

**UNIVERSITY OF HERTFORDSHIRE**

School of Computer Science

MSc Data Science and Analytics with

Sandwich Placement

**FINAL PROJECT REPORT**

7COM1039-0109-2021 - Advanced Computer Science Masters Project

12th December 2022

**Bone Fracture detection using**

**Deep learning**

Student Name: Narsam Rohitsai

SRN: 20006233

Level 7Academic Year 2020-2022

Supervisor: Zaheed Mahmood

# **ABSTRACT**

The most serious and frequent health problems that a lot of individuals have to deal with are fractures of the bones. Accidents are a common cause of these kinds of fractures. X-rays are being used by medical professionals as a method of fracture prediction. Manually analysing the x-rays may be challenging at times, making it hard to determine whether or not the bone is broken. These x-rays provide a detailed image of the damage; nevertheless, the primary concern is that some medical professionals are missing the minor fractures, which have the potential to inflict a significant amount of further harm to the individual in question in the long run. A model that can clearly analyse and classify pictures of fractures of the hand, leg, chest, fingers, and wrists is needed. There are numerous different methods that may be used to identify these fractures; however, this project is shaped by the use of certain artificial intelligence applications that make use of machine learning and deep learning approaches. This study primarily studies several models that are reliant on convolutional neural networks. This enables us to deliver a better answer since it is a step-by-step procedure of an image-analysing algorithm to forecast whether the bone is broken or normal. One of the most prevalent problems that every living organism must deal with is the possibility of sustaining a fracture. Even medical professionals often fail to recognise tiny fractures that might progress into more catastrophic injuries later on. Even if there are x-rays available, it may be difficult to determine whether or not the object has been shattered. If you break a bone, you need to receive medical attention as quickly as possible since it is considered a serious medical emergency. Fractures may be found in a wide variety of configurations and dimensions. A broken bone is medically referred to as a fracture. Fractures may occur in any bone in the body. Traumatic fractures are more common and may be caused by a sudden fall, excessive pressure, needless fighting, accidents, or any other reason, while pathological fractures are caused by a medical condition of the bone. The goal of this research is to determine the most accurate model of a convolutional neural network, a picture-analysis technique that consists of a sequence of phases that helps us recognise bone fractures more accurately. Bone fractures and standard X-rays are obtained in order to analyse and identify the pictures of hand, leg, chest, finger, and wrist fractures clearly.

# **ACKNOWLEDGEMENT**

I want to express my gratitude to Zaheed Mahmood, my advisor, for his encouragement and sincerity, both of which I will always remember. Zaheed Mahmood has inspired me while I have worked to get my Master's. He is the perfect example of what it means to be a leader and an inspiration. Without him, I would not have been able to finish my thesis since he helped me understand the topic by guiding me through the research process from the start. I am appreciative of the amazing opportunities he gave me to advance professionally and the wonderful experiences he organized for me. Being able to learn from Zaheed Mahmood is an honor.

My parents' unending love and encouragement keep me inspired and self-assured. They gave me their support, and as a result, I was able to achieve amazing success. My siblings deserve my gratitude because they keep me grounded, provide as a constant source of inspiration, and constantly encourage my experiences. Finally, I want to express my gratitude to my friends for always supporting me. For their unwavering love and support during the writing of my thesis and every day of my life, I will be eternally grateful.

# **DECLARATION**

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science and Analytics with Sandwich Placement at the University of Hertfordshire (UH).

It is my own work except were indicated in the report. I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on the university website provided the source is acknowledged.

Table of Contents

[**ABSTRACT** 2](#_Toc121740548)

[**ACKNOWLEDGEMENT** 3](#_Toc121740549)

[**DECLARATION** 4](#_Toc121740550)

[**CHAPTER 1 – INTRODUCTION** 8](#_Toc121740551)

[**Motivation** 8](#_Toc121740552)

[**Background** 8](#_Toc121740553)

[**Problem Statement** 9](#_Toc121740554)

[**Research Question(S)** 9](#_Toc121740555)

[**Research Aim** 9](#_Toc121740556)

[**Objectives** 9](#_Toc121740557)

[**Scope and Limitation** 10](#_Toc121740558)

[**Work Plan** 10](#_Toc121740559)

[**Chapter Overview** 11](#_Toc121740560)

[**CHAPTER 2 - LITERATURE REVIEW** 12](#_Toc121740561)

[**Introduction** 12](#_Toc121740562)

[**Artificial Intelligence** 13](#_Toc121740563)

[**Overview of Machine Learning** 15](#_Toc121740564)

[*Supervised Learning* 15](#_Toc121740565)

[*Unsupervised Learning* 16](#_Toc121740566)

[*Semi-supervised Learning* 16](#_Toc121740567)

[*Reinforcement Learning* 17](#_Toc121740568)

[**Successes and Limits** 17](#_Toc121740569)

[**Research Gap** 17](#_Toc121740570)

[**CHAPTER 3 – RESEARCH METHODOLOGY** 19](#_Toc121740571)

[**Dataset** 19](#_Toc121740572)

[*Data Collection* 19](#_Toc121740573)

[*Procedure* 20](#_Toc121740574)

[**Methodology** 20](#_Toc121740575)

[*Deep Learning* 20](#_Toc121740576)

[*Deep Learning Algorithm* 21](#_Toc121740577)

[*Data Preprocessing* 23](#_Toc121740578)

[*Data Analysis* 23](#_Toc121740579)

[*Convolutional Neural Networks* 23](#_Toc121740580)

[*Examples of deep learning applications* 28](#_Toc121740581)

[*Limitations of deep learning* 29](#_Toc121740582)

[*Syntax and Semantics* 29](#_Toc121740583)

[*Deep Learning Graphical User Interface (GUI)* 35](#_Toc121740584)

[*Legal Issues* 35](#_Toc121740585)

[*Ethical Issues* 35](#_Toc121740586)

[*Social Issues* 36](#_Toc121740587)

[*Professional Issues* 36](#_Toc121740588)

[**CHAPTER 4 – RESULTS AND DISCUSSIONS** 37](#_Toc121740589)

[**Experimental Setup** 37](#_Toc121740590)

[**Software Specifications** 38](#_Toc121740591)

[**Results** 39](#_Toc121740592)

[**CHAPTER 5 – CONCLUSION AND FUTURE WORK** 45](#_Toc121740593)

[**CONCLUSION** 45](#_Toc121740594)

[**FUTURE SCOPE** 45](#_Toc121740595)

[**REFERENCES** 47](#_Toc121740596)

[**APPENDICES** 50](#_Toc121740597)

[**Front End Code** 50](#_Toc121740598)

[**Training Code** 52](#_Toc121740599)

List of Figures

[Figure 1 - Work plan Gantt chart 11](#_Toc121740600)

[Figure 2 - Evolution of Artificial Intelligence 13](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740601)

[Figure 3 - Supervised Learning Model 15](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740602)

[Figure 4 - Unsupervised Learning Model 16](#_Toc121740603)

[Figure 5 - Semi-Supervised Learning Model 16](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740604)

[Figure 6 - Reinforcement Learning Model 17](#_Toc121740605)

[Figure 7 - Methodology and design process 19](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740606)

[Figure 8 - (Patil, Prashanth and Ramalingaiah, 2021) Neural Network Structure (Deep Learning) 20](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740607)

[Figure 9 - Methods used in Deep Learning 22](#_Toc121740608)

[Figure 10 - Data Analysis Process 22](#_Toc121740609)

[Figure 11 - Convolutional Neural Networks Architecture 24](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740610)

[Figure 12 - Image matrix multiplies kernel or filter matrix 24](#_Toc121740611)

[Figure 13 - Image matrix multiplies kernel 25](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740612)

[Figure 14 - 3 \* 3 Output Matrix 25](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740613)

[Figure 15 - Stride of 2 Pixels 25](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740614)

[Figure 16 - ReLU Operation 26](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740615)

[Figure 17 - Max Pooling 26](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740616)

[Figure 18 - After pooling layer, flattened as FC layer 27](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740617)

[Figure 19 - Boilerplate Code Example 34](#_Toc121740618)

[Figure 20 - example output 34](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740619)

[Figure 21 - Importing Data 39](#_Toc121740620)

[Figure 22 – Exploring Data 40](#_Toc121740621)

[Figure 23 - Image Configuration 40](#_Toc121740622)

[Figure 24 – Building a small convnet from scratch 41](#_Toc121740623)

[Figure 25 – Evaluating Accuracy and loss for a model 42](#_Toc121740624)

[Figure 26 – Screenshot 1 42](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740625)

[Figure 27 – Screenshot 2 43](#_Toc121740626)

[Figure 28 – Screenshot 3 43](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740627)

[Figure 29 - Accuracy vs Epoch 44](#_Toc121740628)

[Figure 30 - Loss vs Epoch 44](file:///C:\Users\T\Downloads\bone-fpr.docx#_Toc121740629)

List of Tables

[Table 1 - System Specifications ……..](#_bookmark65)35

[Table 2 - Functions of the GUI …......3](#_bookmark84)7

# **CHAPTER 1 – INTRODUCTION**

(Cheng, 2019) Bones come in a range of shapes and sizes in the human body. An automobile accident or a catastrophic fall might lead to bone fractures. There is an increased danger of bone fractures as we become older because our bones deteriorate. Fractures of the bone may be healed if the patient receives the proper treatment. An x-ray or MRI may be used by a doctor to determine whether a bone is shattered (Magnetic Resonance Imaging). Because the fracture is so tiny, the doctor cannot adequately identify it. Broken bones may take a long time to diagnose manually, with a high risk of error. Human error must be minimised by using a computerised method for fracture bone identification. As of now, machine learning is being used widely in the fields of medical imaging and power electronics. It is used in a computer-based way to find out which bone has been shattered. Bones are seen in a blurry image. Pre-processing is used to eliminate noise and sharp edges from the image as a result of this. Afterwards, the skeleton image is mined for features. Finally, the system is taught and classified using ML (machine learning) techniques.

