



**COLLEGE OF ENGINEERING  
DESIGN ART, AND  
TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL  
AND COMPUTER ENGINEERING**

**AUTOMATIC BONE FRACTURE  
DETECTION IN X-RAY IMAGES  
USING DEEP LEARNING**

**SSERUBOMBWE RICHARD  
18/U/36137/PSA**

**Supervisors**

**Main Supervisor: DR. COSMAS MWIKIRIZE**

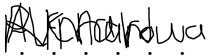
**Co Supervisor: DR. ANDREW KATUMBA**

*A final year project report submitted in partial fulfillment of  
the requirements for the award of the degree of Bachelor of  
Science in Electrical Engineering*

October 3, 2022

# Declaration

I SSERUBOMBWE RICHARD, sincerely declare that to the best of my knowledge, the information presented in this project report is an original work resulting from my sole effort and intellect. Except for extracts whose references are stated herein, it has never been published or submitted to this organization or any other institution of training for any academic reward.

Signature. . . . .  . . . . .

.  
Date. . . . . 10/10/2022 . . . . .

# Approval


This report has been submitted for examination with the approval of the project supervisors.

## Main Supervisor

Dr. Cosmas Mwikirize

Lecturer

Department of Electrical and Computer Engineering Makerere University

Signature. . . . .  . . . . .

Date. . . . . [October 3, 2022](#) . . . . .

.

## Co-Supervisor

Dr. Andrew Katumba

Lecturer

Department of Electrical and Computer Engineering Makerere University

Signature. . . . .  . . . . .

Date. . . . . [10.10.2022](#) . . . . .

.

# Dedication

I dedicate this report to my parents who morally, financially, and spiritually supported me throughout the execution of the project. I also dedicate it to my project partner Isreal Ssekitoleko with whom I shared experiences and combined knowledge to accomplish the project. We shall always look up to such wonderful learning experiences. Especially, I dedicate this report to the ilabs community for it generally made my pursuit of the Bachelor's degree worthwhile.

# Acknowledgement

I thank the Almighty God for giving me the courage, strength, wisdom, understanding, guidance, and the ability to successfully complete my final year project and report writing. I extend my sincere gratitude to my project supervisors Dr. Cosmas Mwikirize and Dr. Andrew Katumba for their tireless effort in guiding the direction of research and implementation of this project. I also thank Dembe Oscar and the ilabs community for the technical and non-technical support he rendered during the execution of the project. Special gratitude to my project partner Isreal Ssekitoleko with whom I closely worked all through. I am also very grateful to all my family members for motivating me in several ways all through my academic journey. Lastly, I appreciate the department of Electrical and Computer Engineering staff members for the guidance and knowledge they have equipped me with as it came in handy to see the fruition of the project.

# Abstract

Bone fractures are a leading cause of morbidity and mortality worldwide. In Uganda, statistics for the prevalence of bone fractures are unknown, although anecdotal evidence points to a high incidence, mostly arising out of traffic accidents and falls. The situation becomes worse year on year due to a rising life expectancy, and thus an increasing number of the aging population who are more prone to fractures. To reduce the debilitating effects of these fractures and improve quality of life, it is important that the fractures are accurately diagnosed early on. X-ray imaging is the most common imaging modality for fracture diagnosis in Uganda, but its manual interpretation is usually error-prone, potentially leading to missed diagnoses. To address this challenge, this project aimed at developing an automated bone fracture detection system for the efficient diagnosis, utilizing a deep learning approach. Images of fractured bones were obtained from Roboflow and the open-source dataset. We developed a model for localization of the bone fractures, utilizing the YOLOv5 architecture. Our best model achieved a mean average precision of 85.6%. Comparison with alternative approaches such as EfficientDet, and Detectron2 reveals the superior performance of our model. Our model, when integrated into a clinical decision support system, is potentially a promising approach to improve clinical outcomes based on accurate and efficient bone fracture detection from x-ray data.

# List of Acronyms

**AI** Artificial Intelligence

**ML** Machine Learning

**NN** Neural Network

**CT** Computerized Tomography

**ANN** Artificial Neural Networks

**DL** Deep Learning

**EDA** Exploratory Data Analysis

**GPU** Graphics Processing Unit

**API** Application Programming Interface

**SDG** Stochastic Gradient Decent

**GIOW** Generalised Intersection Over Union

**ROI** Regions of Interest

# Contents

|   |           |
|---|-----------|
| <b>Declaration</b>  | <b>i</b>  |
| <b>Approval</b>   | <b>ii</b> |
| <b>Dedication</b>   | <b>ii</b> |
| <b>Acknowledgement</b>                                      | <b>iv</b> |
| <b>Abstract</b>   | <b>v</b>  |
| <b>Acronyms</b>   | <b>v</b>  |
| <b>1 Introduction</b>                                       | <b>1</b>  |
| 1.1 Project Background . . . . .                            | 1         |
| 1.2 Problem Statement . . . . .                             | 4         |
| 1.3 Justification . . . . .                                 | 4         |
| 1.4 Project Objectives . . . . .                            | 4         |
| 1.5 Project Scope . . . . .                                 | 5         |
| 1.6 Summary of Report . . . . .                             | 5         |
| 1.6.1 Project Title . . . . .                               | 5         |
| 1.6.2 Abstract . . . . .                                    | 5         |
| 1.6.3 Introduction . . . . .                                | 5         |
| 1.6.4 Literature Review . . . . .                           | 6         |
| 1.6.5 Methodology . . . . .                                 | 6         |
| 1.6.6 Results and Discussion . . . . .                      | 6         |
| 1.6.7 Conclusions, Challenges and Recommendations . . . . . | 6         |
| <b>2 Literature Review</b>                                  | <b>7</b>  |
| 2.1 Introduction . . . . .                                  | 7         |
| 2.2 Bone Fractures . . . . .                                | 7         |
| 2.2.1 Types of Bone Fractures . . . . .                     | 8         |
| 2.2.2 Severity of Bone Fractures . . . . .                  | 9         |



|          |  |           |
|----------|--|-----------|
| 2.2.3    | Incidence of Bone Fractures . . . . .  | 10        |
| 2.3      | Image-based Diagnosis of Bone Fractures . . . . .  | 10        |
| 2.4      | Deep Learning . . . . .  | 14        |
| 2.4.1    | Types of Deep learning Algorithms . . . . .  | 15        |
| 2.5      | Applications of Deep Learning to Bone Fracture. . . . .  | 27        |
| 2.6      | Training and validation . . . . .  | 29        |
| 2.6.1    | Testing and Evaluation . . . . .   | 29        |
| 2.6.2    | Parameters and Hyper Parameters . . . . .  | 29        |
| 2.7      | Neural Networks . . . . .  | 31        |
| 2.7.1    | Deep Neural Networks . . . . .   | 31        |
| 2.7.2    | Transfer learning . . . . .  | 31        |
| 2.7.3    | Computer Vision and Object detection . . . . .   | 32        |
| 2.8      | Base Networks . . . . .  | 32        |
| 2.8.1    | Single Stage Detector . . . . .  | 35        |
| 2.8.2    | Two-stage Object Detectors . . . . .   | 36        |
| 2.9      | Related work . . . . .   | 39        |
| 2.9.1    | Convolutional Neural Networks For Automated Frac-<br>ture Detection and Localization on Wrist Radio-graphs | 39        |
| 2.9.2    | Fracture Detection in Wrist X-ray Images Using Deep<br>Learning-Based Object Detection Models . . . . .    | 43        |
| <b>3</b> | <b>Methodology</b>   | <b>49</b> |
| 3.1      | Introduction . . . . .   | 49        |
| 3.2      | Data Set Development . . . . .   | 49        |
| 3.2.1    | Obtaining Data-set . . . . .   | 49        |
| 3.3      | Exploratory Data Analysis . . . . .  | 50        |
| 3.3.1    | Random Sampling . . . . .  | 51        |
| 3.3.2    | Data-set Distribution . . . . .  | 51        |
| 3.3.3    | Anatomy of the Data-set . . . . .  | 52        |
| 3.3.4    | Data Preprocessing . . . . .   | 55        |
| 3.3.5    | Model Architecture . . . . .   | 56        |
| 3.3.6    | Model Training . . . . .   | 57        |
| 3.4      | Training results . . . . .   | 57        |
| 3.5      | Model Deployment . . . . .   | 60        |
| <b>4</b> | <b>Results and discussion</b>  | <b>63</b> |
| 4.1      | Introduction . . . . .   | 63        |
| 4.2      | Qualitative Results . . . . .  | 63        |
| 4.3      | Quantitative Results . . . . .   | 64        |
| 4.3.1    | Model Evaluation . . . . .   | 66        |
| 4.4      | Model Deployment Results . . . . .   | 67        |

|          |   |           |
|----------|---|-----------|
| 4.5      | Result Analysis and Discussion . . . . .          | 69        |
| <b>5</b> | <b>Conclusion, Challenges and Recommendations</b> | <b>70</b> |
| 5.1      | Conclusion . . . . .                              | 70        |
| 5.1.1    | Challenges . . . . .                              | 71        |
| 5.1.2    | Recommendations . . . . .                         | 71        |

