www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 48-53

e-ISSN : 2583-1062
Impact Factor : 7.001

BONE FRACTURE DETECTION USING DEEP LEARNING

Mrs. T. Ratnamala[1], Jaina Shivani[2], Rayabarapu Sri Mallika[3], Ravinuthala Meri Prasanna[4], Tudhu Vaishnavi[5]

[1]Assistant Professor, ACE Engineering College Hyderabad, India.

[2,3,4,5]Student, ACE Engineering College Hyderabad, India.

Email, t.ratnamala1990@gmail.com, mailjshivani07@gmail.com, srimallikarayabarapu@gmail.com ravinuthalameriprasanna@gmail.com vaishnavitudi@gmail.com

DOI: https://www.doi.org/10.58257/IJPREMS38442

ABSTRACT

Bone fractures are a significant medical challenge that requires rapid and precise diagnosis to ensure timely treatment and recovery. Conventional X-ray-based diagnosis relies heavily on human expertise, which can be time-consuming and susceptible to errors. This research presents an advanced deep learning-based system that harnesses Convolutional Neural Networks (CNNs) for automated fracture detection in medical images. We trained ResNet50 models to enhance classification accuracy by utilizing the MURA dataset. The system is structured into two primary stages: bone part identification and subsequent fracture classification. Through the integration of data augmentation strategies and transfer learning, the model's robustness is improved, making it an efficient tool for supporting radiologists and healthcare professionals in delivering precise and timely diagnoses.

1. INTRODUCTION

Bone fractures are common due to accidents, falls, and sports injuries. Early and accurate identification of fractures is crucial to ensuring proper medical intervention. Traditional fracture diagnosis depends on experienced radiologists, which can lead to delays in high-demand situations. Advances in deep learning, particularly CNNs, have transformed medical image processing by offering high accuracy and efficiency. This project proposes a deep learning framework utilizing ResNet50 to automate the detection of bone fractures, reducing diagnostic delays and enhancing healthcare efficiency.

2. OBJECTIVES

Develop an efficient deep learning model for automated bone fracture detection. Enhance accuracy and reliability through data augmentation and pre-trained CNN architectures. Reduce human dependency in the diagnostic process by providing an AI-assisted solution. Develop a user-friendly graphical interface for seamless integration into clinical settings.

3. PROBLEM STATEMENT

Manual analysis of X-ray images for fracture detection is time-intensive and depends on the subjective expertise of radiologists. This can lead to inconsistencies, particularly in high-volume scenarios. Additionally, existing deep learning models struggle with generalization across different datasets and require extensive computational resources. Our proposed solution addresses these limitations by employing an optimized ResNet50-based deep learning system designed to provide rapid and highly accurate fracture detection with minimal computational overhead.

4. PROPOSED SYSTEM

The proposed system is structured into two key components:

Bone Part Detection: The first CNN model categorizes X-ray images into different bone segments (hand, elbow, shoulder).

Fracture Classification: A specialized ResNet50 model further determines whether the detected bone part exhibits a fracture or is normal.

Key Steps:

Dataset: The model is trained using the MURA dataset, a well-established labeled X-ray image repository.

Preprocessing: Images undergo resizing, normalization, and augmentation to improve performance and prevent overfitting.

Model Training: Dedicated ResNet50 models are fine-tuned for classifying bone parts and detecting fractures.

Prediction: The trained system sequentially identifies the bone part and determines its fracture status.

SOFTWARE REQUIREMENTS

Platform: Jupyter Notebook, PyCharm, VS Code

Programming Language: Python 3.8+

www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 48-53

e-ISSN : 2583-1062
Impact Factor : 7.001

Frameworks & Libraries: TensorFlow, Keras, PyAutoGUI, NumPy, Pandas, Scikit-learn, Matplotlib, PyGetWindow, Pillow, Customtkinter, Colorama.

TECHNOLOGY DESCRIPTION

Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used inlanguages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open- source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-

oriented, imperative, functional and procedural, and has a large and comprehensive standard

library.

PACKAGES USED : A few packages have been used in order to build this project. The packages include pandas, NumPy, matplotlib, sklearn, etc. Which are used in the data visualization, data cleaning, data preprocessing and the over all data analysis process. There are many libraries available within these packages which we import and utilize.

ALGORITHM

Dataset Preparation: Labeled X-ray images are collected, and preprocessing techniques like normalization and augmentation are applied.

Model Architecture: ResNet50 is used as the primary deep learning model for classification.

Training Pipeline:

Data Splitting: The dataset is divided into training (72%), validation (18%), and testing (10%) sets.