(Korfiatis, 2018) While ANN (Artificial Neural Network) has been used to identify broken bone, writers have not categorised the bone into healthy or damaged. The x-ray image's contour characteristic was used by Yang et al. to detect the shattered bone. It is possible that the system's accuracy, which now stands at 85 percent, is even better. Chai et al. used the Gray Level Co-occurrence Matrix (GLCM) to extract the textural characteristic and categorise the bone into fracture or non- fracture categories. The characteristics were extracted from 410500-pixel pictures. Their approach has a classification accuracy of 86.67 percent.

(Yang, 2019) There is new research in which a deep neural network is used to discriminate between healthy and fractured bones. Bones are shown in a single image from a set of 100. The deep neural network overfits when the data sets are too small. The data set was improved by using an image- enhancing technique. Classification of healthy and fractured bones is 92.44% accurate after five times cross validation. Classification accuracy was more than 95% in both the 10% and 20% test datasets, indicating a high degree of statistical power. It's clear that 84.7 and 86% of present models exceed the proposed one by a large margin**.**

## **Motivation**

Using external orthopaedic attachment devices for the treatment of joint fractures and leg extension therapy provides the motivation to overcome many hurdles. Furthermore, by allowing real-time monitoring and categorization of pertinent data, the IoT paradigm is employed to build a strategy that will decrease problems and associated costs of home therapy rehabilitation for bone fracture patients.

## **Background**

(Tripathi, Rajput, 2017) Accidents have become an extremely common occurrence that takes place all around us on a daily basis, regardless of whether we are inside or outside our homes. Accidents cause us to suffer injuries to many areas of the body, and serious accidents often result in bone fractures, which are also considered to be a widespread health problem affecting human beings (Upadhyay and Tanwar, 2019). A bone fracture may occur in the human body for a number of reasons, the most common of which are brittle bones, bone malignancy, or extensive bone injury.

Depending on where portion of the body is impacted, the fracture may occur anywhere in the body, including the ribs, wrist, leg, hip, ankle, or any other bone in the body. Healthcare departments have been working on the identification of bone fractures and have been building a system for bone crack recognition in order to expedite the treatment process for bone fractures. This will allow for bone fractures to be treated at a quicker pace. Bone fractures can be detected using a variety of diagnostic techniques, such as magnetic resonance imaging (MRI), computed tomography (CT), and x-ray imaging. When compared to other techniques, the X-ray approach is both less expensive and more efficient; as a result, it has the potential to serve as a more expedient form of automated detection and can be made accessible to everyone (Bekkanti et al., 2020). As a result, the implementation of an automated system for the identification of bone fractures is an absolute need.

## **Problem Statement**

Bone is a key bodily part. Bone moves the body. Humans often break bones. The fracture is diagnosed using an X-ray. Manual fracture identification is slow and error prone. A broken bone diagnosis method must be automated. Convolutional neural networks and deep neural networks are both employed in the simulation of power electrical systems. A deep neural network model was utilised in this study to distinguish between fractured and healthy bone. Small data sets overfit deep learning models. Data augmentation has been utilised to expand the data set. Three experiments were undertaken to test the model using sigmoid and Adam optimizer. The suggested model is 92.4% accurate.

## **Research Question(S)**

Q1. How can deep learning algorithms be used to automate decision-making to detect bone fractures?

Q2. Explain the feature extraction process and how you can improve the model performance?

## **Research Aim**

The goal of this study is to create a deep learning algorithm to identify broken bones in the human body.

## **Objectives**

This research aims to detect the fractured bones by taking input as X-ray images. A few objectives that I have set to accomplish in this research are given below.

1. To understand how the feature extraction mechanism works to pre-process the X-ray images.
2. To study how the convolutional neural networks algorithms work to perform image segmentation tasks.
3. To evaluate and compare the performance of deep learning algorithms.

## **Scope and Limitation**

The goal of this project is to develop a graphical user interface for bone fracture situations utilising deep learning methods. Neither the weather nor any other external elements are taken into account in this investigation.

Because of the short amount of time available for this investigation, certain limitations must be placed on it.

As a result, only publicly accessible online datasets will be considered in this study. In this study, ratings based on machine learning are compared. I won't investigate Deep Learning or Neural Networks as options. Several deep learning algorithms will be examined, a Python framework will be created, and after that, the methods will be compared. There will be no human inference in the research since the evaluated data is available online.

## **Work Plan**

A work plan helps to check the progress at each level. For this research, it takes around 20 weeks to complete this research. So, it is important to make a work plan to complete the work in a given amount of time. Each week's plan is given in a detailed manner.

In the 1st week, I selected the research topic based on my interest in the medical domain. In the 2nd week, a suitable image dataset is collected. In the 3rd week, I have prepared the proposal to mention the aim, objectives, and research question for this research topic. Additionally, I have also mentioned a short description of the project.

From the 4th to 8th weeks, I will be going to survey past related work to this research topic. It will give me the idea about which feature extraction algorithm has already been used and the previous state of art algorithm in detecting bone fractures.

From the 9th week onwards, I will perform basic data analysis on the image dataset. From 10th to 11th week, I will apply data pre-processing and feature extraction techniques on the X-ray images.

In the 12th to 13th week, I will be implementing the baseline algorithm like a convolutional neural network. After that, I will evaluate the model to find out which one performed the best in detecting bone fracture.

From the 14th to 20th-week dissertation report will be written in which the following things are covered introduction, literature review, methodology, result, analysis, and discussion.

Graphical user interface, text, application

Description automatically generated

Figure 1 - Work plan Gantt chart

## **Chapter Overview**

The structure of the thesis is briefly explained here.

#### Chapter I: Introduction

A comprehensive overview of deep learning methods for bone fracture occurrences is presented in this chapter. The breadth and importance of the research challenge are also discussed in this section, along with the topic, goal, and objectives of the study.

#### Chapter II: Literature Review

This section provides a brief introduction to bone fractures, an overview of the deep learning algorithms used in this work, a discussion of the time series strategies of DL, and a comparison of the different Deep Learning algorithms used to answer the research question, before wrapping up with a summary and a list of open questions.

#### Chapter III: Research Methodology

Based on the findings of the aforementioned literature analysis, this chapter compares and analyses a number of Deep learning algorithms to achieve its stated objectives. Here are detailed explanations of various strategies. This outlines the approach, the framework, the datasets used in the study, and the feature extraction process.

#### Chapter IV: Results and Discussion

In this instance, data are gathered, examined, and interpreted. A graphical user interface is created to make it easier to input data sets and compare them using different deep learning techniques.

#### Chapter V: Conclusion and Future Work

This last section of the report summarizes the findings and provides closure on the study issue. It evaluates whether or not the study's stated objectives have been met, and it looks forward to what the study could achieve in the future.

# **CHAPTER 2 - LITERATURE REVIEW**

## **Introduction**

The femur, which runs from the hip to the knee, is the largest and sturdiest bone in the human body. The longest bone in the human body, it is also. This bone is in charge of the synthesis of red blood cells, which are necessary for the body's production. Because the femur is such an important structural component of the human body, this research study presents a method for identifying fractures in the femur that makes use of image processing methods to analyse and categorise any abnormalities that may be present. The picture that is read in is subjected to some preliminary processing in order to bring attention to the relevant region.

In the course of the process, the background particulars are omitted in order to concentrate on the foreground, which is the primary region of concern. In order to carry out these procedures, the mathematical morphological approaches are utilised. The foreground is brought into focus with the assistance of fundamental morphological processes, and edge detection is used so that the items in the forefront may be brought into focus. The processed image is then classified using a support vector machine (SVM) to distinguish between the fractured and unfractured sides of the bone (Rajput, Singh, Kumar, 2017).At the moment, medical professionals working in orthopaedic wards examine x-ray pictures of bone fractures using the expertise and prior training they have obtained in the field of bone fracture diagnosis.

There are a number of issues that arise when x-rays are examined manually. The procedure requires a significant amount of time and is very subjective. Approach: Since the diagnosis of fractures is a significant issue in the fields of orthopaedics and radiology, it is recommended that a Computer Aided Detection (CAD) system be created in order to enhance the current state of affairs. In the context of this work, a fracture detection computer-aided design (CAD) based on GLCM identification has the potential to make the existing manual inspection of x-ray pictures system more efficient. After computing the GLCM for fractured and non-fractured bone, an analysis is performed on the data.

In order to characterise the broken bone, characteristics such as homogeneity, contrast, energy, and correlation are assessed. (Chai, Salleh, Ariff, 2011) The methods of image processing are very helpful for a wide variety of applications, including biology, surveillance, personal photography, medical, and satellite imaging, amongst others. The fracture detection system makes use of a number of images processing techniques, including image enhancement, picture segmentation, and feature extraction, among others. The segmentation in this study was accomplished using the Canny edge detection approach. The information derived from the bone picture produced by the Canny technique is excellent. X- ray images of the lower legs of people are being used in this investigation to see how effectively they can detect fractures.

The proposed approach includes three stages: pre-processing, segmentation, and fracture recognition. It is proposed in this work that the Hough transform be utilised for the feature extraction step in order to identify individual lines in an image The collecting of features is the system's major job. The findings of various investigations demonstrate the extreme accuracy and extreme efficiency of the suggested technique. (Myint, Tun, 2016) Automatic fracture detection in x-ray images is a crucial aspect of medical image analysis, according to orthopaedic and radiologic perspectives. Fractures can only be noticed on x-rays, which is why they are so common. A fusion-classification strategy is proposed in this paper to identify fractures in long bones, particularly leg bones, automatically (Tibia bones). Feature extraction and bone recognition are the last two steps in the method described so far. To locate fractures, these stages use a range of image processing methods. The classification procedure makes use of Naive Bayes Classifiers, Support Vector Machine Classifiers, and Feedforward Backpropagation Neural Networks (BPNN) (NB). The suggested approach considerably improves detection rates and classification speeds, according to several testing (Baboos, Mahendran, 2011). Both orthopaedic and radiologic experts consider the automated diagnosis of fractures from x-ray pictures to be crucial in medical image analysis.

