit tests perform basic tests at component level and test a particular commerce handle, application, and/or framework configuration. Unit tests guarantee that each interesting way of a commerce handle performs precisely to the archived details and contains clearly characterized inputs and anticipated comes about.

Unit testing is more often than not conducted as portion of a combined code and unit test stage of the program lifecycle, in spite of the fact that it is not unprecedented for coding and unit testing to be conducted as two distinct stages

Test procedure and approach

Field testing will be performed physically and utilitarian tests will be composed in detail

Test goals

- All field passages must work legitimately.
- Pages must be actuated from the distinguished interface.
- The passage screen, messages and reactions must not be postponed.

Features to be tried

- Confirm that the sections are of the rectify organize
- No copy passages ought to be permitted
- All joins ought to take the client to the redress page.

VII. VI INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more coordinates software components on a single stage to create disappointments caused by interface absconds. The errand of the integration test is to check that components or computer program applications, **e.g.** components in a program framework or – one step up – computer program applications at the company level – associated without blunder. Integration tests are outlined to test coordinates computer program components to decide if they actually run as one program. Testing is occasion driven and is more concerned with the essential outcome of screens or areas. Integration tests illustrate that in spite of the fact that the components were individually fulfillment, as appeared by effectively unit testing, the combination of components is adjust and steady. Integration testing is particularly pointed at uncovering the issues that arise from the combination of components.

Test Results: All the test cases said over passed effectively. No surrenders experienced.

VII.VII ACCEPTANCE TESTING

User Acknowledgment Testing is a basic stage of any extend and requires noteworthy cooperation by the conclusion client. It moreover guarantees that the framework meets the utilitarian necessities.

Test Results: All the test cases said over passed effectively. No abandons experienced**.**

DOI: 10.17148/IARJSET.2024.111028

VIII. TEST CASES

S.NO	Test Name	Inputs	Process	Expected Output	Actual Output	Status
1)	Input data validation	Images (JPEG, Png)	Test the system's response when provided with invalid or corrupted image data.	Takes all the formats of images except corrupted.	Works on all the images except corrupted.	Success
2)	Detection accuracy	labeled images	Test the system's ability to correctly identify cervical spine fractures in a set of labeled images.	Measures accuracy, sensitivity and specificity.	Results of all the metrics to predict the fracture.	Success
3)	Boundary cases	Images with fractures	Test the system's performance on images with minor and severe fractures.	Detects the fracture and its type.	Kind of fracture that has occurred.	Success
4)	Performance load	Multiple images	Test the system's response time under varying loads.	Can handle multiple images at a time.	More than one image can be accessed at a time.	Success

IX. RESULTS

The implementation of the cervical spine fracture detection system yielded promising outcomes, as depicted by the user interface screenshots and prediction results. This section elaborates on the system's functionalities and the interpretation of its outputs.

Training Phase Results: The training phase of the system involved data processing and modeling, as illustrated in Figures 9 and 10, respectively. These screenshots demonstrate the system's capability to preprocess medical imaging data and develop a model for fracture detection.