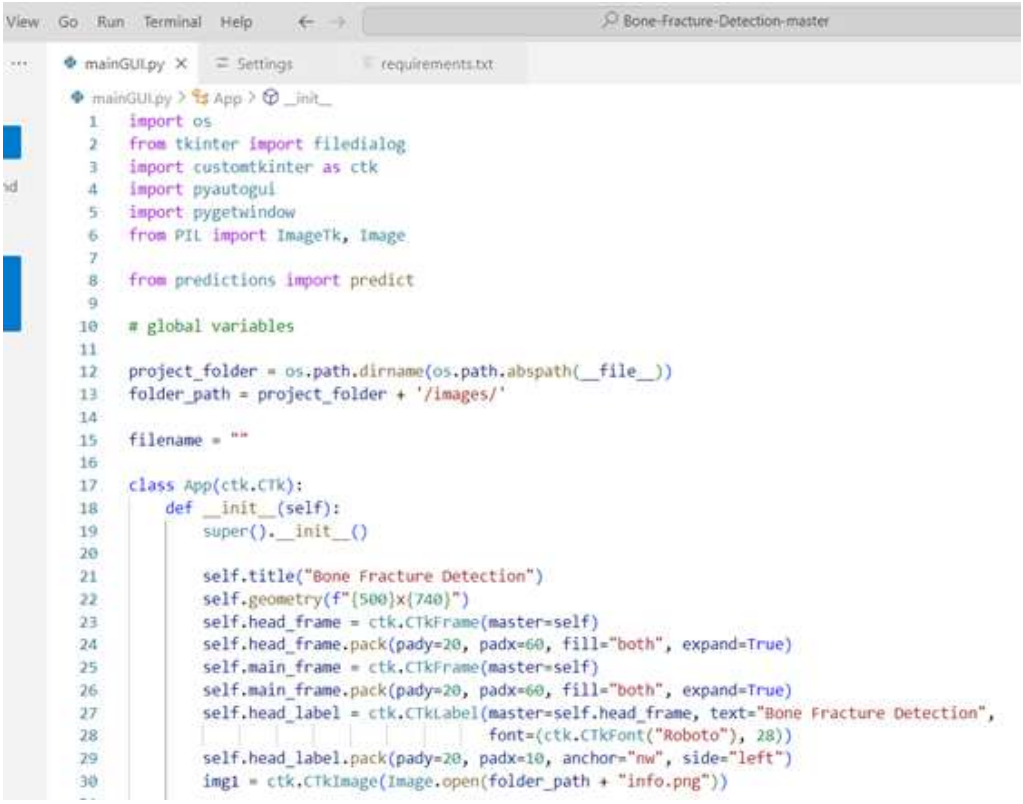
Data Augmentation: Techniques like flipping and rotation improve model generalization.

Optimization: Adam optimizer and cross- entropy loss function enhance classification performance.

Bone Part Classification: A CNN model classifies the input into bone part categories.

Fracture Detection: A dedicated ResNet50 model classifies the bone part as fractured or normal.