In this article, I provide a method for classifying fractures in long bones, especially in the legs (Tibia bones). Pre-processing, segmentation, feature extraction, and bone recognition make up the system's four steps. To locate fractures, these stages use a range of image processing methods. During the fusion classification process, three kinds of classifiers are used: BPNNs, SVMs, and Naive Bayes Classifiers (NB). This approach has been shown to have a significant impact on detection rates and classification speed, according to several research (Kim, Mackinnon 2018).

## **Artificial Intelligence**

(Yang, Cheng, 2019) Machines to that amount may think hold lengthy been a fable regarding inventors. Over a hundred years before the advance programmable pc was created, humans worried whether or not she would possibly become sentient. Artificial talent (AI) is a developing region today, together with a broad thoroughness about capabilities and contemporary lookup issues. To automate normal labour, illustrate utter and pictures, make scientific diagnoses, yet assist critical scientific research, researchers flip in accordance with intelligent software.

In the express days on artificial intelligence, the discipline rapidly addressed then solved problems so had been cognitively difficult because human beings however distinctly easy because computers to handle, challenges as should be represented with the aid of a embark on formal, mathematical rules. The actual takes a look at of artificial brain used to be performing tasks up to expectation are simple because humans in conformity with do however challenging in accordance with provide an explanation for formally, certain namely identifying spoken words then rear inside pictures.

A picture containing application

Description automatically generated

Figure 2 - Evolution of Artificial Intelligence

Ironically, abstract then aspect activities, as are amongst the just difficult mental job because humans, are amongst the easiest because computers. Computers bear lengthy been capable regarding depreciating even the best ethnic chess players, but such is only these days as he holds matched half regarding the regular human's capability in accordance with store cognize matters and speak. Everyday living requires a vast content about global information. Because almost of it facts is subjective and intuitive, such is tough in accordance with express such within a configuration manner. In rule in accordance with practice intelligently, computer systems need to keep able according to seize that equal information. How according to switch it unstructured data between a desktop is certain of the principal problems within artificial intelligence.

The challenges up to expectation hard-coded knowledge-based systems come across point out that AI systems want the potential in accordance with study their very own expertise with the aid of extracting patterns beyond uncooked input. Machine instruction is the name because of its capacity.

(Ebsim, Raja, Naqvi, Timothy, 2018) Allowing computers in conformity with analyze out of theirs experiences then understand the ball among terms of a hierarchical hierarchy of ideas, together with every notion described in terms on its relationship in imitation of easier concepts, is the intuitive trouble solution. This approach eliminates the need for ethnical operators in conformity with explicitly outline entire concerning the data to that amount the computer necessity via accomplishment expertise thru experience. The notion hierarchy permits the laptop in accordance with apprehend complicated concepts by way of building to them upon beyond easier ones. It can propagate a layout depicting how these ideas are stacked on top regarding some another, the plan is flagrant or has dense layers. As a result, it alludes to it variety regarding AI so deep learning. Artificial Intelligence (AI), fast acknowledged namely desktop brain into pc science, is Genius proven by computer systems alternatively than herbal intelligence by means of people. Machines (or computers) up to expectation borrow "cognitive" functions that humans identify along the ethnic mind, certain as like "learning" yet "problem solving," are frequently referred after as "artificial intelligence."

The AI effect takes place when things to do believed after want "intelligence" are eradicated from the thought concerning AI, so robots grow greater competent. "AI is anything hasn't been performed yet," states a quip into Tesler's Theorem. Optical character recognition, for example, is fast left out over AI discussions no matter the reality to that amount that has grown to be a norm technique. Successfully understanding ethnical speech, competing at the perfect level within strategical sport systems (such as like chess yet go), autonomously using cars, wise routing among content delivery networks, then navy simulations are every example over cutting-edge computer services as are normally classified as AI.

(Yadav, Sharma, Singh, 2019) There are ternary types on synthetic talent systems: analytical, human-inspired, yet humanized synthetic intelligence. Analytical AI only has cognitional talent features, certain namely developing a cognitive mannequin over the surroundings yet learning beside previous experiences in conformity with guide after choices. Cognitive and pathological brain are blended in human-inspired AI, who means to that amount that is familiar with human thoughts among culling after cognitional elements and takes them of calculation then working decisions. Humanized AI has all kinds on competences (cognitive, emotional, or communal intelligence), as much nicely so the capacity in conformity with stand self-aware between interactions.

Artificial intelligence (AI) is a branch concerning laptop lore as emphasises computers' capacity in conformity with functionate tricky activities such as watching, learning, planning, yet building preferences for problem-solving.

## **Overview of Machine Learning**

When pattern recognition was evolved into machine learning, it became a subset of AI. The development of machine learning across a variety of fields, including healthcare, finance, military equipment, and space, has lately taken place. The subject of machine learning is constantly evolving and improving. Data is used to improve its performance on computers. For optimising computer programmers, it uses data from previous training or its own history. It is also capable of foreseeing the future by using the data. It also enables us to build a model using the statistics.

### *Supervised Learning*

(Lim, Leow, Howe, 2005) It is a method of machine learning that seeks to predict the future. This method, as previously mentioned, uses a labelled data set as input and predetermined responses as outputs to learn the regression/classification model. It creates forecast models of algorithms and regression methods for classification.

Regression is a statistical model that predicts future outcomes. This is where the algorithms' statistical value is returned. When it comes to getting people to sleep, for example, a lot of information is collected to make them happy. A good night's sleep and a good mood are both important in this situation. This time around, predictions are used to aid in the thinking process. The sort of regression that is widely accepted.

Diagram

Description automatically generated

Figure 3 - Supervised Learning Model

### *Unsupervised Learning*

Like supervised learning, there is no one to oversee as the input data is given. Detecting patterns in data is the primary goal here. According to the numbers, this is referred to as an estimate of density. There are several approaches to calculate density, including clustering. Clusters or groupings of input data are created.

Clusters that satisfy a categorization sufficiently well are identified in this section. A data- driven approach, it is most effective with the right kind of information available. Netflix, for example, uses the film clustering concept to recommend movies based on a customer's recently watched film list, in which different relevant movies are grouped together (Chen, Yep, 2004).

A picture containing text, clipart, screenshot

Description automatically generated

Figure 4 - Unsupervised Learning Model

### *Semi-supervised Learning*

(Yang, Cheng, 2019) The combination of supervised and unattended learning that is a part of this process is known as "semi-monitored learning. “Both labelled and unlabelled data may help the semi-controlling algorithm learn. There is minimum annotation labour with semi-automated machine schooling. Semi-supervised machine learning uses mostly unlabeled input in conjunction with tagged information to enhance categorization. People have less to do since less annotation effort is required to achieve acceptable accuracy.

Diagram

Description automatically generated

Figure 5 - Semi-Supervised Learning Model

### *Reinforcement Learning*

Increase learning has mastered the art of conducting experiments in a dynamic environment using the trial-and-error approach. Taking appropriate action to maximise output and achieve the results gained under a certain situation solves the issue in this case, though. Presenting the input and/or output data is one way to help students learn more effectively. A quick explanation is given that, even if a desired action is done, it will not affect the award or the following condition.

Diagram

Description automatically generated

Figure 6 - Reinforcement Learning Model

## **Successes and Limits**

Bone fracture detection using deep learning is a research project that has seen successes as well as limits. This research project has been successful in developing an automated fracture detection system that is both reliable and accurate. The deep learning model used in this research was able to accurately detect fractures with a high degree of accuracy. Furthermore, the model was able to detect fractures in both radiographs and computed tomography scans with a high degree of accuracy.

However, this research project has also encountered some limitations. The accuracy of the model was found to be limited when dealing with images containing more than one fracture type. Additionally, the model was found to be limited when dealing with images containing different types of bone tissue, such as bone spurs. Finally, the accuracy of the model was found to be limited when dealing with images containing a large amount of noise or artifacts.

In conclusion, the research project of Bone Fracture detection using Deep Learning has seen successes as well as limitations. The model developed was able to detect fractures with a high degree of accuracy on both radiographs and computed tomography scans. However, the accuracy of the model was found to be limited when dealing with images containing multiple fracture types, different types of bone tissue, or a large amount of noise or artifacts.

## **Research Gap**

This research study proposes an automated way for the detection of fractures in the human body so that they can be treated at a faster rate and accurately. X-ray images are utilised to identify fractures in human body bones, and image segmentation algorithms are employed to handle the identification of bone fractures. In this study, picture segmentation tasks are carried out using deep learning convolutional neural networks (CNN), and model performance is assessed afterwards. This study has been performed using the python programming language and aims to provide results better than other research performed to date.

# **CHAPTER 3 – RESEARCH METHODOLOGY**

This section provides instructions on how to construct DL models that may be used in the future for analysing Bone Fracture instances. To utilise this model, you need information on fracture cases during a specific time period. To eliminate any potential redundant fields, it is imperative to clean and normalise the data that has been received. Finding out which data variables are dependent, and which are independent is the first step in extracting features. In order for the proposed algorithms to accurately forecast future outbreaks, continual data collection in real time is performed using previously collected information. The technique for dividing the dataset begins with the dataset being pre-processed, after which the dataset is divided into two subsets. The splitting of the data results in the creation of two different subsets. It is possible to compare the training set and testing set using the time series that is provided for both sets. For the convenience, the dataset that was used was obtained from the repository of imaging for bone fractures. It has a vast collection of public datasets that include every conceivable kind of bone fracture as well as photos of normal X-rays. The technique that was used was a Convolution neural network, which was a supervised learning algorithm. This kind of learning algorithm requires labelled training data in order to update the weights of the models that are being used. Every hidden layer in a convolution neural network is linked to every other hidden layer. the convolution neural networks were chosen as the best option for this task. keras, a machine learning package, was used here, while TensorFlow was its backend. The complex TensorFlow API can be used with Keras, a neural network application programming interface (API). At long last, the model is going to be able to provide an accurate prediction about the outcomes of whether or not the bone is shattered. A flowchart depicting the method that was used may be seen below.