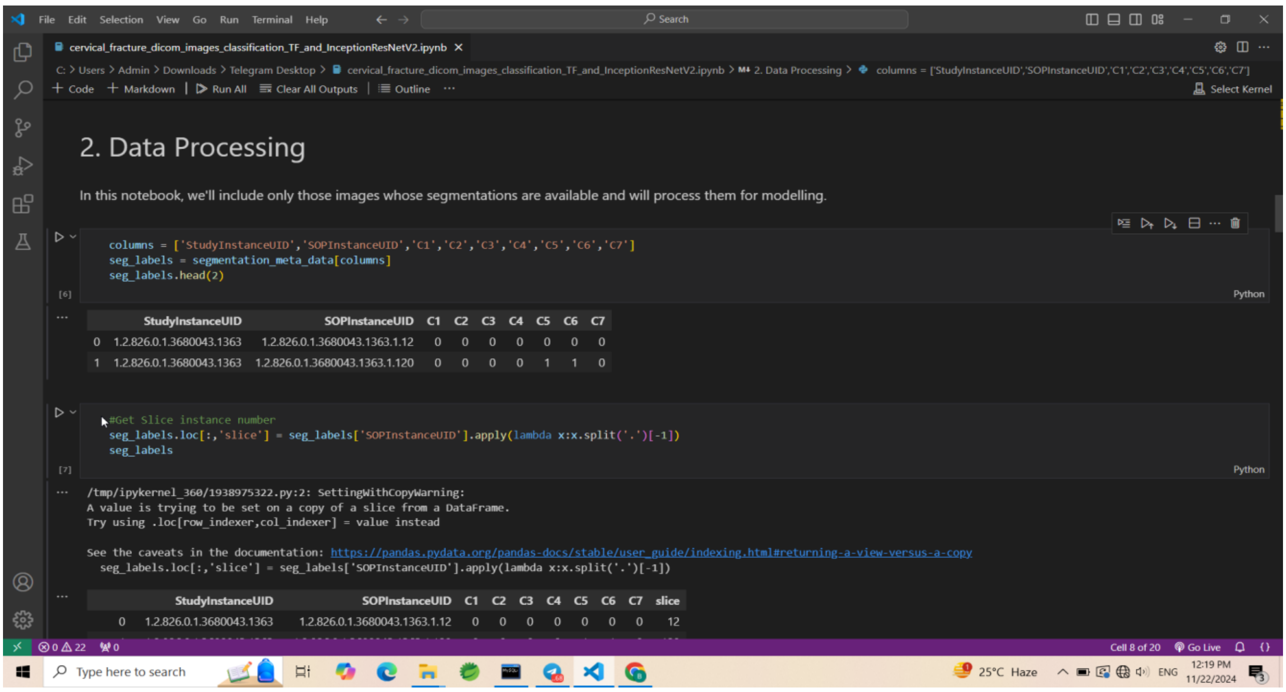


Fig 9: Data Processing

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 10, October 2024

DOI: 10.17148/IARJSET.2024.111028

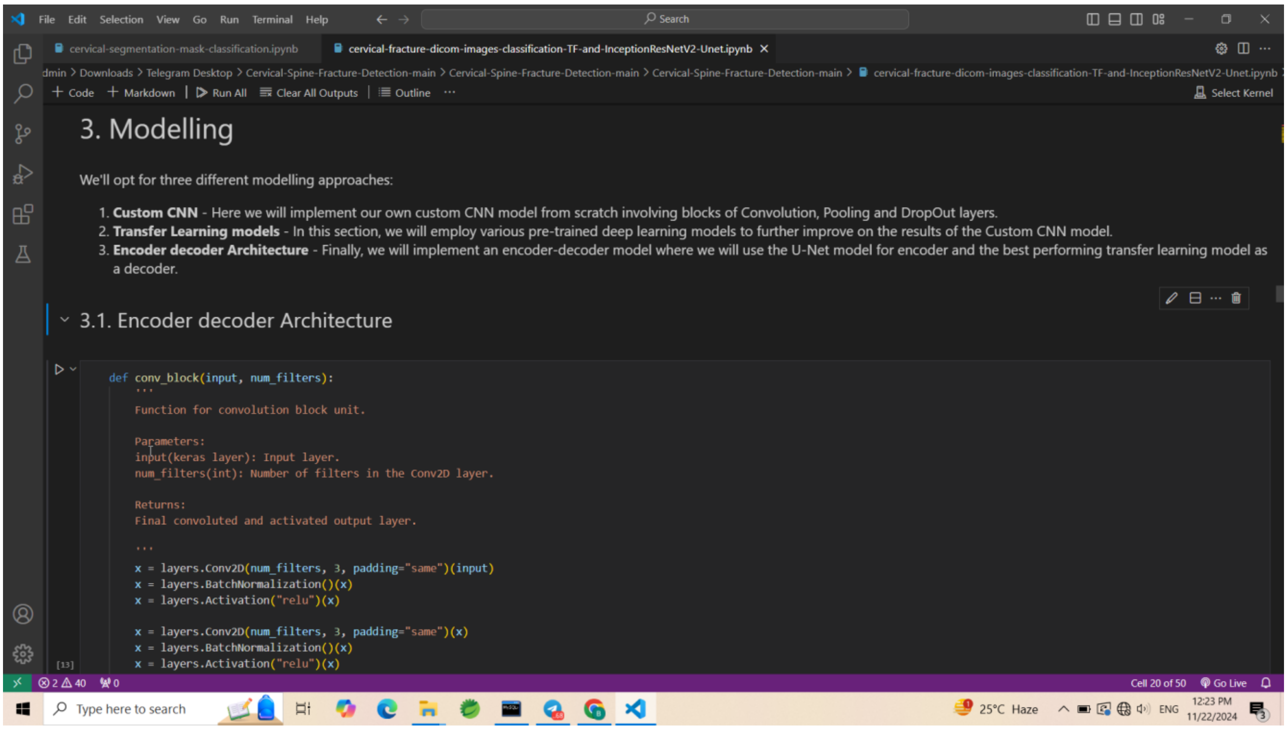


Fig 10: Model generation

Prediction Phase Results: During the prediction phase, users interact with the system through an upload page, as depicted in Figure 11. This feature enables users to input cervical spine images for fracture detection conveniently. Subsequently, the system generates prediction results, indicating whether a fracture is detected or not, as illustrated in Figures 12 and 13.

Interpretation of Prediction Results: The prediction results provide valuable insights into the system's performance. When a fracture is detected (Figure 12), it signifies the system's ability to accurately identify pathological abnormalities in the cervical spine images. Conversely, when no fracture is detected (Figure 13), it suggests the absence of evident abnormalities, thereby reinforcing the system's specificity and reliability.