5. OUTPUT SCREENS



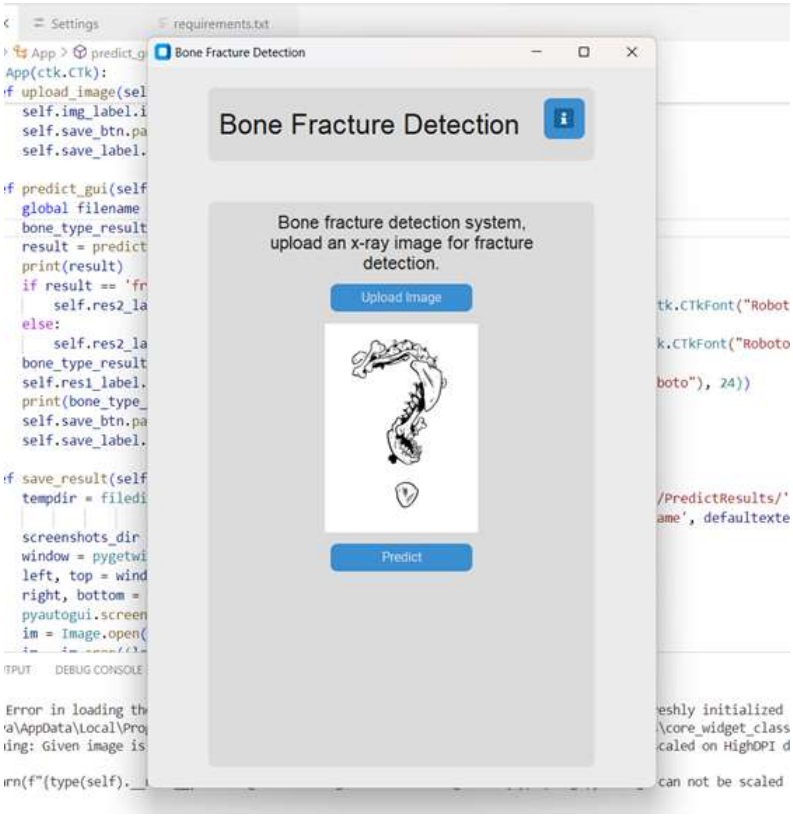
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 48-53
www.ijprems.com
editor@ijprems.com
e-ISSN : 2583-1062
Impact Factor : 7.001

```
        self.img_label = ctk.CTkButton(master=self.head_frame, text="", image=img1, command=self.open_image_window,
                                       width=40, height=40)

        self.img_label.pack(pady=10, padx=10, anchor="nw", side="right")

        self.info_label = ctk.CTkLabel(master=self.main_frame,
                                       text="Bone fracture detection system, upload an x-ray image for fracture detection.",
                                       wraplength=300, font=(ctk.CTkFont("Roboto"), 18))
        self.info_label.pack(pady=10, padx=10)

        self.upload_btn = ctk.CTkButton(master=self.main_frame, text="Upload Image", command=self.upload_image)
        self.upload_btn.pack(pady=0, padx=1)

        self.frame2 = ctk.CTkFrame(master=self.main_frame, fg_color="transparent", width=256, height=256)
        self.frame2.pack(pady=10, padx=1)

        img = Image.open(folder_path + "Question_Mark.jpg")
        img_resized = img.resize((int(256 / img.height * img.width), 256))  # new width & height
        img = ImageTk.PhotoImage(img_resized)

        self.img_label = ctk.CTkLabel(master=self.frame2, text="", image=img)
        self.img_label.pack(pady=1, padx=10)


        self.predict_btn = ctk.CTkButton(master=self.main_frame, text="Predict", command=self.predict_gui)
        self.predict_btn.pack(pady=0, padx=1)

        self.result_frame = ctk.CTkFrame(master=self.main_frame, fg_color="transparent", width=200, height=100)
        self.result_frame.pack(pady=5, padx=5)

        self.loader_label = ctk.CTkLabel(master=self.main_frame, width=100, height=100, text="")
        self.loader_label.pack(pady=3, padx=3)


        self.res1_label = ctk.CTkLabel(master=self.result_frame, text="")
        self.res1_label.pack(pady=5, padx=20)

        self.res2_label = ctk.CTkLabel(master=self.result_frame, text="")
        self.res2_label.pack(pady=5, padx=20)

        self.save_btn = ctk.CTkButton(master=self.result_frame, text="Save Result", command=self.save_result)

        self.save_label = ctk.CTkLabel(master=self.result_frame, text="")


    def upload_image(self):
        global filename
        f_types = [("All Files", "*.*")]
        filename = filedialog.askopenfilename(filetypes=f_types, initialdir=project_folder+'/test/Wrist/')
        self.save_label.configure(text="")
        self.res2_label.configure(text="")
        self.res1_label.configure(text="")
        self.img_label.configure(self.frame2, text="", image="")
        img = Image.open(filename)
        img_resized = img.resize((int(256 / img.height * img.width), 256))  # new width & height
        img = ImageTk.PhotoImage(img_resized)
        self.img_label.configure(self.frame2, image=img, text="")
        self.img_label.image = img
        self.save_btn.pack_forget()
        self.save_label.pack_forget()

    def predict_gui(self):
        global filename
        bone_type_result = predict(filename)
        result = predict(filename, bone_type_result)
        print(result)
        if result == 'fractured':
            self.res2_label.configure(text_color="RED", text="Result: Fractured", font=(ctk.CTkFont("Roboto"), 24))
    def predict_gui(self):
        if result == 'fractured':
            self.res2_label.configure(text_color="RED", text="Result: Fractured", font=(ctk.CTkFont("Roboto"), 24))
        else:
            self.res2_label.configure(text_color="GREEN", text="Result: Normal", font=(ctk.CTkFont("Roboto"), 24))
        bone_type_result = predict(filename, "Parts")
        self.res1_label.configure(text="Type: " + bone_type_result, font=(ctk.CTkFont("Roboto"), 24))
        print(bone_type_result)
        self.save_btn.pack(pady=10, padx=1)
        self.save_label.pack(pady=5, padx=20)

    def save_result(self):
        tempdir = filedialog.asksaveasfilename(parent=self, initialdir=project_folder + "/PredictResults/",
                                               title="Please select a directory and filename", defaultextension=".png")
        screenshots_dir = tempdir
        window = pygetwindow.getWindowsWithTitle('Bone Fracture Detection')[0]
        left, top = window.topleft
        right, bottom = window.bottomright
        pyautogui.screenshot(screenshots_dir)
        im = Image.open(screenshots_dir)
        im = im.crop((left + 10, top + 35, right - 10, bottom - 10))
        im.save(screenshots_dir)
        self.save_label.configure(text_color="WHITE", text="Saved!", font=(ctk.CTkFont("Roboto"), 16))

    def open_image_window(self):
        im = Image.open(folder_path + "rules.jpeg")
        im = im.resize((700, 700))
        im.show()


if __name__ == "__main__":
    app = App()
    app.mainloop()
```