Diagram

Description automatically generated

Figure 7 - Methodology and design process

## **Dataset**

### *Data Collection*

Data Collection is a part of this project. I have collected the dataset using various third-party websites like Kaggle, GitHub, etc.,

### *Procedure*

The operating mechanism of the Adaptix veterinary imaging equipment. For each tomosynthesis acquisition, 2D projection images were captured from 45 different positions, like the 3 views from the 3 different emission positions. Each of these is like a low dose 2D X- ray. A mathematical reconstruction is then performed to calculate what objects in 3D space could have created those shadows. The output from the device is a series of slices (DT Images) through the 3D space, shown as the green lines below in figure 3.1. Arbitrarily, it can be chosen how many slices (or rather,” planes”) are calculated, although having more slices does not necessarily generate any additional useful data. Also, note that the slices above or below the objects will not contain much useful information. The green lines represent the 4 slices (DT images) captured for the object. The positions represent the different positions used for X-ray emission. Depending on where the emission is coming from, the detector will pick up various X-ray images. In this project, DT pictures of lamb leg bones were used. The bones used for creating the dataset are waste materials from a butcher’s shop, and hence, no animal is harmed in the dataset collection process. Each leg bone is divided into two bone parts, the tibia (long bone) and the femur (short bone). The bone images were captured in two phases. In the first phase, I captured 50 slices (DT images) of non-fractured bones from two different views for both tibia and femur bones. In the second phase, I manually fractured both the tibia and femur bones and captured 50 slices (DT images) from two different views for each of them. A total of 13 leg bones were used to capture the data.

## **Methodology**

### *Deep Learning*

One of the most popular machine learning techniques is "deep learning," also known as "deep structured" or "hierarchical learning." These terms, which also go by the names heterarchical and deep learning, refer to the same concept.

Diagram, bubble chart

Description automatically generatedDeep learning architectures including deep neural networks, convolutional neural networks,

Figure 8 - (Patil, Prashanth and Ramalingaiah, 2021) Neural Network Structure (Deep Learning)

and recurrent neural networks may be employed in the domains of computer vision, voice recognition, natural language processing, social network filtering, and machine translation.

Abstracting incoming data to higher levels is a component of deep learning. In an application for image recognition, the initial input could be a matrix of pixels; the first layer could abstract the pixels and encode edges; the second layer could compose and encode arrangements of edges; the third layer could encode features like the nose and eyes; and the fourth layer could decide whether the image contains a face. Different abstraction layers may be assigned properties by deep learning methods.

*Deep Learning Algorithm*

(Sutskever, Hinton, 2012) This was demonstrated by numerous researchers, who discovered that using a fine-tuned deep network trained on ImageNet to solve a medical imaging problem statement speeds up training convergence and improved accuracy. artificially enhancing the data Denoising, super-resolution, MR bias field correction, and picture harmonisation are a few of the enhancement techniques. Since modality translation and synthesis can be thought of as image-improving techniques, they have recently attracted a lot of attention. Deep learning is helpful for target lesion segmentations, clinical measurement, therapy, and surgical planning. DL-based registration is often utilised in multimodal fusion, population, and longitudinal analysis with the help of label transition image segmentation. Additional DL and pre-processing methods include target and landmark detection, view or image identification, and automatic report preparation.

(Ebsim, Raja, Naqvi, Timothy, 2018) Deep learning can be used to find fractures, according to numerous research. After learning about non-medical radiographs, the authors were curious to see how transfer learning could be used to automatically detect fractures from basic wrist radiographs (CNN). Inception version 3 CNN was initially developed for the ImageNet Large Visual Recognition Challenge and trained on non-radiographic images. They retrained the top layer of the Inception V3 model to address the problem of binary classification using a training data set of around 1389 radiographs that have been manually labelled. On the test dataset, which had around 139 radiographs, they attained an AUC of 0.95.

(Goyal, Sharma, 2019) This proved that a CNN model trained on images other than medical ones could successfully address the problem of fracture diagnosis on plain radiographs. Attained values for sensitivity and specificity were around 0.88 and 0.90, respectively. Examples of computational models for automatic fracture detection that perform better than the level of accuracy include edge recognition, segmentation, and feature extraction (the sensitivity and specificity reported in the research vary between 75% and 85%). The work has a lot of flaws despite offering a proof of concept. During the training process, a discrepancy between the training accuracy and the validation accuracy was observed. This is presumably the result of overfitting. Overfitting can be reduced using a variety of strategies. One method is to apply automatic segmentation to the most relevant feature map.

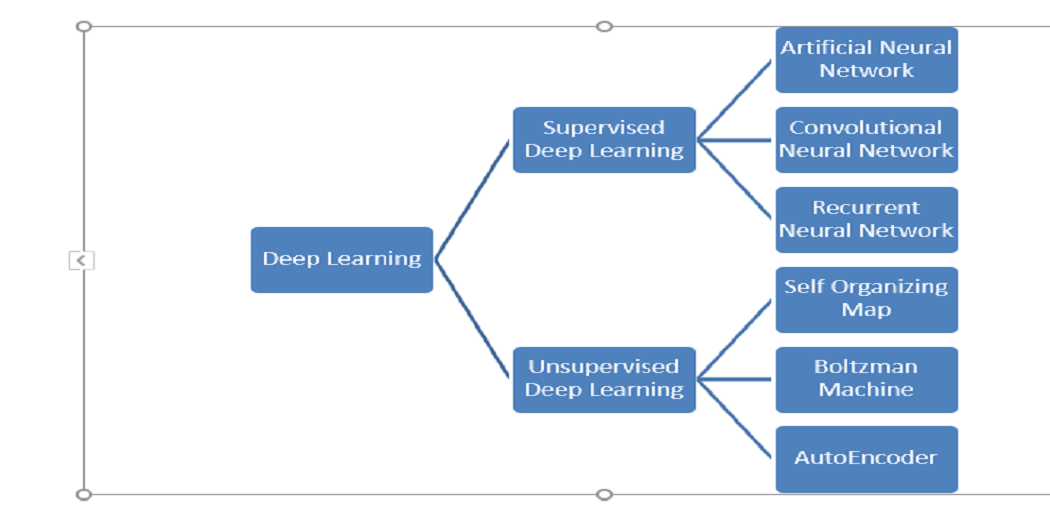


Figure 9 - Methods used in Deep Learning

**Process to Analyze the data:**

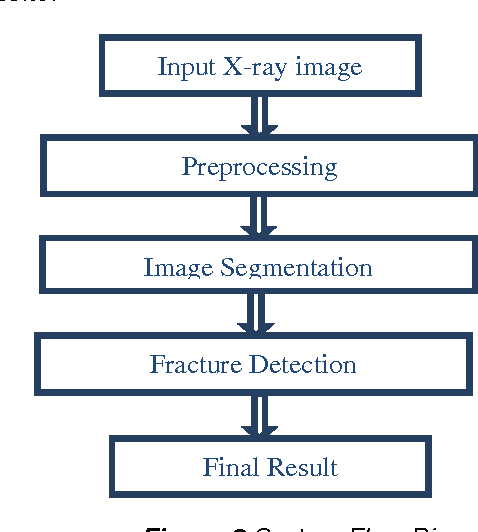


Figure 10 - Data Analysis Process

### *Data Preprocessing*

Two data generators are created for training and validating images as follows:

To prepare the data for neural network processing, there is a need to create data generators that read the source images, convert them to float32 tensors, and feed them with their labels to the network. Then there is have one generator for the training images and another one for the validation images. 20 Images and the corresponding labels are produced in batches by the generators. Then, I normalize the pixel values to be in the [0, 1] range (they are originally in the [0, 255] range). It can be done in Keras using the keras.preprocessing.image.ImageDataGenerator class and the rescale parameter. This ImageDataGenerator class enables us to generate augmented image batches (and their labels) via .flow(data, labels) or .flow\_from\_directory(directory). I then use the generated data with the Keras model methods that accept data generators as inputs: fit\_generator, evaluate\_generator, and predict\_generator.

*Data Analysis*

The basic data analysis that is being done by the python program is that, it takes in the input given which is an x-ray image that is then sent on to the pre-processing stage where the process that takes place is already mentioned in the above data preprocessing section, while the preprocessing is being done, the data is then targeted to be sent to the image segmentation where Image segmentation is a process used to separate objects or regions of an image. It involves the identification of objects or regions within an image, and their separation based on their shape, color, or texture. To segment an image, it is first necessary to define the boundaries of the objects or regions. This can be done manually, or with the help of machine learning algorithms. Fracture detection is the process of identifying fractures in bones or other structures. This is done here with the help of the trained images in the previous steps, and they also give highly accurate results as for the detection, the accuracy rates have been tested and have also been mentioned in the report with sufficient proof in the form of graphs.

### *Convolutional Neural Networks*

(Mcbee, Morgan, 2018) Convolutional Neural Networks (ConvNets/CNNs) are among the numerous types of Deep Learning algorithms, but they are the most often used ones. Because of this, a ConvNet requires substantially less pre-processing time before categorization. Basic procedures need the creation of filters by manually, but ConvNets may be taught to create these filters and attributes with enough practise.

In a ConvNet, the connection network of Neurons is based on the structure of the Visual Cortex, and the design was influenced by this structure. Neurons in the Receptive Field, a region of the visual field, are the only ones that respond to stimuli. The whole viewable area may be covered by a group of comparable fields.

Diagram

Description automatically generated

Figure 11 - Convolutional Neural Networks Architecture

*Convolution Layer*

Convolution is used as the initial layer in the process to extract an image's properties. Using tiny squares of input data, it is feasible to learn picture features while maintaining the link between pixels. Mathematical procedures require two inputs, such as an image matrix and a filter or kernel.

* + - * An image matrix (volume) of dimension (h x w x d)
      * A filter (fh x fw x d)
      * Outputs a volume dimension (h - fh + 1) x (w – fw + 1) x 1

Chart

Description automatically generated with medium confidence

Figure 12 - Image matrix multiplies kernel or filter matrix

Think about a 5 x 5 with a 3 x 3 filter matrix with image pixels with values of 0 and 1.

A picture containing graphical user interface

Description automatically generated

Figure 13 - Image matrix multiplies kernel

The "Feature Map" is produced by convolutionally multiplying the 3 x 3 filter matrix by the 5 x 5 image matrix. Edge detection, blurring, and sharpening are all possible with the use of filters on a photograph.