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | Bone Fracture X-ray Image . . . . .                                | 8  |
| 2.2  | Types of Bone Fractures . . . . .                                  | 9  |
| 2.3  | An X-ray Bone Image . . . . .                                      | 12 |
| 2.4  | CT Scan Image . . . . .  | 13 |
| 2.5  | MRI Image . . . . .  | 14 |
| 2.6  | Artificial Neural Network . . . . .                                | 15 |
| 2.7  | Image Processing in a CNN . . . . .                                | 16 |
| 2.8  | LSTM Operation . . . . .   | 17 |
| 2.9  | Unfolded RNN . . . . .   | 18 |
| 2.10 | Operation of the GAN . . . . .                                     | 19 |
| 2.11 | Illustration of RBFN . . . . .                                     | 20 |
| 2.12 | Illustration of MLPs . . . . .                                     | 21 |
| 2.13 | Illustration of MLPs . . . . .                                     | 22 |
| 2.14 | Illustration of DBN . . . . .                                      | 23 |
| 2.15 | Illustration of RBMs . . . . .                                     | 25 |
| 2.16 | Illustration of Auto-encoders . . . . .                            | 26 |
| 2.17 | Illustration of Alex-net . . . . .                                 | 33 |
| 2.18 | Illustration of Google-net . . . . .                               | 34 |
| 2.19 | Illustration of VGG . . . . .                                      | 35 |
| 2.20 | Illustration of Yolo . . . . .                                     | 36 |
| 2.21 | Illustration of a Mask R-cnn . . . . .                             | 37 |
| 2.22 | Illustration of a Detectron 2 . . . . .                            | 38 |
| 2.23 | Illustration of a ROC per study . . . . .                          | 40 |
| 2.24 | Illustration of a Radio-graphs . . . . .                           | 41 |
| 2.25 | Illustration of a Radio-graphs . . . . .                           | 42 |
| 2.26 | Illustration of Training loss of Validation at an AP of 50 . . . . | 44 |
| 2.27 | Illustration of Training loss of validation at an AP of 50 . . . . | 45 |
| 2.28 | Test AP at 50 of Detection models . . . . .                        | 45 |
| 2.29 | Precision and Recall curve . . . . .                               | 46 |
| 2.30 | Precision and Recall curve . . . . .                               | 47 |
| 2.31 | Comparison of various Wrist Test Data-sets . . . . .               | 47 |

|      |  |    |
|------|--|----|
| 3.1  | Images from the Dataset . . . . .  | 50 |
| 3.2  | Sampling Table for the EDA . . . . .   | 51 |
| 3.3  | Data-set distribution . . . . .  | 52 |
| 3.4  | Bones of Wrist and Hand . . . . .  | 53 |
| 3.5  | Clavicle Scapula and Humerus Image . . . . .   | 54 |
| 3.6  | Yolov5 Architecture . . . . .  | 56 |
| 3.7  | Table showing Data-set Split . . . . .   | 57 |
| 3.8  | Train Objectness Loss . . . . .  | 57 |
| 3.9  | Train Bounding Loss . . . . .  | 58 |
| 3.10 | Train Classification Loss . . . . .  | 58 |
| 3.11 | Validation Box Loss . . . . .  | 59 |
| 3.12 | Validation Objectness Losss . . . . .  | 59 |
| 3.13 | Validation Classification Loss . . . . .   | 60 |
| 3.14 | MVC Architecture of Django . . . . .   | 61 |
| 3.15 | Application Structure in Line with the MVC Architecture of<br>Django Framework . . . . . | 62 |
| 4.1  | Sample of the Model's Inference . . . . .  | 64 |
| 4.2  | Recall and Precision . . . . .   | 65 |
| 4.3  | Mean Average Precision at 0.5 . . . . .  | 66 |
| 4.4  | Model Results . . . . .  | 67 |
| 4.5  | Web page of login In . . . . .   | 67 |
| 4.6  | Web page for Uploading a Model . . . . .   | 68 |
| 4.7  | Web page for uploading an Image . . . . .  | 68 |
| 4.8  | Web page for Performing a Prediction . . . . .   | 69 |

# Chapter 1

## Introduction

### 1.1 Project Background

A bone fracture is a break or a crack in the bone. The main cause of bone fractures in Uganda is road traffic accidents. Other causes include; falls, repetitive motions, trauma, or direct blow to the body.[1]

Radiology is a vital health diagnostic tool, giving essential information for routine injury and disease prevention and evaluation. It uses a variety of imaging modalities, each with its own set of physical principles and levels of complexity.

Medical imaging is very important in bone fracture diagnosis since the naked eye cannot view the human body to determine the severity of the fracture. And different medical imaging techniques include:

- x-ray
- Magnetic Resonance Imaging
- Ultrasound
- Computer Tomography (CT) scans

However, in Uganda, the X-ray is one of the most commonly used, fast, and easily accessible diagnostic imaging tests. They're a type of radiation, where an X-ray beam is passed through the body where a portion of the X-rays are either absorbed or scattered by the internal structures, and the remaining X-ray pattern is transmitted to a detector (e.g., film or a computer screen) for recording or further processing by a computer.[2]

Radiologists in Uganda use their naked eyes to identify the anatomical bone and also look for fractures in x-ray images. As a result of user-dependent variation, this can result in missing some fractures, and misinterpretation of X-ray images, hence leading to ineffective patient management.[3]

To address limitations associated with human interpretation of medical images, AI has become popular. Artificial Intelligence (AI) is the replication of human intellect in machines that are programmed to think and act like humans. AI has an influence on almost every element of modern life, including entertainment, economics, and healthcare. The modern applications of AI within the healthcare industry include image processing disease analysis and diagnosis, development of drugs, patient monitoring, and surgery. These applications enable quick, cheaper, and reliable diagnosis health treatment, resulting in improved quality of life.

Machine learning is a form of AI that performs autonomous predictions by using algorithms that iteratively improve, or learn, in response to training data.[4]

Deep learning has proved successful in medical imaging, a sub-field of machine learning dealing with algorithms inspired by the structure and function of the brain called artificial neural networks. In other words, It mirrors the functioning of our brains. An example is to detect disease or abnormalities from X-ray images and classify them into several disease types or severity in radiology. While existing works focus on the classification of fractures this project focuses on bone fracture detection in x-ray images using deep learning. [4]

There has been recent work in bone fracture detection, a study led by author Rachel Kuo, from the Botnar Research Centre, Nuffield Department of Orthopaedics, Rheumatology and Musculoskeletal Sciences in Oxford, England. This proposed that AI can reduce the rate of early misdiagnosis in challenging circumstances in the emergency setting, including cases where patients may sustain multiple fractures. It has potential as an educational tool for junior clinicians. “It could also be helpful as a ‘second reader,’ providing clinicians with either reassurance that they have made the correct diagnosis or prompting them to take another look at the imaging before treating patients. [5]

Yang ling Ma and Yixin Lou proposed Bone fracture detection through the two-stage system of a Crack-Sensitive Convolutional Neural Network. In their paper, there proposed a new two-stage system to detect fractures. Firstly, there used a Faster Region with a Convolutional Neutral Network (Faster R-CNN) to detect 20 different types of bone regions in X-ray images, and then there recognized whether each bone region is fractured by using

a Crack-Net. The Faster R-CNN and crack-sensitive CNN were tested on small data set which limits the learning of the model.[6]

Thian Yee Liang, Li Yiting, and Jagmohan Pooja proposed Convolutional Neural Networks for Automated Fracture Detection and Localization on Wrist Radio-graphs. The purpose of their model was to demonstrate the feasibility and performance of an object detection convolutional neural network (CNN) for fracture detection and localization on wrist radio-graphs. But however, their system had some drawbacks like radius and ulna fractures and did not evaluate all potential fractures on a wrist radio-graph, such as carpal or metacarpal fractures, therefore there is uncertainty if the model would be able to perform adequately if there are limited training examples of certain classes. [7]

Hardalaç Firat, Faith Uysal, Ozan Peker, and Murat Ciceklidag proposed a Fracture Detection system in Wrist X-ray Images Using Deep Learning-Based Object Detection Models. The aim of their study was to perform fracture detection by use of deep-learning on wrist X-ray images to support physicians in the diagnosis of these fractures, particularly in the emergency service. After five different ensemble models were developed and then used to reform an ensemble model to develop a unique detection model, however, the system had some drawbacks which included Fracture labeling in other small bones ie (trapezoid, trapezium, scaphoid, capitate, hamate, triquetrum, pisiform, lunate) fractures in the Wrist were not studied and were ignored. [8]

Bin Guana,Guoshan Zhanga, Jinkun Yao, and Mengxuan Wang proposed an Arm fracture detection system in X-rays based on an improved deep convolutional neural network. This paper has three main which include three aspects. First, a new backbone network is established based on feature pyramid architecture to gain more fractural information. Second, an image preprocessing procedure including opening operation and pixel value transformation is developed to enhance the contrast of original images. Third, the receptive field adjustment containing anchor scale reduction and tiny ROIs expansion is exploited to find more fractures. However, this has a drawback of using the mura data-set images which are of low quality hence getting an average precision of 62.04% which is quite low. [9]

## 1.2 Problem Statement

A deep learning object detection system with localising capabilities, better accuracy in terms of mean average precision to solve the challenges in the current state of the art AI models like low mean average precision and also find application in developing a computer aided diagnosis system in a clinical setting to help solve observer variability among radiologists due to differences in knowledge and experience to reduce early misdiagnosis in an emergency setting.

## 1.3 Justification

In 2019 ministry of Health records indicate that Uganda had only 48 radiologists meant to serve almost 40 million Ugandans. This meant that one radiologist had to handle over 850,000 people with different complexities including bone fracture complications. Thus to save radiologists' time, improve efficiency, and as well reduce their workload upon them, an automatic bone detection system is required.

## 1.4 Project Objectives

### Main Objectives

- To develop an artificial intelligence-guided system that performs automatic bone fracture detection in X-ray images.

### Specific Objective

- To collect and curate a suitable open source data-sets for fractures in chest x-ray images.
- To develop a model for fracture detection.
- To integrate both models in a decision support system for automatic fracture detection.



## **1.5 Project Scope**

This project is aimed at developing an automatic bone fracture detection system using deep learning object detection that will be deployed on a web based application using the Django frame work.

## **1.6 Summary of Report**

### **1.6.1 Project Title**

Deep learning was the form of machine learning used in detection of fractures in xray images.

### **1.6.2 Abstract**

In this we looked at fractures being the major cause of mortality and to reduce the debilitating effects of these fractures an automated bone fracture detection system in x-ray images were needed.

### **1.6.3 Introduction**

- In the background we explicitly highlight the difficulty in bone fracture recognition in x-ray images looking at the anatomical regions where fractures are difficult to detect then introducing deep learning as a solution and also reviewing various papers related to the use of deep learning in x-ray bone fracture detection while giving their limitations.
- In the Problem Statement we looked at the challenges in the current AI models in regard to performance and clinical application.
- In the Justification we looked at the radiologists to patient ratio in Uganda giving an automatic bone fracture detection system as the solution.
- In the Project Objectives we give a description of our major goal of the project.
- In the scope we give the entire coverage of our project

#### **1.6.4 Literature Review**

In this section, we give a description of bone fractures in terms of types, and incidence severity and also look at the image-based diagnosis of bone fractures then lastly review related literature to our project.

#### **1.6.5 Methodology**

In this section, we describe in detail the entire processing of development, training, and deployment of the deep learning model for the detection of fractures in x-ray images. The process involved the use of different tools and software such as Anaconda, and Google co-lab among others as will be elaborated.

#### **1.6.6 Results and Discussion**

In this section we look at the results we obtained after model training and deployment using qualitative and quantitative analysis.

#### **1.6.7 Conclusions, Challenges and Recommendations**

In this section, we give where our developed system can be applied through conclusion, the various challenges encountered during the project, and how our project can be improved in future works.

# Chapter 2

## Literature Review

### 2.1 Introduction

This chapter looks at the types of bone fractures, incidence, imaging diagnostics, and different research papers relating to our project.