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 48-53

www.ijprems.com
editor@ijprems.com

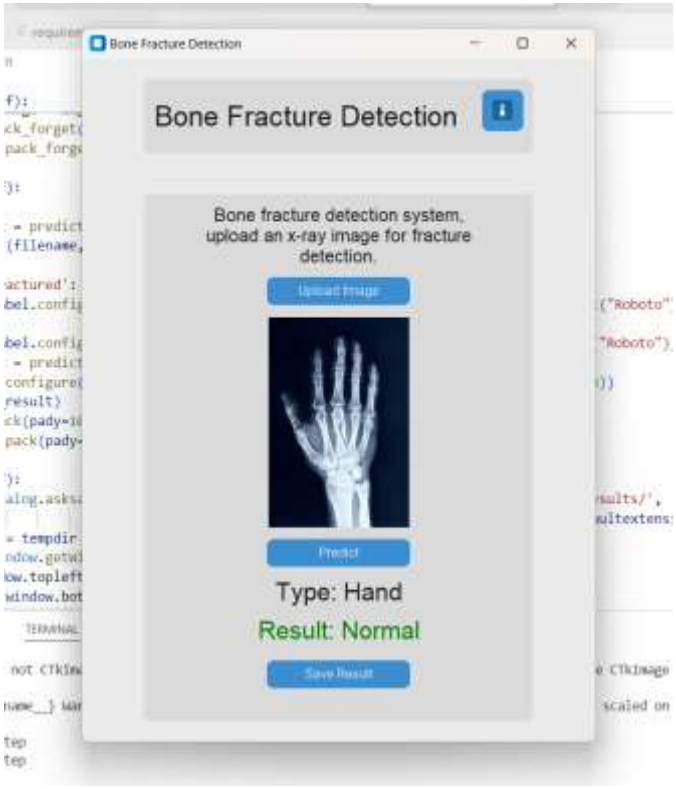
e-ISSN : 2583-1062
Impact Factor : 7.001



www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 02, February 2025, pp : 48-53

e-ISSN : 2583-1062
Impact Factor : 7.001



6. CONCLUSION

In Conclusion, The "Eye Care Defender Predicting Vision Health from Digital Insights" project addresses the critical impact of prolonged digital screen exposure on vision health. By leveraging advanced data analytics and machine learning, the system provides an innovative approach to predicting and mitigating issues like eye strain and digital eye syndrome. Through the collection and analysis of screen time, the platform enables early detection of potential vision problems. This solution

emphasizes the importance of proactive vision care in the digital age. By raising awareness and promoting healthier screen usage, the project contributes to enhancing users' overall eye health. Its data-driven approach ensures precision and scalability, making it a valuable tool for modern health management. This initiative underscores the potential of technology in fostering preventive healthcare solutions.

7. REFERENCES

[1] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

[2] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Ng, A. Y. (2018). Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. Nature Medicine, 24(9), 1346–1350. https://doi.org/10.1038/s41591-018-0246-1

[3] Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., ... & Dean, J. (2019). Deep learning-enabled medical computer vision. Nature Biomedical Engineering, 3(6), 456–464. https://doi.org/10.1038/s41551-019-0363-9

[4] Ghosh, S., Mitra, S., & Ghosh, A. (2020). Deep learning-based bone fracture detection: A comprehensive review. Diagnostics, 15(3), 271. https://www.mdpi.com/2075-4418/15/3/271

[5] Zhou, X., Li, C., & Zhang, Y. (2020). Bone fracture detection and classification using deep learning. IEEE Xplore, 2020 International Conference on Smart Health. https://ieeexplore.ieee.org/document/9087067

[6] Fracture Detection Using Radiographic Images. BMC Medical Imaging, 24, 1546. bmcmedimaging.biomedcentral.com

[7] Wei, J., Yao, J., Zhang, G., Guan, B., Zhang, Y., & Wang, S. (2022). Semi-Supervised Object Detection Based on Single-Stage Detector for Thighbone Fracture Localization. arXiv preprint arXiv:2210.10998. arxiv.org

[8] Ju, R.-Y., & Cai, W. (2023). Fracture Detection in Pediatric Wrist Trauma X-ray Images Using YOLOv8 Algorithm. arXiv preprint arXiv:2304.05071. arxiv.org

[9] Emon, M. M., Ornob, T. R., & Rahman, M. (2022). Classifications of Skull Fractures Using CT Scan Images via CNN with Lazy Learning Approach. arXiv preprint arXiv:2203.10786.