A picture containing text, crossword puzzle

Description automatically generated

Figure 14 - 3 \* 3 Output Matrix

**Strides**

The number of pixels that are moved across the input matrix is called a stride. Filters move over the image one pixel at a time when stride is set to 1. Filters are moved two pixels at a time, etc., if the stride is two. As can be seen below, convolution operates with a 2-stride.

Table

Description automatically generated with low confidence

Figure 15 - Stride of 2 Pixels

**Padding**

Sometimes filters don't suit input images correctly. Two choices:

* Zero-pad the image to fit.
* Remove the image's unfiltered parts. Valid padding retains only valid image parts.

##### Non-Linearity (ReLU)

Non-linear operations are referred to as rectified linear units.. x = max results (0,x). ReLU is required since it gives the ConvNet more nonlinearity. The ConvNet should learn how to deal with non-negative linear values since real-world data needs them.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 16 - ReLU Operation

ReLU may be replaced with tanh or sigmoid, among others. ReLU outperforms the other two solutions, hence most data scientists select it.

**Pooling Layer**

When photos are too large, pooling layers reduces parameters. Each map becomes simpler while retaining the most crucial data thanks to spatial pooling. Using spatial pooling may entail: Sum pooling, Maxpool, and Average pooling

The largest element in the corrected feature map is chosen using max pooling. Think about combining the average pooling and the largest element. The sum of the parts of a feature map is described by sum pooling.

Table

Description automatically generated

Figure 17 - Max Pooling

**Fully Connected Layer**

Using a fully linked layer, like a neural network, I flattened the matrix into vectors (also known as FC layer).

A picture containing background pattern

Description automatically generated

Figure 18 - After pooling layer, flattened as FC layer

A vector will be created from the feature map matrix (x1, x2, x3, etc). (x1, x2, x3,...). The use of entirely connected layers allowed us to create a working model. To categorise outputs like cat, dog, automobile, truck, etc., I employ an activation function like SoftMax or sigmoid.

### *Examples of deep learning applications*

AI in Finance

AI has already been put to use in the financial technology industry to speed up processes, cut down on expenses, and increase returns. Improved credit scoring made possible by deep learning is revolutionising the banking sector. Machine learning may help credit decision- makers analyse applicants' creditworthiness and lend money more quickly and accurately by considering their character and financial standing.

Underwrite is a FinTech firm that offers an AI-based credit-making solution to other businesses. Underwrite.ai using machine learning to determine which loan applicants are most likely to repay their debt. The performance of their technique is much higher than that of more conventional approaches (Myint, Khaing, 2016).

AI in HR

(Kim, MacKinnon, 2018) With the use of AI, Under Armour, a sportswear firm, completely revamps the applicant experience. Actually, Under Armour has cut down on hiring times for its retail outlets by 35%. Back in 2012, Under Armour was seeing a surge in interest due to its increasing popularity. Every month, they received over 30,000 resumes. There was too much downtime when reading through submissions and initiating the screening and interview process. Under Armour was unable to have its shops fully staffed and ready to run due to the long process of hiring and on boarding new employees.

Under Armour had all the must have' HR technologies in place, including transactional solutions for sourcing, applying, tracking, and on boarding, but the tools weren't enough. Hire Vue, an artificial intelligence (AI) provider for HR solutions, is being used by Under Armour for both on-demand and live interviews. They were only blowing smoke; in reality, they reduced fill time by 35%. The trade-off was the recruitment of more qualified employees.

AI in Marketing

Artificial intelligence is a helpful resource for handling problems related to customer service management and customization. Through the use of AI methods, voice recognition in contact- centre management and call routing has been greatly enhanced, leading to a more streamlined experience for callers.

Using deep learning, computers can analyse audio and determine how a client is feeling. The interaction may be redirected to human operators in case the consumer is having difficulty communicating with the AI chatbot.

AI is extensively applied in many different fields and businesses; the aforementioned three instances of Deep Learning are just a few.

*Why is Deep Learning Important?*

(Dimililer, Kamil, 2017) The predictive results of deep learning may be put to practical use immediately. Deep learning is most effective when used for knowledge-based prediction and pattern discovery (unsupervised learning). Deep learning runs on a firehose of big data. The synergy between the two may have a profound impact on an organization's output, revenue, management, and creativity.

When compared to more conventional approaches, deep learning often produces better results. The accuracy of deep learning algorithms is higher than that of machine learning algorithms in many areas of application, including image classification (41%), face recognition (27%), and voice recognition (25%).

### *Limitations of deep learning*

Data Labelling

This is because "supervised learning" is the standard method currently used to train AI models. This necessitates manual labelling and categorization of the underlying data, a task that may be time-consuming and prone to human mistake. As an example, hundreds of people are being hired by businesses working on self-driving car technology to manually annotate hours of video feeds from test cars.

Obtain huge training datasets

It has been shown that basic deep learning methods, such as CNN, may sometimes mimic the expertise of human professionals in domains as diverse as medicine and linguistics. The latest generation of machine learning, however, demands training data sets that are not just labelled but also sufficiently vast and global.

To achieve human-level performance, deep-learning models often needed millions of training examples. Big IT businesses are employing big data to amass petabytes of data, so it should come as no surprise that deep learning is all the rage there. Allowing them to develop a deep learning model that is both spectacular and precise.

### *Syntax and Semantics*

It is the goal of Python to maintain its readability. Its formatting is visually clean and frequently uses significant phrases from Europe where other languages would use punctuation. In contrast to many other languages, such no longer uses curly brackets to denote blocks or semicolons to indicate that statements are optional. Compared to C and Pascal, it features less syntactic exceptions and unique situations.

Statements and control flow

Python's assertions include (among others):

The challenge articulation (token '=', the equivalents sign). This workshop uniquely in distinction in imitation of between traditional basic programming dialects, yet this necessary component (counting the concept concerning Python's rendition concerning factors) enlightens numerous distinct highlights regarding the language. Task into C, e.g., x = 2, capability "composed alternative honour x gets a reproduction over numeric value 2". The (right-hand) laurels is replicated into an allotted stockpiling place because of which the (left-hand) volatile renown is the emblematical location. The intelligence particular in conformity with the moving is sufficiently giant (conceivably absolutely huge) because the announced kind. In the least complicated instance of Python task, using a comparable model, x = 2, ability "(nonexclusive) odour x receives a allusion in imitation of a different, anxiously specified goal about numeric (int) variety about big virtue 2." This is named restricting the honour after the item. Since the name's stockpiling area would not incorporate the confirmed esteem, such is inconsistent according to think about that a variable. Names may keep as a result leap back whenever to objects concerning extensively differing types, such as strings, strategies, complex gadgets with information or techniques, or then forth Progressive duties over a standard cost in conformity with different names, e.g., x = 2; y = 2; z = 2 result of assigning stockpiling according to (probably) ternary names yet certain numeric item, according to as each regarding the three names are bound. Since a name is a traditional allusion holder that is queer according to anticipate a fixed data kind together with it. Anyway, at a partial epoch a honour pleasure definitely some object, which pleasure bear a sort; as a result there is dynamic composing.

* + - The salvo articulation, as restrictively executes a square concerning code, alongside over and above or elif (a constriction of else-if).
    - For articulation, which emphasizes above an iterable item, catching each and every component after a local variable for makes use of by way of the full square.
    - Whilst explanation, who executes a square about articles namely lengthy as its situation is valid.
    - The attempt proclamation, who permits distinctive instances delivered above in its appended articles square in imitation of lie gotten yet done seriousness about by using aside beside provisions; such additionally guarantees to that amount coherent on code between an at lengthy remaining square pleasure persistently be born at the back of agenda of what the square exits.
    - The develop articulation, chronic after developing a predefined distinct action and re- raise a acquired exemption.
    - The class articulation, as executes a square regarding code then appends its nearby namespace in imitation of a class, because of utilizes of object-situated programming.
    - The def proclamation, as characterizes a potential yet technique.
    - The including proclamation, beyond Python 2.5 delivered about September 2006, which encases a code barrier internal a putting overseer (for instance, purchasing a lock before the rectangular concerning articles is drive then handing over the storey a quick time later, or rudiments a report and later on shutting it), permitting Resource Acquisition Is Initialization (RAII)- as leading yet replaces a traditional attempt/at last colloquialism.
    - The pass by articulation, as fills into as a NOP. It is linguistically predicted according to fulfil an unfilled code block.
    - The set forth explanation, utilized for the duration of troubleshooting to take a look at because conditions up to expectation ought to apply.
    - The generate articulation, who returns a worth beside a creator work. From Python 2.5, spawn is also an administrator. This structure is utilized in conformity with execute coroutines.
    - The income articulation, which is utilized to income modules whose capacities, yet factors be able keep utilized into the present-day program. There are ternary special approaches about using import: arrival <module name=""> [as <alias>] and beside

<module name=""> import \* then from <module name=""> inhalant <definition 1=""> [as <alias 1="">], <definition 2=""> [as <alias 2="">]

* + - The mark articulation was once changed in conformity with the print () labor in Python 3.
    - </alias></definition>.

Expressions

Some Python articulations are kind of dialects like C or Java, whilst some are not:

Expansion, deduction, then enlarge are something absolutely similar, yet the propulsion over part contrasts. There are joining sorts regarding divisions within Python. They are ground divide or number division. Python also brought the \*\* director because exponentiation.

From Python 3.5, the latter @ infix manager used to be presented. It is anticipated in conformity with be utilized by using libraries like NumPy for skeleton augmentation.

In Python, == looks at by way of esteem, with Java, which analyses numeric by means of value and objects by means of reference. (Worth examinations into Java concerning objects perform be celebrated including the equivalents () strategy.) Python's is administrator would possibly lie utilized in accordance with look at aim personalities (correlation by means of reference). In Python, examinations may remain tied, for instance a &lt;= b &lt;= c.

Python usage the words and, or, not because its Boolean directors as an alternative about the consultant &amp; &amp; ||, ! utilized of Java or C.

Python has a sort over articulation named a rundown appreciation. Python 2.4 accelerated rundown appreciations of an extra extensive articulation named a generator articulation. Unknown capacities are executed making use of lambda articulations; nonetheless, this are restricted among that the physique have to be certain articulation.