### 2.2 Bone Fractures

A bone fracture is a medical definition of a broken bone. Fractures are usually caused by traumas like falls, car accidents, or sports injuries. But some medical conditions and repetitive forces (like running) can increase your risk of experiencing certain types of fractures. A bone fracture happens when something hits your bone with enough force not only to damage it but to break it in at least one place. Fractures are more serious injuries and can take much longer to heal.



Figure 2.1: Bone Fracture X-ray Image

### 2.2.1 Types of Bone Fractures

Bone fractures are categorized into types according to the following criteria i.e. pattern, cause and body part. The following are the common types of bone fractures which include.

- Stable Fracture. The broken ends of the bone line up and are barely out of place.
- Open (compound) Fracture. The skin may be pierced by the bone or by a blow that breaks the skin at the time of the fracture. The bone may or may not be visible in the wound.

- Transverse Fracture. This type of fracture has a horizontal fracture line.
- Oblique fracture. This type of fracture has an angled pattern.
- Com-minuted Fracture. In this type of fracture, the bone shatters into three or more pieces.
- Spiral Fracture: one part of the bone has been twisted at the breaking point.
- Com-minuted Fracture: the bone breaks into several pieces.
- Green-stick Fracture: an incomplete fracture in which the bone is bent; occurs most often in children.

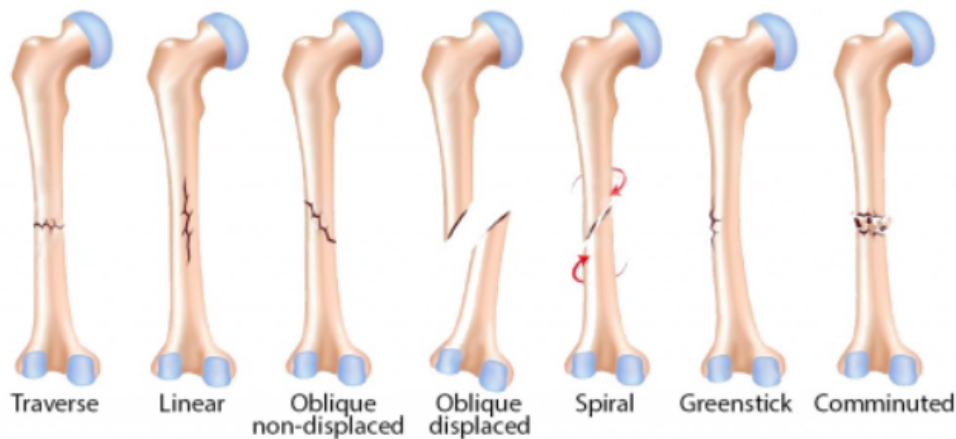


Figure 2.2: Types of Bone Fractures

[10]

### 2.2.2 Severity of Bone Fractures

The severity of a fracture usually depends on the force that caused the break. If the bone's breaking point has been exceeded only slightly, the bone may crack rather than break all the way through. If the force is extreme, such as that caused by an automobile crash or gunshot, the bone may shatter. Serious fractures can have dangerous complications if not treated promptly;

possible complications include damage to blood vessels or nerves and infection of the bone (osteomyelitis) or surrounding tissue. Recuperation time varies depending on the age and health of the patient and the type of fracture. A minor fracture in a child may heal within a few weeks; a serious fracture in an older person may take months to heal.

### **2.2.3 Incidence of Bone Fractures**

Incidence refers to the proportion or rate of persons who develop a condition during a particular time period. A study by Nienke P Dosa in a New-York referral hospital found that Two hundred twenty-one consecutive patients aged 2–58 years were evaluated in 2003 at a regional referral center. Twenty percent were children aged 2–10 years; 30% were adolescents aged 11–18 years, and 50% were adults aged 19– 58 years. The annual incidence of fractures among children, adolescents, and adults was 23/1000; 29/1000; and 18/1000, respectively. The crude fracture rate was 9/1000 per patient-year. Five fractures were reported during the study year, Most fractures involved the tibia and femur. Fractures were more common in children than adults. The median age at first fracture was 11 years, though were also present in adults, and in adults, there were typically attributed to accidental falls (commonly in the bathroom) or during exercise.

[11]

According to a study by Isaac Kajja in Entebbe Hospital, it was established that from 1st November 2014 to 28th February 2015 with a sample space of 101 patients included 72 (71.3%) males and 29 (28.7%) females with ages ranging from three to 79 years. Injuries were mainly caused by road traffic accidents affecting 36 (35.6%) patients. 95 (95.0%) patients presented with fractures (91 as fractures alone and four as fractures with a dislocation) while six (6.0%) presented with dislocations (three lower limb and three upper limb dislocations). The majority of patients ( 78.95%) with fractures had simple fractures. Up to 94 out of the 95 patients had limb fractures – 52 upper limbs and 42 lower limbs. Only one spine fracture was recruited. [12]

## **2.3 Image-based Diagnosis of Bone Fractures**

Diagnostic imaging techniques help narrow the causes of an injury or illness and ensure that the diagnosis is accurate. These techniques include x-rays, computed tomography (CT) scans, and magnetic resonance imaging (MRI). These imaging tools let your doctor "see" inside your body to get a "picture" of your bones, organs, muscles, tendons, nerves, and cartilage. This is a way

the doctor can determine if there are any abnormalities.

- X-rays (radio-graphs) are the most common and widely available diagnostic imaging technique. Even if you also need more sophisticated tests, you will probably get an x-ray first. The level of radiation exposure from x-rays is not harmful, but your doctor will take special precautions if you are pregnant. Bones, calcifications, some tumors, and other dense matter appear white or light because they absorb the radiation. Less dense soft tissues and breaks in bones let radiation pass through, making these parts look darker on the x-ray film. Most times patients will probably be x-ray-ed from several angles. If you have a fracture in one limb, your doctor may want a comparison x-ray of your uninjured limb. X-rays may not show as much detail as an image produced with more sophisticated techniques. They are, however, the most common imaging tool used to evaluate an orthopedic problem and are readily available in most doctors' offices.





- Computed Tomography (CT) is an imaging tool that combines x-rays with computer technology to produce a more detailed, cross-sectional image of your body. A CT scan lets your doctor see the size, shape, and position of structures that are deep inside your body, such as organs, tissues, or tumors. You may need a CT scan if you have a problem with a small, bony structure or if you have severe trauma to the brain, spinal cord, chest, abdomen, or pelvis. Sometimes, you may be given a dye or contrast material to make certain parts of your body show up better. A CT scan costs more and takes more time than a regular x-ray. It can be done in either a hospital setting or an outpatient imaging center.

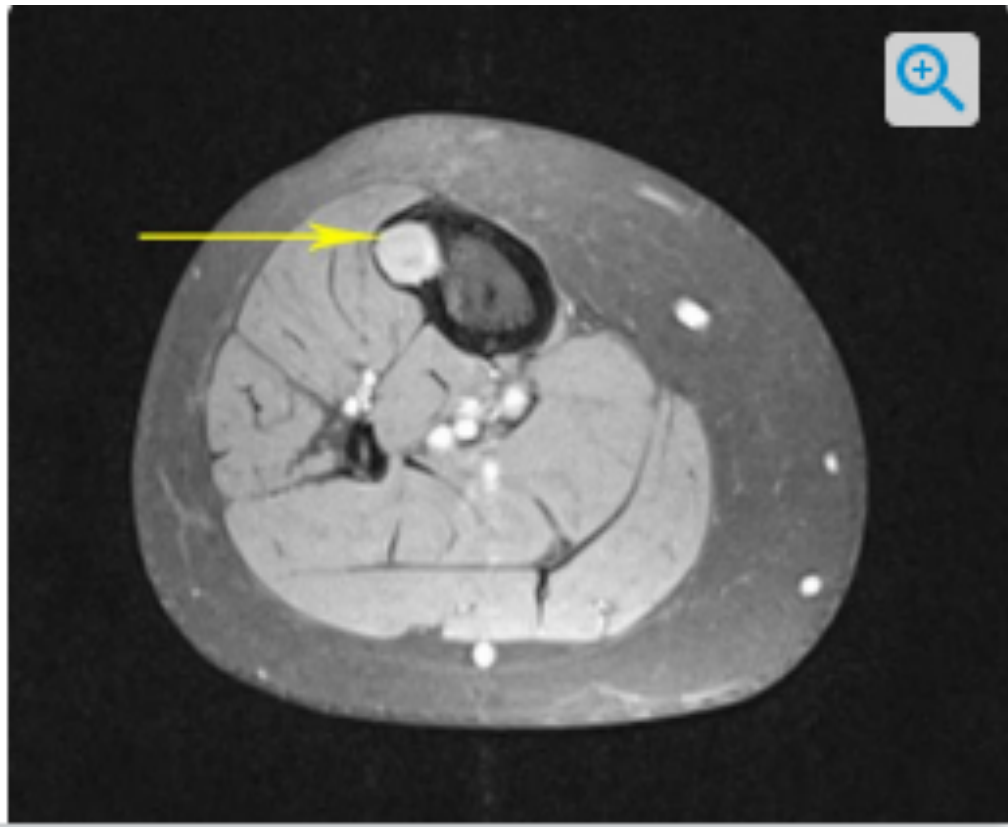


Figure 2.4: CT Scan Image

- Magnetic Resonance Imaging (MRI) is another diagnostic imaging technique that produces cross-sectional images of your body. Unlike CT scans, MRI works without radiation. The MRI tool uses magnetic fields and a sophisticated computer to take high-resolution pictures of your bones and soft tissues. The MRI creates a magnetic field around

you and then pulses radio waves to the area of your body to be pictured. The radio waves cause your tissues to resonate. A computer records the rate at which your body's various parts (tendons, ligaments, nerves, etc.) give off these vibrations, and translates the data into a detailed, two-dimensional picture. An MRI may be used to help diagnose torn knee ligaments and cartilage, torn rotator cuffs, herniated disks, osteonecrosis, bone tumors, and other problems.