Fig 11: Upload CT image

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 11, Issue 10, October 2024

DOI: 10.17148/IARJSET.2024.111028

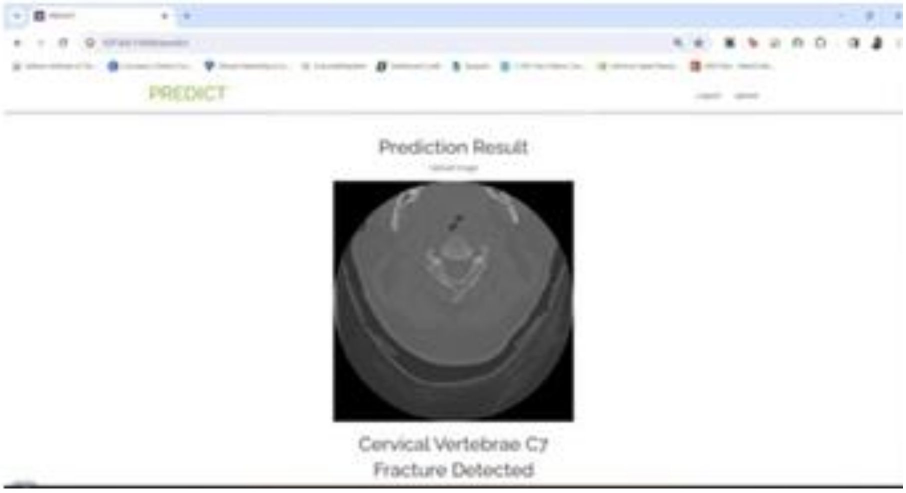


Fig 12: Fracture is detected

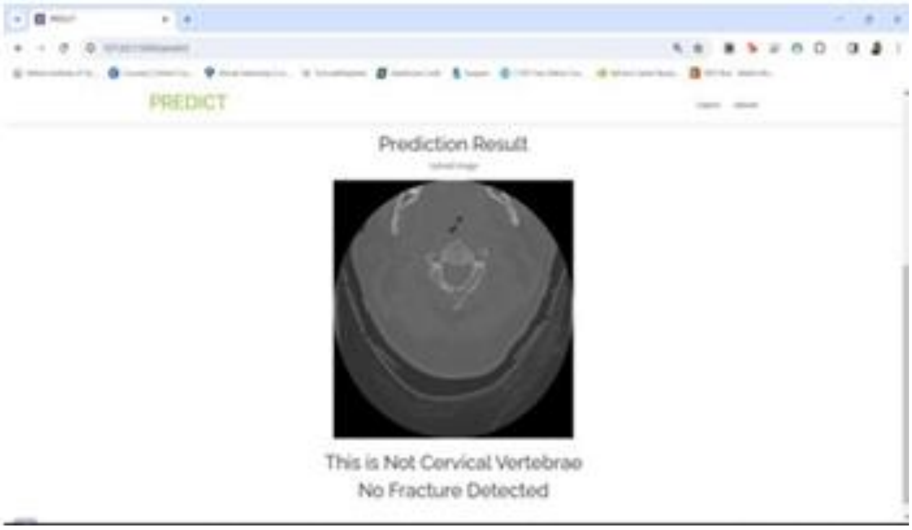


Fig 13: No Fracture is detected

X. CONCLUSION

Conclusion: The automated cervical spine fracture detection system, employing deep neural networks and transfer learning, marks a substantial leap forward in medical imaging. By accurately delineating vertebrae boundaries and discerning fractures in CT images, it significantly improves diagnosis efficiency. Integration with Flask ensures user-friendly interaction. Minimizing human error, it enables timely interventions, potentially transforming fracture diagnosis and patient care. Further advancements in technology promise continued refinement, driving improvements in fracture detection and patient outcomes.