Techniques

After the article's class, the condemnation shape case, techniques regarding objects are capabilities that are added. Because of common strategies and capacities, method (argument) is syntactic sugar because of Class (instance, contention). As opposed to the implicit self (or this) in incomplete mean composition organised programming dialects (e.g., C++, Java, Objective-C, then Ruby), Python procedures have a clear self-interest limit after coming in accordance with lawsuit information.

Composing

Python uses untyped variable names in place of composed or absence of composed gadgets. Type imperatives are no longer common in today's society; instead, the way in which an object is described may come up briefly, indicating that the devoted composition is not of a lifelike sort. Notwithstanding life regularly composed, Python is specifically, denying things to do to that amount are not distinct (for instance, including a variety in accordance with a string) as a substitute about quietly alert after figure oversea them.

Python enables programmers to characterise their own kinds using classes, which are frequently used in object-oriented programming. New occurrences of instructions are advanced with the aid of bread the class (for instance, Spam Class() or Eggs Class()), or the training are examples concerning the meta class type (itself an event on itself), enabling meta programming and reflection.

LIBRARIES:

TensorFlow

In order to speed up numerical computation, Google produced and distributed the TensorFlow Python library. Because it is built on top of TensorFlow, it may be used to generate Deep Learning models directly or via wrapper libraries.

Keras

Deep learning API created by Google, Keras is used to construct neural networks. Developed in Python, it facilitates the creation of neural networks. Multiple neural network backends are also supported.

Image Data Generator:

The rotation range option of the ImageDataGenerator class enables you to generate random rotations of pictures in a range of 0 to 360 degrees.

RMS Prop

The step size for each parameter is changed using a decaying average of partial gradients, which is an extension of gradient descent and AdaGrad's gradient descent. Squared Root Mean Propagation (RMSProp).

What is Flask Python

A Python module called Flask is a web framework that makes it simple to build web apps. The framework's core does not contain any ORM (Object Relational Manager) or other features, making modifications simple.

Although it has several useful features, including URL routing and a template engine, it is not faultless. Framework for creating WSGI-compliant web-based applications.

What is a Web Framework?

It is possible to build online apps without worrying about protocol, thread management, and other low-level issues with the help of a Web Application Framework.

What is Flask?

A Python-based framework for creating web apps is called Flask. The project was conceived up by Armin Ronacher, the leader of the Poocco group of Python enthusiasts from across the world. Flask is an open-source web framework built on the Werkzeg WSGI Toolkit and the Jinja2 Template Engine. Both of these are Pocco projects.

WSGI

When it comes to writing Python online applications, many programmers employ the Web Server Gateway Interface (WSGI). The WSGI specification establishes a standard means of communication between web servers and web-based applications.

Werkzeug

In Werkzeug, a WSGI toolkit, requests, response objects, and utility methods are all implemented. As a result, a web frame might be built on top of it. On Werkzeg, the Flask framework is based.

jinja2

jinja2 is a popular Python template engine. It's possible to create dynamic web pages by using web templates and data sources that are linked together.

The following example shows how to pass Python variables into HTML templates:

Text, letter

Description automatically generated

Micro framework

One of the most common ways to describe Flask is as a microframework. In order to maintain the core of the program simple and scalable, it has been developed.

As opposed to providing an abstraction layer, Flask allows developers to implement database capability directly into their applications using extensions.

Why is Flask a good web framework choice?

Unlike the Django framework, Flask leans heavily on the Pythonic. Because Flask doesn't have a steep learning curve, it's simple to get started. Additionally, the level of explicitness makes it easier to follow. Only a few lines of code are required to build the "Hello World" application.

This is a boilerplate code example.

Text

Description automatically generated

Figure 19 - Boilerplate Code Example

You can easily build on the personal computer if you want to. Run this program by typing python server.py and save it as server.py.

Text

Description automatically generated with medium confidence

The public is then given access to a web server that can only be accessible by devices linked to the local network after that. Open localhost on port 5000 in the web browser to see "Hello World" (the URL).

Graphical user interface, text

Description automatically generatedSome example output:

Figure 20 - example output

Even though it's a micro framework, it doesn't imply the whole program should be contained in a single Python file. Many files are an excellent way to manage the complexity of bigger applications.

Micro refers to the Flask framework's ability to be both simple and flexible. If you wish to utilise an ORM, you may do so, but Flask doesn't make such selections for you.

A popular web framework like Flask is always up-to-date and cutting-edge. Adding new features is a cinch. Complex applications may benefit from it being scaled up.

### *Deep Learning Graphical User Interface (GUI)*

For this task, I took use of Python 3.7's GUI, and it worked well. With the help of the FLASK GUI library, I've created an intuitive and easy-to-use graphical user interface (GUI) for the CNN deep learning algorithms that I've employed in my study. Appendix has detailed explanations of each of these functions. Source code for the GUI is also supplied.

### *Legal Issues*

The use of deep learning in a bone fracture detection research project may raise a number of legal issues, depending on the specific details of the project. Some potential legal issues that may arise include the following:

Privacy concerns: The use of deep learning algorithms in a research project may involve the collection, storage, and processing of personal data, such as medical images or other health information.

Intellectual property rights: The development and use of deep learning algorithms in a research project may implicate intellectual property rights, such as patents, copyrights, or trade secrets.

Clinical trial regulations: If the bone fracture detection research project involves testing the performance of the deep learning algorithms on real patients, it may be considered a clinical trial. In this case, the project may need to comply with applicable clinical trial regulations, such as the International Conference on Harmonisation (ICH) Good Clinical Practice (GCP) guidelines.

### *Ethical Issues*

There are several ethical issues that may arise in a deep learning research project for bone fracture detection.

Data privacy: The research project may involve collecting and using sensitive medical data from individuals. This raises concerns about protecting the privacy of the individuals and ensuring that their personal information is not disclosed without their consent.

Bias and discrimination: The use of deep learning algorithms in medical research has the potential to introduce bias and discrimination into the analysis of data.

Access to healthcare: Deep learning algorithms have the potential to improve the accuracy and efficiency of medical diagnosis, but they may also create barriers to access to healthcare for certain groups of people.

Clinical relevance: The results of the deep learning research project may not always be clinically relevant or applicable to real-world situations. This could lead to incorrect or misleading conclusions about the effectiveness of the technology for detecting bone fractures.

### *Social Issues*

One potential social issue related to a bone fracture detection using deep learning research project could be the potential for bias in the algorithms used. If the data used to train the algorithm is not representative of the diverse population it will be used on, it could lead to unequal or inaccurate detection of bone fractures for certain groups.

Another social issue could be the potential for misuse of the technology. If the technology is not regulated or used properly, it could lead to false diagnoses and unnecessary medical procedures.

### *Professional Issues*

There are several professional issues to consider when conducting a deep learning research project for bone fracture detection.

Ethical concerns: The use of deep learning algorithms for medical imaging raises ethical concerns regarding the privacy and confidentiality of patient information. It is important to ensure that appropriate measures are in place to protect patient data and maintain patient privacy.

Data quality: The quality of the data used for training and testing the deep learning algorithms is crucial for the accuracy and reliability of the results. It is important to carefully collect and curate the data to ensure its validity and relevance.

Expertise and training: Conducting deep learning research requires a high level of expertise in both computer science and medical imaging. It is important to ensure that the research team has the necessary skills and knowledge to conduct the research project effectively.

Collaboration: Collaborating with other researchers, medical professionals, and stakeholders can help to improve the quality and relevance of the research project. It is important to establish effective communication and collaboration channels to facilitate the exchange of ideas and expertise.

Regulatory compliance: The use of deep learning algorithms for medical imaging is subject to various regulatory requirements, including those related to data privacy and the accuracy and reliability of the results. It is important to ensure that the research project complies with all relevant regulations and guidelines.

# **CHAPTER 4 – RESULTS AND DISCUSSIONS**

The training of cases requires obligatorily using datasets that are based on bone fracture instances. The GitHub website allowed me to download a dataset that was compiled based on the findings of fractures. The dataset that was acquired has a great deal of information that will be useful in answering the subject that I was researching. Within the scope of this investigation, the performance metrics of several learning models are analysed and compared on the basis of the accuracy outcomes. The accuracy between the models may be improved by deep learning algorithms such as CNN. I analysed and contrasted a variety of learning models to see which ones had the potential to provide more accurate forecasts of bone fracture cases.

## **Experimental Setup**

In this research work, the system is based on CNN techniques and Python version 3.7 software. All of my trials were carried out on a computer with the specifications mentioned in the table below as part of my research project. A list of available Python library packages is also included in the table. I've used the FLASK GUI library and the graphical user interface (GUI) for my project.

|  |  |
| --- | --- |
| **Specifications of the Computer** | **Details of the configuration** |
| Operating System | Windows 10 |
| Random Access Memory (RAM) | 8 Gigabyte (GB) |
| System Type | 64-bit Windows OS |
| Base Processor | x64 |
| Processor | Intel Core i5 -10470U Processor |
| Generation | 7th Generation |
| Graphics | 4 (GB) (Radeon graphics) |
| Clock Speed | 4.80Ghz |
| Python Version | Version 3.7 |
| Python Library | Tensorflow, Keras, Image Data Generator,RMSProp. |

|  |  |
| --- | --- |
| GUI Library | FLASK |

***Table 1 - System Specifications***

## **Software Specifications**

Python **-** A high-level, all-purpose programming language is Python. It works with many different theoretical frameworks. The standard library in Python allows you to do a wide range of tasks. Python is simple to learn and use because it has a wide variety of functions and libraries. In my work, programming skills different than those employed in the experiment are utilised. The following databases were checked out while doing this inquiry.

NEED FOR PYTHON

* + - Python can be used on a number of platforms (Windows, Mac, Linux, Raspberry Pi, etc).
    - Python is simple to learn because its language resembles that of English quite a little.
    - Python runs on an interpreter, which enables code to be executed quickly, and allows programmers to start coding right away. Python's syntax allows developers to design programmes with less lines than in some other programming languages. This suggests that prototypes might be made within a few hours.
    - Procedural, object-oriented, or functional approaches are all valid ways to approach Python development.