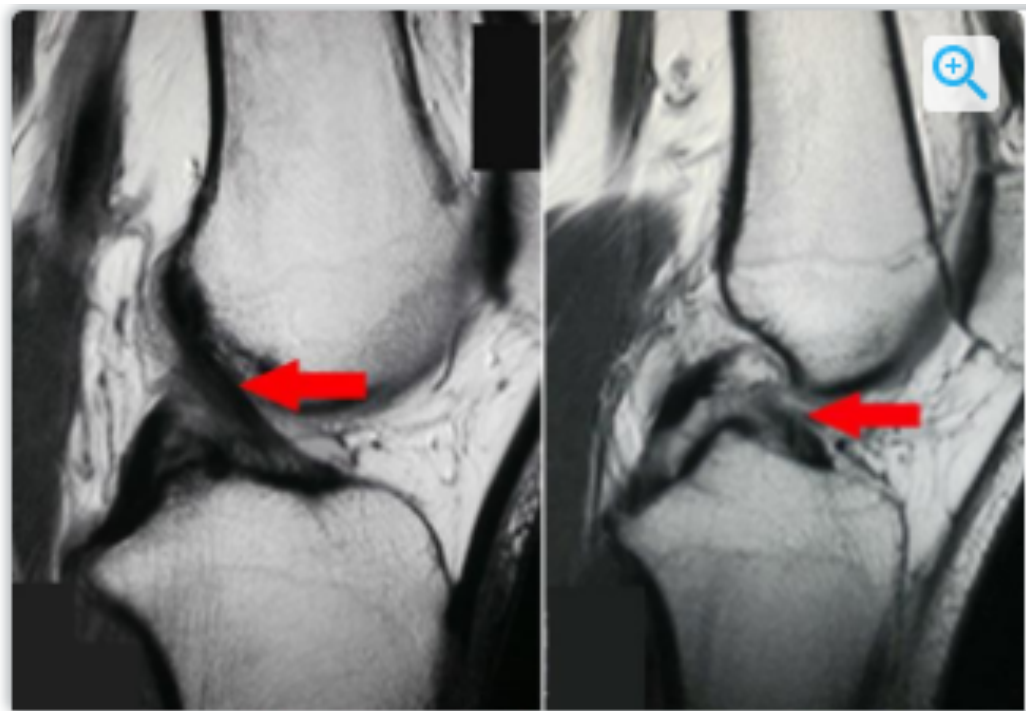


Figure 2.5: MRI Image

- Other Imaging Studies include Ultrasound uses high-frequency sound waves that echo off the body. It is painless and noninvasive and does not require radiation A bone scan which uses a small amount of radioactive material to identify areas of increased bone activity.

[9]

## 2.4 Deep Learning

Deep learning can be considered a subset of machine learning. It is a field that is based on learning and improving on its own by examining computer

algorithms.

Deep learning works with artificial neural networks, which are designed to imitate how humans think and learn.

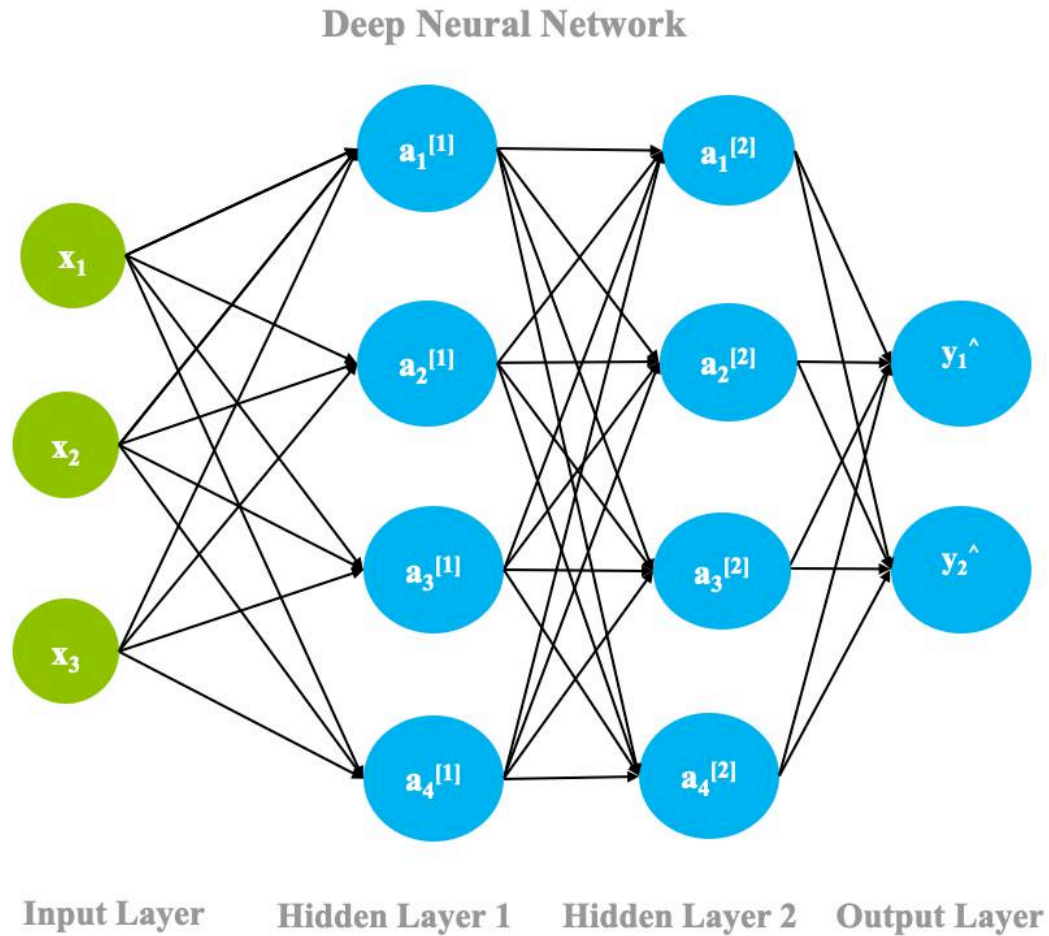


Figure 2.6: Artificial Neural Network

### 2.4.1 Types of Deep learning Algorithms

- Convolutional Neural Networks (CNNs) CNN's, also known as ConvNets, consist of multiple layers and are mainly used for image processing and object detection. Yann LeCun developed the first CNN in 1988 when it was called LeNet. It was used for recognizing characters like ZIP codes and digits.

CNN's have multiple layers that process and extract features from data. The CNN has the following layers

- Convolution Layer. CNN has a convolution layer that has several filters to perform the convolution operation.
- Rectified Linear Unit (Re-LU) CNN's have a Re-LU layer to perform operations on elements. The output is a rectified feature map.
- Pooling Layer
  - \* The rectified feature map next feeds into a pooling layer. Pooling is a down-sampling operation that reduces the dimensions of the feature map.
  - \* The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into a single, long, continuous, linear vector by flattening it.
- Fully Connected Layer A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the images.

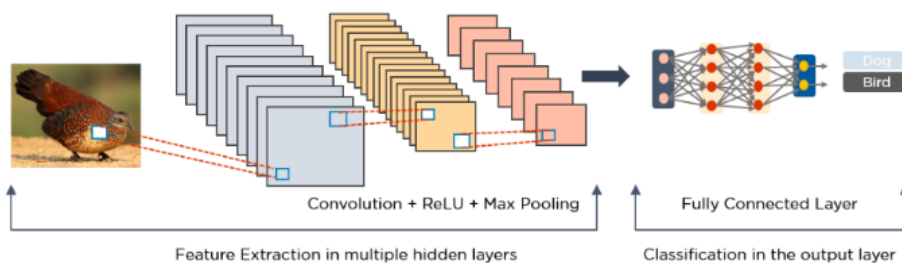


Figure 2.7: Image Processing in a CNN

- Long Short Term Memory Networks (LSTMs) LSTMs are a type of Recurrent Neural Network (RNN) that can learn and memorize long-term dependencies. Recalling past information for long periods is the

default behavior.

How does the LSTMS work?

- First, they forget irrelevant parts of the previous state.
- Next, they selectively update the cell-state values.
- Finally, the output of certain parts of the cell state.

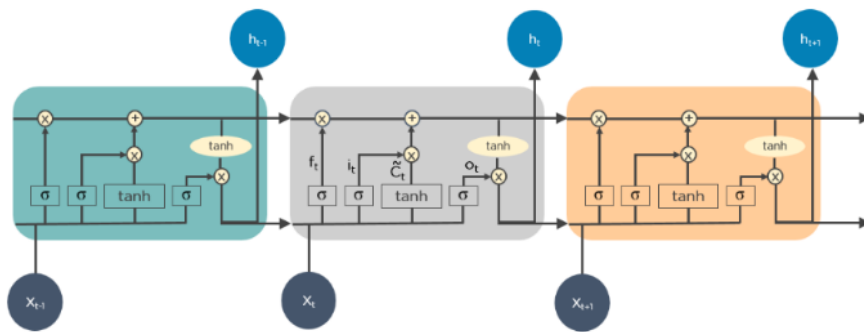


Figure 2.8: LSTM Operation

- Recurrent Neural Networks (RNNs) RNN has connections that form directed cycles, which allow the outputs from the LSTM to be fed as inputs to the current phase.

The output from the LSTM becomes an input to the current phase and can memorize previous inputs due to its internal memory. RNNs are commonly used for image captioning, time-series analysis, natural language processing, handwriting recognition, and machine translation. How do RNNs work

- The output at time  $t-1$  feeds into the input at time  $t$ .
- Similarly, the output at time  $t$  feeds into the input at time  $t+1$ .
- RNNs can process inputs of any length.

- The computation accounts for historical information, and the model size does not increase with the input size.

An unfolded RNN looks like this:

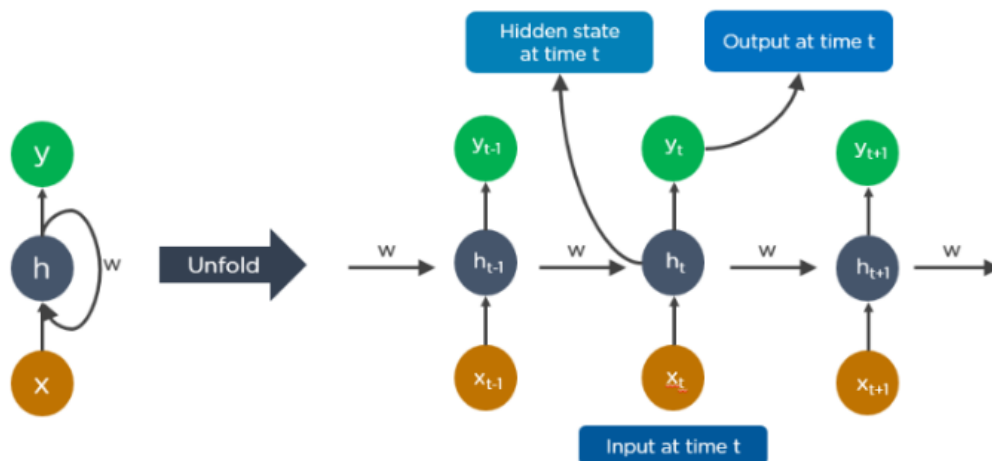


Figure 2.9: Unfolded RNN

- Generative Adversarial Networks (GANs) are generative deep learning algorithms that create new data instances that resemble the training data. GAN has two components: a generator, which learns to generate fake data, and a discriminator, which learns from that false information.