Future Scope: Future enhancements for "Cervical Spine Fracture Detection Using Deep Neural Networks" aim to bolster model performance, expand capabilities, and integrate seamlessly into clinical workflows. Key areas for development include leveraging advanced deep learning architectures like Vision Transformers and Graph Neural Networks for improved spatial understanding. Multi-modal fusion, incorporating clinical data, and enhancing interpretability with Explainable AI (XAI) techniques can enhance diagnostic accuracy and clinical decision-making. Real-time applications require optimizing model efficiency for rapid analysis in emergency settings. Continuous learning approaches and collaborative research partnerships can ensure the model stays current and effective in diverse healthcare settings. Regulatory approval and commercialization pathways are essential for translating research findings into tangible clinical solutions, ensuring widespread adoption and impact on patient care.

DOI: 10.17148/IARJSET.2024.111028

REFERENCES

[1]. C. Germann, A.N. Meyer, M. Staib, et al., "Performance of a Deep Convolutional Neural Network for MRI Based Vertebral Body Measurements and Insufficiency Fracture Detection," European Radiology, 2023.

[2]. M.B.S. Bhavya, M.V. Pujitha, and G.L. Supraja, "Cervical Spine Fracture Detection Using Pytorch," 2022.

[3]. Y. Liu, Z. Ma, X. Liu, S. Ma, K. Ren, "Privacy-Preserving Object Detection for Medical Images With Faster R CNN," IEEE Transactions on Information Forensics and Security, 2022.

[4]. S. Iyer, A. Blair, C. White, L. Dawes, D. Moses, A. Sowmya, "Vertebral Compression Fracture detection using Multiple Instance Learning and Majority Voting," 2022 26th International Conference on Pattern Recognition (ICPR).

[5]. J.E. Small, P. Osler, A.B. Paul, M. Kunst, "CT Cervical Spine Fracture Detection Using a Convolutional Neural Network," AJNR American Journal of Neuroradiology, 2021.

[6]. K.-M. Vong, T.B. Dinh, "Pneumothorax Segmentation In Chest X-Rays Using UNet++ And EfficientNet," 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI).

[7]. P. Ahmad et al., "MH UNet: A Multi Scale Hierarchical Based Architecture for Medical Image Segmentation," IEEE Access, 2021.

[8]. H. Salehinejad et al., "Deep Sequential Learning For Cervical Spine Fracture Detection On Computed Tomography Imaging," 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI).

[9]. Z. Zhao, K. Chen, S. Yamane, "CBAM Unet++: Easier to Find the Target with the Attention Module "CBAM"," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE).

[10]. N. Saini, N. Sood, "Modified UNet++ Model: A Deep Model for Automatic Segmentation of Lungs from Chest X ray Images," 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC).

[11]. G. Sha, J. Wu, B. Yu, "Detection of Spinal Fracture Lesions based on Improved Yolov2," 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA).

[12]. P. Shajudeen et al., "Modeling and Analysis of Ultrasound Elastographic Axial Strains for Spine Fracture Identification," May 2020.

[13]. G. Sha, J. Wu, B. Yu, "Detection of Spinal Fracture Lesions Based on Improved Faster-RCNN," 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS).

[14]. L. Liu, F.-X. Wu, Y.-P. Wang, J. Wang, "Multi-Receptive-Field CNN for Semantic Segmentation of Medical Images," IEEE Journal of Biomedical and Health Informatics, 2020.

[15]. S.H. Ahammad, V. Rajesh, and M.Z.U. Rahman, "Fast and Accurate Feature Extraction-Based Segmentation Framework for Spinal Cord Injury Severity Classification," 2019.

[16]. N. Tomita, Y.Y. Cheung, S. automatic detection of osteoporotic vertebral fractures on CT scans," Computers in Biology and Medicine, 2018.

[17]. M. Mustra, M. Grgic, B. Zovko-Cihlar, "Alignment of X-ray bone images," 2014 X International Symposium on Telecommunications (BIHTEL). Hassanpour, "Deep neural networks for