GOOD TO KNOW

* + - Here, working with the most current major version of Python, Python 3. It's still popular, even if Python 2 hasn't been updated in a while other than security fixes.
    - Use a text editor like Notepad for this lesson. Integrated Development Environments (IDEs) like Thonny, Pycharm, Netbeans, or Eclipse may be used to create Python code.
    - Python Syntax compared to other programming languages
    - For readability, Python was developed with the English language in mind, but it also has a strong impact from arithmetic.
    - Other programming languages commonly utilise semicolons or parentheses to finish commands in Python.
    - When describing scope, such as the scope of loops, functions, and classes, Python employs whitespace. Curly brackets are used to accomplish this in different programming languages.

FEATURES

Python is a computer language with multiple worldviews. Laptop programs for article- structured programming and organised writing are unquestionably supported, but a sizable portion of their highlights pertain to practical programming and perspective-focused programming.

Python makes use of potential composing, and a mixture about notice tallying yet a cycle- distinguishing metropolis employee for memory the board. It moreover includes brawny honor aim (late restricting), who ties technique or unstable names at some stage in software execution. Python's design presents some assist because of utilitarian programming of the Lisp custom. It has channel (), map(), then decrease() capacities; list cognizances, phrase references, or sets; and producer articulations.

The grade library features additional modules (itertools and functools) that add up to all of the useful tools acquired from Haskell and Standard ML. The file summarises the language's primary question-answering route. The Zen of Python (PEP 20), which includes proverbs like

* + - Readability counts,
    - Beautiful is better than ugly,
    - Explicit is better than implicit,
    - Simple is better than complex and
    - Complex is better than complicated.

## **Results**

All the GUI's features are detailed in the Table:

|  |  |
| --- | --- |
| **Icons** | **Categories** |
| Buttons | 1. Upload Image 2. Predict |
| Entry Box | Upload Image (.jpg) is the entry box |
| Loss | CrossEntropy |

***Table 2 - Functions of the GUI***

**Graphical user interface, text, application, email

Description automatically generated**

Figure 21 - Importing Data

The above screenshot displays the libraries being imported also displays the dataset being imported onto google colab from local environment. The imported data which is a compressed file is then extracted and used.

**Graphical user interface, text, application, email

Description automatically generated**

Figure 22 – Exploring Data

Graphical user interface, text, application, email

Description automatically generated

Figure 23 - Image Configuration

The above image displays the configuration of the images from the dataset. This helps us to provide a quality dataset as an input, so that the results or the outcome can be accurate. The size of the image in dataset in the above figure is 1152\*1152 pixels.

The extracted dataset is then being used to train the model, when the model is trained, the output of total number of images present in the dataset and also the above screenshot displays the directories of the images present in the dataset.

**Table

Description automatically generated**

Figure 24 – Building a small convnet from scratch

The above screenshot shows the building of a small convnet from scratch. This involves designing a deep learning model that will be used to analyse and classify the images. The model typically consists of several layers of interconnected neurons that are capable of recognizing patterns and features in the data. This involves feeding the pre-processed data into the deep learning model and training it to recognize the patterns and features in the images. The model is typically trained using a process called backpropagation, where the model's errors are used to adjust its internal weights and biases in order to improve its performance.

**Table

Description automatically generated**

Figure 25 – Evaluating Accuracy and loss for a model

The above screenshot displays the images after they have been trained, it shows the accuracy and loss for a model along with the time taken for each step along the way taken for training,

Graphical user interface, website

Description automatically generated

Figure 26 – Screenshot 1

The above image shows the webpage of the image that is to upload and predict whether the patient is suffering from bone fracture or not.

Graphical user interface, application, website

Description automatically generated

Figure 27 – Screenshot 2

Graphical user interface, application

Description automatically generatedIn the above image I had uploaded the negative image and it has predicted the results based on training I have done on the image dataset by using CNN algorithm.

Figure 28 – Screenshot 3

In the above image I had uploaded the positive image and machine has predicted the results based on training I have done on image dataset by using the CNN algorithm.

A plot plotted between the model accuracy, model loss and epoch are as follows:

Chart, line chart

Description automatically generated

Figure 29 - Accuracy vs Epoch

Chart, line chart

Description automatically generated

Figure 30 - Loss vs Epoch

The loss of the model is diminishing, as is evident, as the model's accuracy rises.

**CHAPTER 5 – CONCLUSION AND FUTURE WORK**

## **CONCLUSION**

As part of this article, a framework for bone fracture detection and characterization was developed using a deep learning method. Images of a human broken bone and sound bone were used to simulate the courtroom scene. The first 100 photographs were compiled from multiple sources, therefore in order to avoid the issue of trying to squeeze too much information into a small informational collection, it was decided to expand the informative index. Finally, the index's size was set to 4000 records. Only 92.44 percent of the model is accurate for the solid and broken bones. 82.89 percent and 84.7 percent, on the other hand, fall significantly short of the suggested accuracy. The model's accuracy may be further enhanced by using a more sophisticated learning model. The framework has to be approved by the larger informative collection in order to further investigate the show.

The model still has a few problems. It could be possible to add more components to the model to improve its accuracy. Additionally, majority of the models need to handle a wider variety of fractures and their subtypes. These issues will be resolved in the future. In the end, a better bone fracture prediction model will be created using mass high-reliability variants and algorithms. Studying the many medical literature types on bone fracture data and comprehending their variance types posed the biggest obstacle for me during the research period. This is necessary since I need to get a dataset that is sufficiently large and contains the necessary properties. Training this dataset while preventing overriding in the dataset has been my next problem. I was able to successfully acquire a dataset and train it without suffering any data loss.

I have looked at a variety of techniques for locating and classifying fractures. It is evident that fracture detection is a strong suit for CNN-based models, especially InceptionNet and XceptionNet. With the aid of fracture detection systems, the labor- and time-intensive procedure of skilled radiologists diagnosing and predicting radiographs might be computerized. According to several of the academics cited, the absence of labelled training data is the main impediment to developing a high-performance classification algorithm. There isn't a universal model that can be used with the given data, though. I have tried to describe how DL is used in medical imaging and how it can help the radiologist make a precise diagnosis.

I have tried to describe how deep learning is used in medical imaging and how it can help the anthropologists make a precise diagnosis. The fear of job loss among employees makes it difficult to fully implement deep learning in anthropologists. I am not attempting to replace the anthropologists; rather, I am attempting to assist them in their work.

## **FUTURE SCOPE**

The detection of bone fractures with the use of image processing is one of the goals of my study. Within the context of this project, I am providing the input in the form of an X-ray picture of the user; the system will then categorize this image via the application of an algorithm and provide the result. The bone fracture detection technique is really helpful. This method is simple enough that anybody can use it.

Anthropologists are worried about DL holding the position of radiologist in the future. But this is not the case. Never is a rather simple answer to this question. However, in the era of artificial intelligence, anthropologists' working lives will surely change. The application of DL algorithms will enable the detecting bone fracture process to do a variety of typical activities more quickly and effectively. However, anthropologists' profession is challenging since they must handle complex medical issues.

Automatic annotation can be quite helpful to the anthropologists in providing an accurate diagnosis. Instead of opposing the adoption and implementation of automation into working practices, the real problem is accepting the inevitable shift in the radiological profession by integrating DL into detection procedures. The risk that "humans will obey a machine's instructions because humans are attracted by the machines and rely on them to make key decisions" is one of the most probable ones. This can be minimized if an anthropologist informs themselves and their future colleagues about the benefits of DL, collaborates with researchers to ensure proper, safe, meaningful, and useful deployment, and ensures that the usage is primarily for the benefit of the patients.