How Do GANs work?

- The discriminator learns to distinguish between the generator's fake data and the real sample data.
- During the initial training, the generator produces fake data, and the discriminator quickly learns to tell that it's false.
- The GAN sends the results to the generator and the discriminator to update the model.

Below is a diagram of how GANs operate

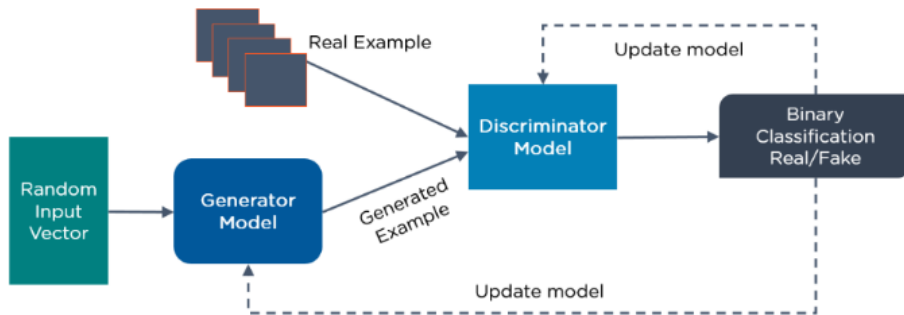


Figure 2.10: Operation of the GAN

- Radial Basis Function Networks (RBFNs)

RBFNs are special types of feed-forward neural networks that use radial basis functions as activation functions. They have an input layer, a hidden layer, and an output layer and are mostly used for classification, regression, and time-series prediction.

How Do RBFNs Work?

- RBFNs perform classification by measuring the input's similarity to examples from the training set.
  - RBFNs have an input vector that feeds to the input layer. They have a layer of RBF neurons.
  - The function finds the weighted sum of the inputs, and the output layer has one node per category or class of data.
  - The neurons in the hidden layer contain the Gaussian transfer functions, which have outputs that are inversely proportional to the distance from the neuron's center.
  - The network's output is a linear combination of the input's radial-basis functions and the neuron's parameters.
- example of an RBFN://

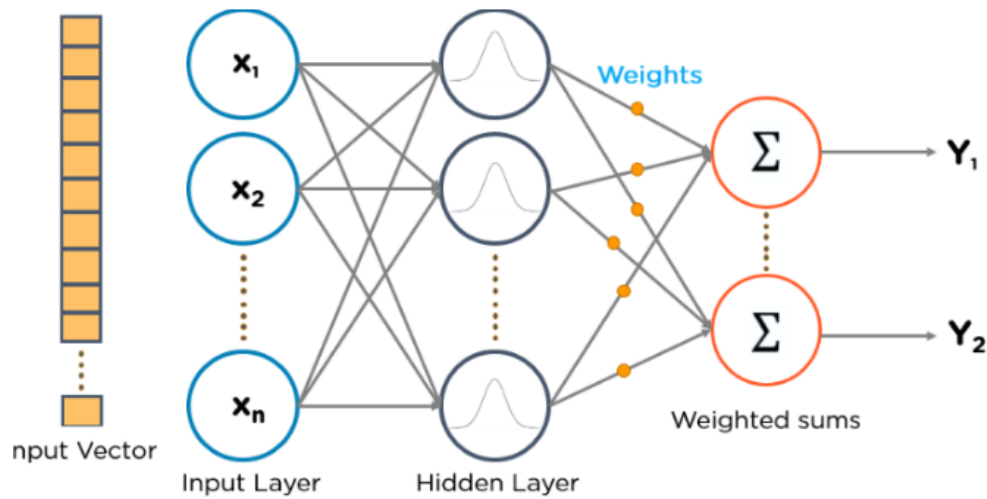


Figure 2.11: Illustration of RBFN

- Multi-layer Perceptrons (MLPs)
 

MLPs is an excellent place to start learning about deep learning technology.

MLPs belong to the class of feed-forward neural networks with multiple layers of perceptrons that have activation functions.

MLPs consist of an input layer and an output layer that is fully connected. They have the same number of input and output layers but may have multiple hidden layers and can be used to build speech-recognition, image-recognition, and machine-translation software.

How do the MLPs work

  - MLPs feed the data to the input layer of the network. The layers of neurons connect in a graph so that the signal passes in one direction.
  - MLPs compute the input with the weights that exist between the input layer and the hidden layers.
  - MLPs use activation functions to determine which nodes to fire. Activation functions include ReLUs, sigmoid functions, and tanh.
  - MLPs train the model to understand the correlation and learn the dependencies between the independent and the target vari-



ables from a training data set.  
Below is a diagram of MLP

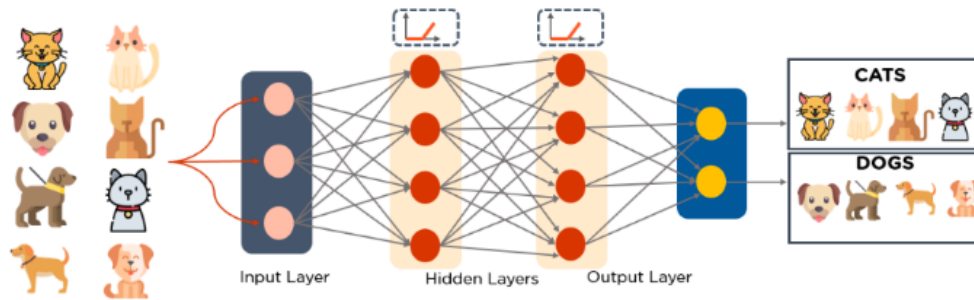


Figure 2.12: Illustration of MLPs

- Self Organizing Maps (SOMs)

Professor Teuvo Kohonen invented SOMs, which enable data visualization to reduce the dimensions of data through self-organizing artificial neural networks.

Data visualization attempts to solve the problem that humans cannot easily visualize high-dimensional data. SOMs are created to help users understand this high-dimensional information.

How Do SOMs Work?

- SOMs initialize weights for each node and choose a vector at random from the training data.
- SOMs examine every node to find which weights are the most likely input vector. The winning node is called the Best Matching Unit (BMU).
- SOMs discover the BMU's neighborhood, and the amount of neighbors lessens over time.
- SOMs award a winning weight to the sample vector. The closer a node is to a BMU, the more its weight changes

- The further the neighbor is from the BMU, the less it learns.  
SOMs repeat step two for N iterations  
Below is a block diagram of a SOM

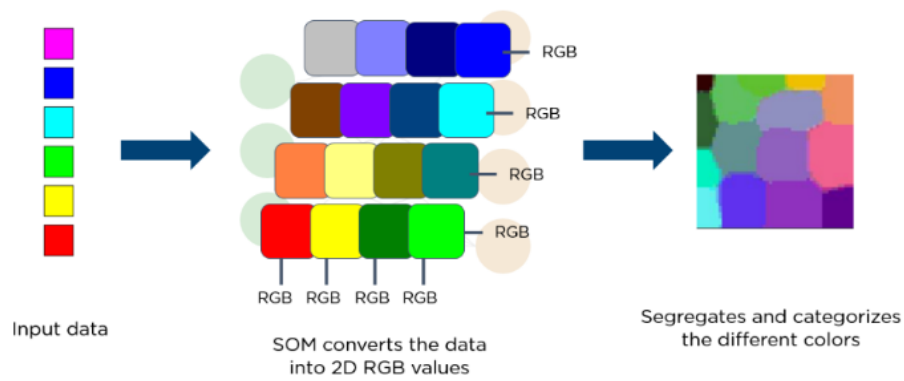


Figure 2.13: Illustration of MLPs

- Deep Belief Networks (DBNs)

DBNs are generative models that consist of multiple layers of stochastic, latent variables. The latent variables have binary values and are often called hidden units.

DBNs are a stack of Boltzmann Machines with connections between the layers, and each RBM layer communicates with both the previous and subsequent layers. Deep Belief Networks (DBNs) are used for image recognition, video recognition, and motion-capture data.

How Do DBNs Work?

- Greedy learning algorithms train DBNs. The greedy learning algorithm uses a layer-by-layer approach for learning the top-down, generative weights.//
  - DBNs run the steps of Gibbs sampling on the top two hidden layers. This stage draws a sample from the RBM defined by the top two hidden layers.//
  - DBNs draw a sample from the visible units using a single pass of ancestral sampling through the rest of the model.//
  - DBNs learn that the values of the latent variables in every layer can be inferred by a single, bottom-up pass.
- block diagram of DBN.

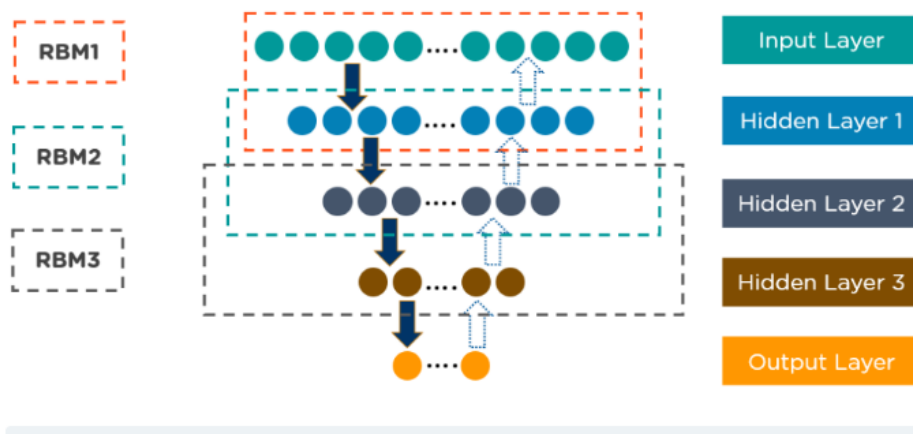


Figure 2.14: Illustration of DBN

- Restricted Boltzmann Machines (RBMs) Developed by Geoffrey Hinton, RBMs are stochastic neural networks that can learn from a prob-

ability distribution over a set of inputs.

This deep learning algorithm is used for dimensional reduction, classification, regression, collaborative filtering, feature learning, and topic modeling. RBMs constitute the building blocks of DBNs.

RBN has two layers which include visible units and hidden units.

Mode of operation of RBN.

- RBMs accepts the inputs and translate them into a set of numbers that encodes the inputs in the forward pass.
- RBMs combine every input with individual weight and one overall bias.
- In the backward pass, RBMs take that set of numbers and translate them to form the reconstructed inputs.
- RBMs combine each activation with individual weight and overall bias and pass the output to the visible layer for reconstruction.
- At the visible layer, the RBM compares the reconstruction with the original input to analyze the quality of the result.

Block diagram RBM.

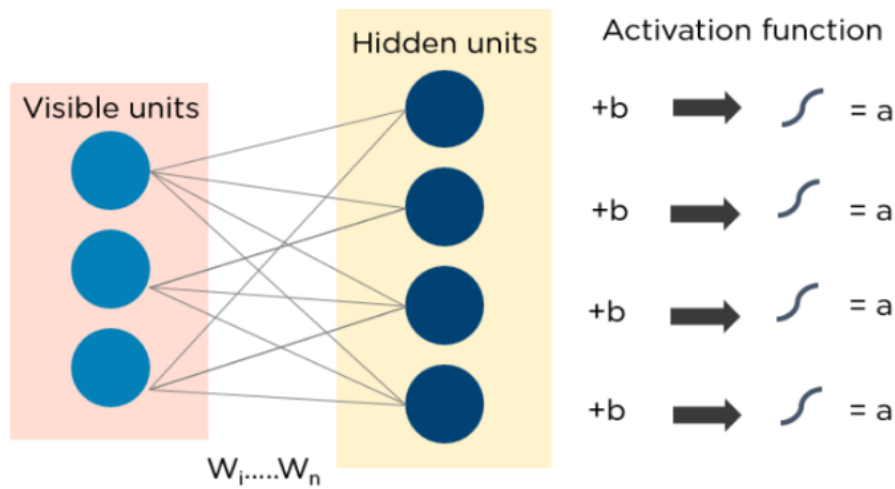


Figure 2.15: Illustration of RBMs

- Auto-encoders Auto-encoders are a specific type of feed-forward neural network in which the input and output are identical. Geoffrey Hinton designed auto-encoders in the 1980s to solve unsupervised learning problems. They are trained neural networks that replicate the data from the input layer to the output layer. Auto-encoders are used for purposes such as pharmaceutical discovery, popularity prediction, and image processing. Mode of operation Auto-encoder.

- Auto-encoders are structured to receive an input and transform it into a different representation. They then attempt to reconstruct the original input as accurately as possible.
- When an image of a digit is not clearly visible, it feeds to an auto-encoder neural network.
- Auto-encoders first encode the image, then reduce the size of the input into a smaller representation.
- Finally, the auto-encoder decodes the image to generate the reconstructed image.

Block diagram of Auto-Encoder

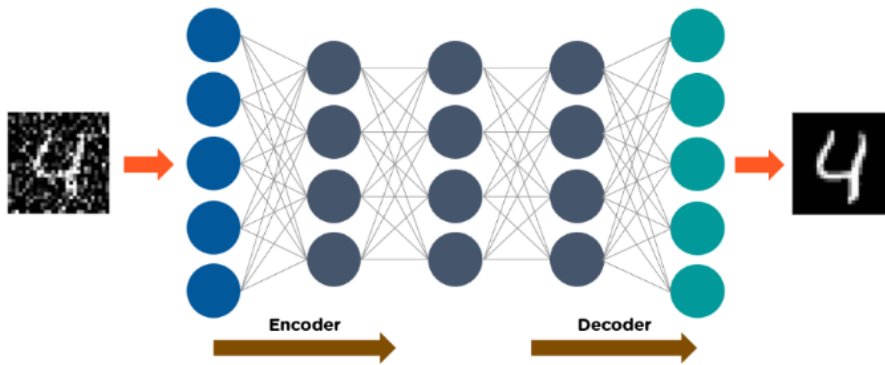


Figure 2.16: Illustration of Auto-encoders

[13]

## 2.5 Applications of Deep Learning to Bone Fracture.

- Applications in Diagnosis Screening osteoporosis. In the era of AI, many researchers have focused their attention on developing practical screening tools for osteoporosis. Easier-to-use and accurate diagnostic tools may improve the prognosis of individuals at high risk of fractures by earlier intervention and aid the effective use of public health resources for individuals at low risk. Taken together, increasing attempts have been made to diagnose osteoporosis using various data sources and ML methods, and performance has improved over time, especially when using images with CNN methods.

Screening fractures. Many studies have reported the application of ML in fracture detection and some of them have become the basis of commercially available programs—such as OsteoDetect (Imagen Technologies, New York, NY, USA; 2018). Several earlier studies used X-ray images to detect fractures, and studies using CT images to detect fractures have been increasing recently. As the basis for the OsteoDetect program, Lindsey et al, used wrist radio-graphs to detect wrist fractures using a CNN and showed performances in AUROC of 0.96 and 0.97 in two internal test data-sets. Also, they showed that when aided with the program, the misinterpretation rate of the average clinician was significantly reduced by 47.0%. Another interesting study conducted by Badgeley et al reported that imaging features from hip X-rays could be used to discriminate fractures using a CNN (AUROC of 0.78) and that patient data with hospital process variables, such as scanner model, scanner manufacturer, and order date showed better performance for fracture detection (AUROC of 0.91) than images. In a subgroup analysis of selected radio graphs matched with patient data and hospital process variables, X-rays could not detect hip fractures [45]. This result implied that the model detected fractures indirectly through the associated clinical variables rather than directly utilizing the image features of the fracture.

- Applications in Risk Prediction. As in other fields of medical research, accurate prediction of musculoskeletal outcomes enables an individualized approach for initiating and monitoring treatments. In terms of predicting fracture, most studies used a database to build prediction models. In men, Su et al, reported that the classification of a high-risk group for hip fractures using a classic ML method of classification and regression trees showed a discrimination power similar to that of FR-

AX.ML could be used to build prediction models and identify novel risk factors. Based on claims data of more than 280,000 individuals, Engels et al. [55] developed a hip fracture prediction model with an AUROC of 0.65 to 0.70 using a super-learner algorithm that considered both regression and ML algorithms, such as support vector machines and RUSBoost.

Moreover, considering the sequential characteristics of electronic health records, Almog et al, developed a short-term incident fracture prediction model based on natural language processing methods. These findings indicate the possibility of using the unique medical history data of the patients over time to predict the risk of fractures. Contrarily, studies using unsupervised learning to identify fractures were also conducted. Kruse et al, found nine different fracture risk clusters based on BMD, clinical risk factors, and medications using simple unsupervised hierarchical agglomerative clustering analysis.

With regard to predicting outcomes other than fracture, few studies have attempted to predict bone loss and falls. The rate of bone loss over 10 years could be predicted better with the artificial neural network than with multiple regression analysis using conventional parameters, such as age, body mass index, menopause, fat and lean body mass, and BMD values [60]. Falls were also accurately predicted using XGBoost, reporting the following top predictors: cognitive disorders, abnormalities of gait and balance, and Parkinson’s disease. The most common problem encountered in learning tasks is a class imbalance because of the low incidence of positive events.

- **Future Applications.** Overall, many studies have consistently shown that ML models can detect fractures better than clinicians, expanding the limits of human performance. Recently, FDA and the Korean FDA and approved some fracture detection algorithms to support clinicians, which makes AI-guided tools within reach. However, AI models exceedingly better than conventional models have not been suggested for the task of predicting fractures. One of the main reasons for the phenomenon could be that the conventional models are well-designed and already have excellent performances in fracture risk prediction, which leaves small room for improvement. Therefore, more AI models combining images of bone and muscle with clinical information are needed in the near future. It could be considered in designing the models whether input images can provide high-quality information to predict fractures, as there is a significant difference in the quality and amount of included information included depending on the image type.



In this era of the overwhelming volume of medical data, AI is a promising tool that may shed light on an individualized approach and a better understanding of the disease in the field of bone and mineral research. The present review aimed to provide an overview of the latest studies using ML to address the issues in the field, focusing on osteoporosis and fragility fractures. ML models for diagnosing and classifying osteoporosis and detecting fractures from images have shown promising performance and have improved over time. Fracture risk prediction is another promising field of research, and studies are being conducted using various data sources.

## **2.6 Training and validation**

Deep learning training is when a deep neural network (DNN) “learns” how to analyze a predetermined set of data and make predictions about what it means. It involves a lot of trial and error until the network is able to accurately draw conclusions based on the desired outcome.

Validation in deep learning is a process where a trained model is assessed with a testing data set. Training and validation intend to locate an ideal model with the best execution.

[13]

### **2.6.1 Testing and Evaluation**

Testing in deep learning refers to the process where the performance of a fully trained model is evaluated on a testing set. The testing set consisting of a set of testing samples should be separated from the both training and validation sets.

[13] Inference: is the process of using a trained dense neural network model to make predictions against previously unseen data,

### **2.6.2 Parameters and Hyper Parameters**

In ML/DL, a model is defined or represented by the model parameters. However, the process of training a model involves choosing the optimal hyper-parameters that the learning algorithm will use to learn the optimal parameters that correctly map the input features (independent variables) to the labels or targets (dependent variable) such that you achieve some form of intelligence. Hyper-parameters are parameters whose values control the

learning process and determine the values of model parameters that a learning algorithm ends up learning. Hyper-parameters are used by the learning algorithm when it is learning but they are not part of the resulting model. Examples of hyper-parameters include

- Train-Test split ratio
- Learning rate in optimization algorithms (e.g. gradient descent)
- Choice of optimization algorithm (e.g., gradient descent, stochastic gradient descent, or Adam optimizer)
- Choice of activation function in a neural network (nn) layer (e.g. Sigmoid, Re-LU, Tanh)
- The choice of cost or loss function the model will use
- Number of hidden layers in a nn
- Number of activation units in each layer
- The drop-out rate in nn (dropout probability)
- Number of iterations (epochs) in training a nn
- Number of clusters in a clustering task
- Kernel or filter size in convolutional layers
- Pooling size
- Batch size

[13] Parameters on the other hand are internal to the model. That is, they are learned or estimated purely from the data during training as the algorithm used tries to learn the mapping between the input features and the labels or targets. At the end of the learning process, model parameters are what constitute the model itself. Examples of parameters include

- The coefficients (or weights) of linear and logistic regression models.
- Weights and biases of an nn
- The cluster centroids in clustering

## 2.7 Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

### 2.7.1 Deep Neural Networks

Deep Learning is a sub-field of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization.

Forward propagation: The input data is fed in the forward direction through the network. Each hidden layer accepts the input data, processes it as per the activation function, and passes it to the successive layer.

Back-propagation: Back-propagation computes the gradient of the loss function with respect to the weights of the network. Back-propagation is one of the important algorithms for training the feed-forward network. Once we have passed through the forwarding network, we get the predicted output to compare with the target output. Based on this, we understood that we can calculate the total loss and say whether the model is good to go or not.