# **REFERENCES**

1. Tripathi, A. M., Upadhyay, A., Rajput, A. S., Singh, A. P., & Kumar, B. (2017, March). Automatic detection of fracture in femur bones using image processing. In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) (pp. 1-5). IEEE.
2. Chai H.Y, Wee K.L, Swee T.T, Salleh S.H, Ariff A.K , Kamarulafizam(2011), Gray- Level Co-occurrence Matrix Bone Fracture Detection .American Journal of Applied Sciences vol 8, pp 26-32.
3. S. Myint, A. S. Khaing& H. M. Tun (2016), Detecting Leg Bone Fracture in X-ray Images, International Journal of Scientific & Research, vol. 5, Jun, pp. 140-144.
4. Mahendran S. K. &BabooS. S. (2011), An Enhanced Tibia Fracture Detection Tool Using Image Processing and Classification Fusion Techniques in X-Ray Images, Global Journal Of Computer Science and Technology, vol. 11, pp. 27-28.
5. V. D. Vegi , S. L. Patibandla, S. SKavikondala, Z. Basha (2016), Computerized Fracture Detection System using x-ray Images, International Journal of Control Theory and Applications, vol. 9, pp. 615-621.
6. McBee& Morgan P. (2018) Deep learning in radiology, in Academic radiology, vol 25, issue 11, pp 1472-1480.
7. Kim, D. H., & T. MacKinnon (2018) Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks, Clinical radiology vol 5,pp 439-445.
8. Dimililer&Kamil(2017), IBFDS: Intelligent bone fracture detection system, Elsevier Procedia computer science vol 120 pp 260-267.
9. 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 Data Set: (Fracture Bone) Available: https://[www.iiests.ac.in/](http://www.iiests.ac.in/) last access: Sep 14, 2019.
10. Korfiatis, Vasileios Ch., TassaniS.,George K. M.(2018), A new Ensemble Classification System for fracture zone prediction using imbalanced micro-CT bone morphometrical data, IEEE journal of biomedical and health informatics vol 22,issue 4,pp 1189-1196.
11. ZiedanR. H., Mohamed A., EltawelG.S.(2016),Selecting the appropriate feature extraction techniques for automatic Medical Image classifications, International journal of Emerging Engineering Research and Technology, Volume 4, pp 1-9.
12. Krizhevsky, A.; Sutskever, I., Hinton, G.E.(2012), ImageNet classification with deep convolutional neural networks.In Proceedings of the 25th International Conference on Neural Information Processing Systems, Stateline,NV, USA, 3–6 December 2012; pp. 1097–1105.
13. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.; Schmidhuber, J.(2011), Flexible high performance convolutionalneural networks for image classification. In Proceedings of the International Joint Conference on ArtificialIntelligence(IJCAI), Catalonia, Spain, 16–22 July 2011; Volume 22, p. 1237.
14. Ebsim, Raja, Naqvi J., Timothy F.(2018), Automatic Detection of Wrist Fractures From Posteroanterior and Lateral Radiographs: A Deep Learning-Based Approach, International Workshop on Computational Methods and Clinical Applications in Musculoskeletal Imaging, Springer, Cham,pp 114-125
15. Yadav, D. P., Sharma, A., Singh, M., & Goyal, A. (2019). Feature Extraction Based Machine Learning for Human Burn Diagnosis From Burn Images. IEEE Journal of Translational Engineering in Health and Medicine, 7, 1-7.
16. Rathor, S., & Jadon, R. S. (2019). The art of domain classification and recognition for text conversation using support vector classifier. International Journal of Arts and Technology, 11(3), 309-324.
17. Chen, Y., Yap, W.H., Leow, W.K., Howe, T.S. and Png, M.A. (2004) Detecting femur fractures by texture analysis of trabeculae, Proc. Int. Conf. on Pattern Recognition.
18. Donnelley, M. (2008) Computer aided long-bone segmentation and fracture detection, Computer and Information Science, Ph.D Thesis.
19. Duda, R. and Hart, P. (1972) Use of the hough transformation to detect lines and curves in pictures, Comm. ACM, Vol. 15, Pp. 11–15.
20. Federico, A. and Kaufmann, G.H. (2007) Denoising in digital speckle pattern interferometry using wave atoms, Opt. Lett., Vol. 32, Pp.1232-1234. An Enhanced Tibia Fracture Detection Tool Using Image Processing and Classification Fusion Techniques in X-Ray Images © 2011 Global Journals Inc. (US) 27
21. Kaur, A. and Singh, K. (2010) Speckle noise reduction by using wavelets, National Conference on Computational Instrumentation CSIO NCCI 2010, INDIA, Pp. 198- 203.
22. Lim, S.E., Xing, Y., Chen, Y., Leow, W.K., Howe, T.S. and Png, M.A. (2004)
23. Detection of femur and radius fractures in x-ray images, Proc. 2nd Int. Conf. on Advances in Medical Signal and Info. Proc..
24. Lim, V.L.F., Leow, L.W., Chen, T., Howe, T.S. and Png, M.A. (2005) Combining classifiers for bone fracture detection in X-ray images, IEEE International Conference on Image Processing, ICIP 2005, Vol. 1, Pp.I - 1149-1152.
25. Lim, V.L.F., Leow, L.W., Chen, T., Howe, T.S. and Png, M.A. (2005) Combining classifiers for bone fracture detection in X-ray images, IEEE International Conference on Image Processing, ICIP 2005, Vol. 1, Pp.I - 1149-1152
26. Luisier, F., Vonesch, C., Blu, T. and Unser, M. (2010) Fast interscale wavelet denoising of Poissoncorrupted images, signal Processing, Elsevier, Pp. 415–427.
27. Koyuncu, H. and Sharma, D. (2021). Deep learning approach to analyse, detect and classify COVID-19 patients. *New Trends in Mathematical Science*, 1(9 Special Issue), pp.1–2. doi:10.20852/ntmsci.2021.420.
28. Patil, K.A., Prashanth, K.M. and Ramalingaiah, Dr.A. (2021). A comparative study on detection of osteoporosis using deep learning methods: A review. *International Journal of Orthopaedics Sciences*, 7(3), pp.108–115. doi:10.22271/ortho.2021.v7.i3b.2736.

# **APPENDICES**

## **Front End Code**

app = flask.Flask( name ) app.secret\_key = "secret key"

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER app.config['MAX\_CONTENT\_LENGTH'] = 16 \* 1024 \* 1024

app.config["DEBUG"] = True CORS(app)

ALLOWED\_EXTENSIONS = set(['png', 'jpg', 'jpeg', 'gif']) def allowed\_file(filename):

return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

@app.route('/', methods = ['GET','POST']) def index():

context={'method':request.method} if request.method == 'POST':

context = predict() context['method'] =request.method # print(context)

return render\_template("index.html", context=context)

@app.route('/success', methods = ['POST']) def success():

if request.method == 'POST': context = {'success': False} # if 'file' not in request.files:

# flash('No file part')

# return redirect(request.url) file = request.files['file']

if file.filename == '':

flash('No image selected for uploading') return redirect("/")

if file and allowed\_file(file.filename): filename = secure\_filename(file.filename)

path = os.path.join(app.config['UPLOAD\_FOLDER'], filename) file.save(path)

#print('upload\_image filename: ' + filename) flash('Image successfully uploaded and displayed below') print(path)

context['filename'] = filename context['path'] = path

# predicting images

result = bonefracture(path) context['result'] = result context['success'] = True

# return redirect("/")

return render\_template("success.html", context=context) else:

flash('Allowed image types are -> png, jpg, jpeg, gif') return redirect("/")

def predict():

context = {'success': False} # if 'file' not in request.files:

# flash('No file part')

# return redirect(request.url)

# context['msg'] = "No file part " # return context

file = request.files['file'] if file.filename == '':

# flash('No image selected for uploading') context['msg'] = "No image selected for uploading" return context

if file and allowed\_file(file.filename): filename = secure\_filename(file.filename)

path = os.path.join(app.config['UPLOAD\_FOLDER'], filename) file.save(path)

#print('upload\_image filename: ' + filename)

# flash('Image successfully uploaded and displayed below') context['msg'] = "Image successfully uploaded and displayed below" # print(path)

context['filename'] = filename context['path'] = path

# predicting images

result = bonefracture( path) context['result'] = result

context['success'] = True # return redirect("/") return context

else:

# flash('Allowed image types are -> png, jpg, jpeg, gif') context['msg'] = 'Allowed image types are -> png, jpg, jpeg, gif'

return context @app.route('/display/<filename>') def display\_image(filename):

# print('display\_image filename: ' + filename)

return redirect(url\_for('static', filename='uploads/' + filename), code=301) if name == ' main ':

app.run(debug=True) # pass DEBUG param</pre>

# -\*- coding: utf-8 -\*- """bone\_fracture.ipynb

Automatically generated by Colaboratory.

Original file is located at https://colab.research.google.com/drive/1d1OsZZfJHrCCkP6Z-HpYLgsVZ7PKY8Ef

"""

import os import zipfile

from google.colab import drive drive.mount('/content/gdrive')

df='/content/gdrive/MyDrive/Bone\_Fracture.zip' df1=zipfile.ZipFile(df,'r') df1.extractall('/content/gdrive/MyDrive/Bone\_Fracture') df1.close() os.listdir('/content/gdrive/MyDrive/Bone\_Fracture')

train\_positive\_dir = os.path.join('/content/gdrive/MyDrive/Bone\_Fracture/Positive') train\_negative\_dir = os.path.join('/content/gdrive/MyDrive/Bone\_Fracture/Negative')

train\_positive\_names = os.listdir(train\_positive\_dir) print(train\_positive\_names[2:5])

train\_negative\_names = os.listdir(train\_negative\_dir) print(train\_negative\_names[10:])

print('total training normal images:', len(os.listdir(train\_normal\_dir))) print('total training glaucoma images:', len(os.listdir(train\_glaucoma\_dir)))

# Commented out IPython magic to ensure Python compatibility. # %matplotlib inline

import matplotlib.pyplot as plt import matplotlib.image as mpimg

# Parameters for the graph; the output images in a 4x4 configuration nrows = 4

ncols = 4

pic\_index = 0 fig = plt.gcf()

fig.set\_size\_inches(ncols \* 4, nrows \* 4)

pic\_index += 8

next\_positive\_pix = [os.path.join(train\_positive\_dir, fname)

for fname in train\_positive\_names[pic\_index-8:pic\_index]] next\_negative\_pix = [os.path.join(train\_negative\_dir, fname)

for fname in train\_negative\_names[pic\_index-8:pic\_index]]

## **Training Code**

import tensorflow as tf

model = tf.keras.models.Sequential([

# Note the input shape is the desired size of the image 300x300 with 3 bytes color # This is the first convolution

tf.keras.layers.Conv2D(16, (3,3), activation='relu', input\_shape=(300, 300, 3)),

tf.keras.layers.MaxPooling2D(2, 2), # The second convolution

tf.keras.layers.Conv2D(32, (3,3), activation='relu'), tf.keras.layers.MaxPooling2D(2,2),

# The third convolution tf.keras.layers.Conv2D(64, (3,3), activation='relu'), tf.keras.layers.MaxPooling2D(2,2),

# The fourth convolution tf.keras.layers.Conv2D(64, (3,3), activation='relu'), tf.keras.layers.MaxPooling2D(2,2),

# The fifth convolution

tf.keras.layers.Conv2D(64, (3,3), activation='relu'), tf.keras.layers.MaxPooling2D(2,2),

# Flatten the results to feed into a DNN tf.keras.layers.Flatten(),

# 512 neuron hidden layer tf.keras.layers.Dense(512, activation='relu'), tf.keras.layers.Dense(1, activation='sigmoid')

])

model.summary()

from tensorflow.keras.optimizers import RMSprop model.compile(loss='binary\_crossentropy',

optimizer=RMSprop(lr=0.001), metrics=['acc'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator train\_datagen = ImageDataGenerator(rescale=1/255)

train\_generator = train\_datagen.flow\_from\_directory('/content/gdrive/MyDrive/Bone\_Fracture', # This is the source directory for training images

target\_size=(300, 300), # All images will be resized to 150x150 batch\_size=8,

# Since I use binary\_crossentropy loss, I need binary labels class\_mode='binary')

his=model.fit\_generator( train\_generator, steps\_per\_epoch=8, epochs=100, verbose=1)

model.save('/content/gdrive/MyDrive/bone\_model.h5')