### 2.7.2 Transfer learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

It is a popular approach in deep learning where pre-trained models are used as the starting point for computer vision and natural language processing tasks given the vast computing and time resources required to develop neural network models for these problems and from the huge jumps in a skill that they provide on related problems.

The transfer learning used in deep learning is Inductive Transfer learning where the source and target domains are the same, yet the source and target tasks are different from each other. The algorithms try to utilize the inductive biases of the source domain to help improve the target task. Depending upon whether the source domain contains labeled data or not, this can be further divided into two subcategories, similar to multitask learning and self-

taught learning, respectively.

### 2.7.3 Computer Vision and Object detection

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs — and take actions or make recommendations based on that information. Computer vision is the broad parent name for any computations involving visual content – that means images, videos, icons, and anything else with pixels involved.

Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results.

## 2.8 Base Networks

After we input the image into the architecture, the first component we come across is the base network. The base network is typically a CNN pre-trained for a particular classification task. This CNN will be used for transfer learning, in particular, feature extraction. The base network is used to extract features from the input image. And include the following.

- Alex-net: The name given to a Convolutional Neural Network Architecture that won the LSVRC competition in 2012. The Alex-Net contains 8 layers with weights; 5 convolution layers and 3 fully connected layers. At the end of each layer, Re-Lu activation is performed except for the last one, which outputs with a soft-max with a distribution over the 1000 class labels.  
Architecture



- Google Net is a 22-layer deep convolutional neural network that's a variant of the Inception Network, a Deep Convolutional Neural Network developed by researchers at Google.

The Google Net architecture presented in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14) solved computer vision tasks such as image classification and object detection.

Architecture



Figure 2.18: Illustration of Google-net

- VGGNET

VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The “deep” refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers. The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures

The VGG network is constructed with very small convolutional filters, the input takes in an input size image of  $224 \times 224$  VGG Architecture

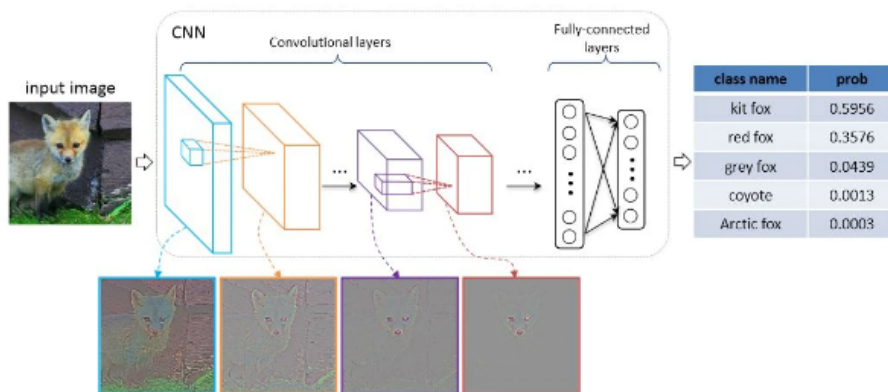


Figure 2.19: Illustration of VGG

### 2.8.1 Single Stage Detector

A single-stage detector removes the RoI extraction process and directly classifies and regresses the candidate anchor boxes. Examples are the YOLO family (YOLOv2, YOLOv3, YOLOv4, and YOLOv5) Corner-Net, Center-Net, and others.

YOLO is an object detection architecture simply called YOU ONLY LOOK ONCE. This involves the use of a single neural network trained end to end to take in a photograph as input and predicts bounding boxes and class labels for each bounding box directly. YOLO is a typical single-stage detector.

It consists of mainly three types of layers: Convolutional, Maxpool, and Fully Connected. The YOLO network has 24 convolutional layers, which do the image feature extraction followed by two fully connected layers for predicting the bounding box coordinates and classification scores. The architecture of Yolo

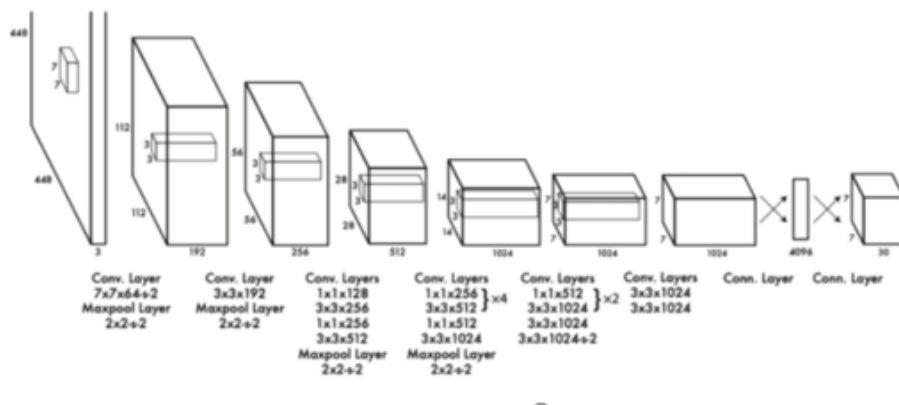


Figure 2.20: Illustration of Yolo

## 2.8.2 Two-stage Object Detectors

Two-stage detectors divide the object detection task into two stages: extract ROIs (Region of interest), then classify and regress the ROIs. Examples of object detection architectures that are 2-stage oriented include R-CNN, Fast-RCNN, Faster-RCNN, Mask-RCNN, and others. Let's take a look at the Mask R-CNN for instance.

### Mask R-CNN

The Mask R-CNN is a typical Object Instance Segmentation technique for object detection. This architecture is an extension of Faster R-CNN by adding a branch for predicting segmentation masks on each RoI, in parallel with the existing branch for classification and bounding box regression. The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to-pixel manner. Below is an architectural demonstration of Mask R-CNN.

The architecture of mask r-cnn



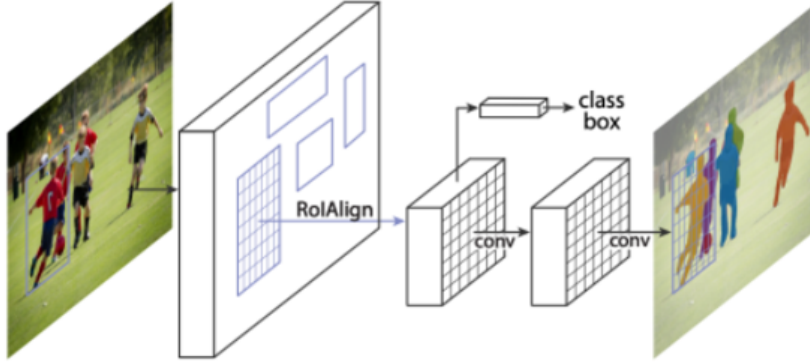


Figure 2.21: Illustration of a Mask R-cnn

Detectron2 is a popular PyTorch-based modular computer vision model library. It is the second iteration of Detectron, originally written in Caffe2. The Detectron2 system allows you to plug in custom state-of-the-art computer vision technologies into your workflow.

Detectron2 includes all the models that were available in the original Detectron, such as Faster R-CNN, Mask R-CNN, RetinaNet, and DensePose. It also features several new models, including Cascade R-CNN, Panoptic FPN, and TensorMask.

The Architecture of Detectron 2

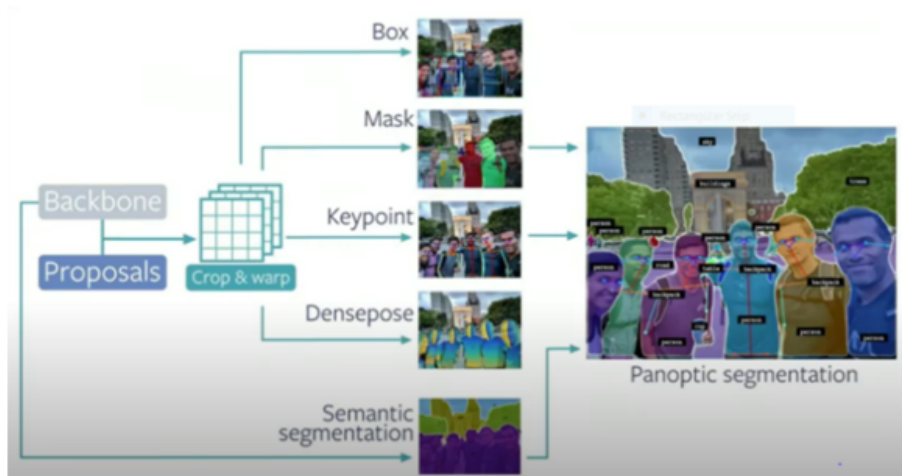


Figure 2.22: Illustration of a Detectron 2

[8]

## 2.9 Related work

### 2.9.1 Convolutional Neural Networks For Automated Fracture Detection and Localization on Wrist Radio-graphs

This project had two deliverables

- A data set was gathered of 7356 wrist radio-graphic studies extracted from a hospital picture archiving and communication system of these there were 245 fracture images. Of the 245, there were 244 lateral images with fractures, consisting of 243 images with one fracture mark and one image with, also there were 188 fractures on the frontal view and 97 fractures on the lateral view. A total of 548 images (268 frontal, 280 lateral) had no marks and for purposes of this study were considered normal.
- To demonstrate the feasibility and performance of an object detection convolutional neural network (CNN) for fracture detection and localization on wrist radio-graphs. A deep learning object detection network detected and localized radius and ulna fractures on wrist radio-graphs with high sensitivity at a per-fracture (frontal 91.2%, lateral 96.3%), per-image (frontal 95.7%, lateral 96.7%), and per-study (98.1%) level, even with a relatively modest training data-set size of 7356 radio-graphic studies.

#### Results

To demonstrate the Feasibility and Performance of an Object detection convolutional neural network (CNN) for fracture detection and localization on wrist radio-graphs. A deep learning object detection network detected and localized radius and ulna fractures on wrist radio-graphs with high sensitivity at a per-fracture (frontal 91.2%, lateral 96.3%), per-image (frontal 95.7%, lateral 96.7%), and per-study (98.1%) level, even with a relatively modest training data-set size of 7356 radio-graphic studies.

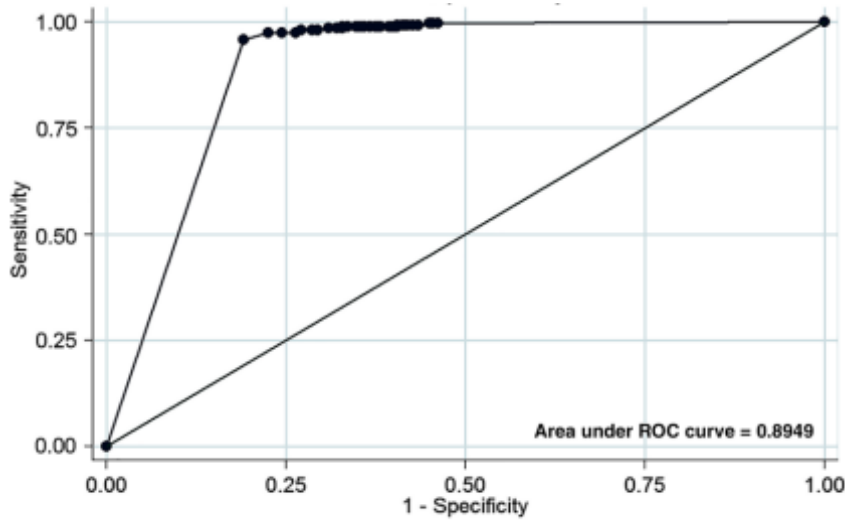


Figure 2.23: Illustration of a ROC per study

Data points represent empirical operating points based on a cutoff value of the convolutional neural network confidence score. There used a more homogeneous set of wrist radio-graphs only for the training which made conversion easy for the CNN, In all our training data were manually checked and annotated by radiologists. This is more time-consuming but provides more accurate data labeling.

The use of bounding boxes to indicate the location of abnormality helps to refine the training of the network to detect image features that are pertinent to the problem at hand.

There used a state-of-the-art object detection network (Inception-ResNet version 2 with Faster R-CNN), which may be more efficient at extracting relevant features from the training data compared with the older VGGNet and Inception networks.

Radio-graphs of selected true-positive examples of radius and ulna fracture

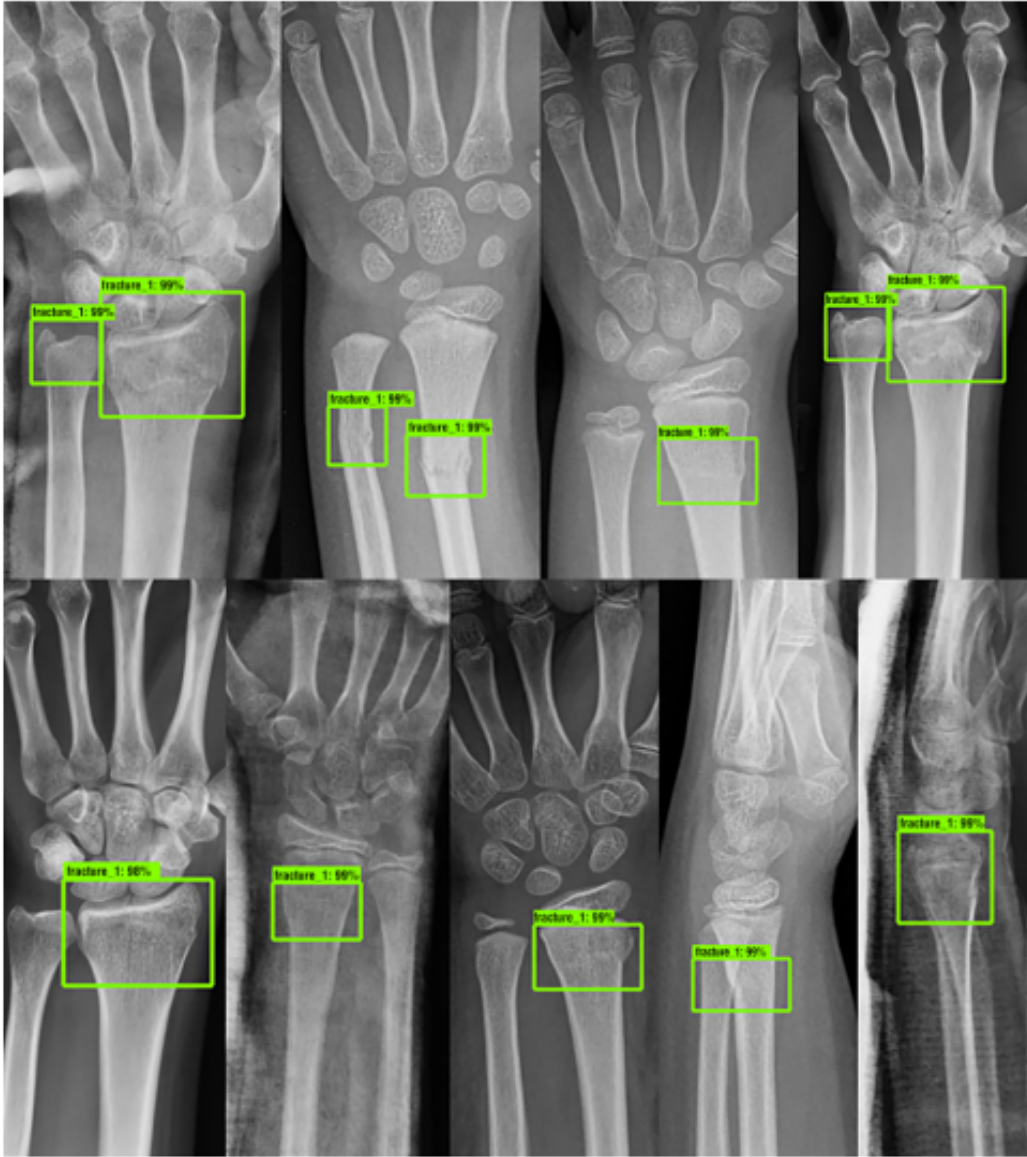


Figure 2.24: Illustration of a Radio-graphs

Green boxes are marks made by the Faster R-convolutional neural network deep learning network trained to detect and localize fractures. Percentages given for each mark reflect the confidence score by the network of a fracture located within the marked box.

Radio-graphs show selected false-positive examples

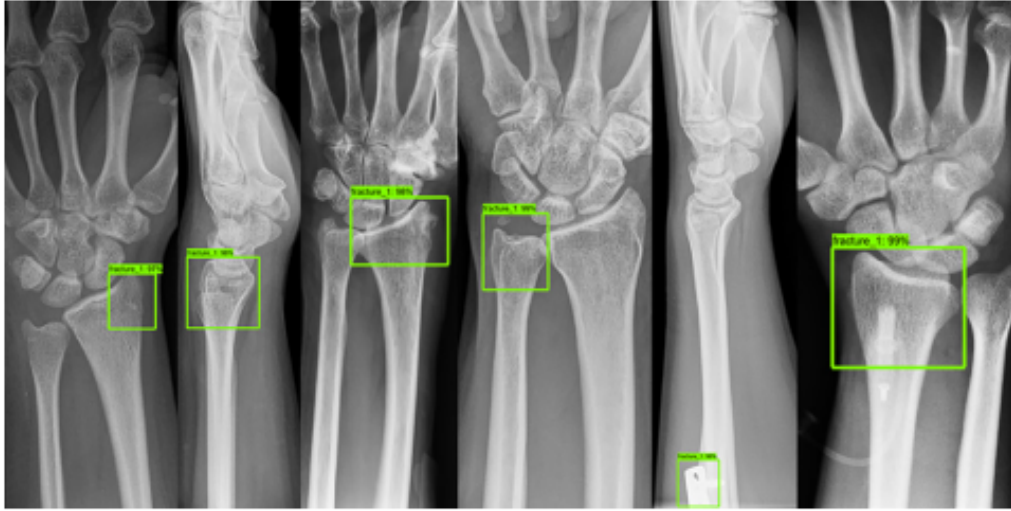


Figure 2.25: Illustration of a Radio-graphs

Old fractures and artifacts on the image were a common cause of false-positive marks (green boxes) made by the trained networks. Percentages given for each mark reflect the confidence score by the network of a fracture located within the marked box.// Drawbacks of the deep learning network

- There only included radius and ulna fractures and did not evaluate all potential fractures on a wrist radio-graph, such as carpal or metacarpal fractures. This is because radius and ulna fractures are much more prevalent, and thus obtaining sufficient training examples for deep learning was feasible, there was uncertainty if the model would perform with limited class examples.
- There only tested our model on emergency department wrist radio-graphs, and our results may not generalize to other settings such as orthopedic outpatient radio-graphs and excluded training and testing with orthopedic outpatient radio-graphs because of the large proportion of metallic implants in routine orthopedic outpatient radio-graphs of the wrist. Including such radio-graphs may unintentionally teach the CNN to associate the presence of metallic implant with the presence of a fracture, rather than discriminate features of the fracture per se.

[7]

### 2.9.2 Fracture Detection in Wrist X-ray Images Using Deep Learning-Based Object Detection Models

In this study the aim was to perform fracture detection by use of deep-learning on wrist X-ray images to support physicians in the diagnosis of these fractures, particularly in the emergency Services, this project had 4 deliverables.

- The wrist X-ray images collected from Gazi University Hospital were used within the scope of the study, physicians gave assistance in data collection and labeling of the x-ray images, and the 542 images collected from the hospital were in Di-com format There is a heterogeneous distribution of both right-wrist and left-wrist images in the data-set from 275 patients.
- In order to use the collected data in CNN-based object detection models, the format was converted from DI-COM format to 3-channel png format, Pydicom library was used to read and extract information from images taken in DI-COM format. Images in DI-COM format were converted to gray-scale PNG format using the library. After certain operations were made on the images in PNG format, normalization was made within the framework in the object detection network, and training and testing were carried out by converting them to RGB format in order to perform operations such as coloring on the image in future studies.
- All of the wrist images taken from Gazi University Hospital consist of fracture (abnormal, unhealthy, positive) images. Therefore, there is at least one fracture in each of the images in the data-sets. There are a total of 569 fractures in 542 wrist images used within the scope of the study. The distribution of the 569 fractures in the data-set is as follows: there are 459 labels in 434 training data, 55 labels in 54 validation data, and 56 labels in 54 test data.
- To develop the most compatible model for performing fracture detection in wrist X-ray images.
- To develop in the future a portable x-ray tool for diagnosing fractures.

#### Results

For fracture detection in wrist X-ray images, a total of 20 fracture detection procedures were performed, with and without augmentation, in 10 different

deep learning-based models.

Training loss and epoch results for the highest validation AP50 in detection models

| Models            | Without Augmentation |        |     |            | With Augmentation |        |     |            |
|-------------------|----------------------|--------|-----|------------|-------------------|--------|-----|------------|
|                   | TB_Loss              | T_Loss | TT  | Best Epoch | TB_Loss           | T_Loss | TT  | Best Epoch |
| DCN Faster R-CNN  | 0.0938               | 0.1821 | 174 | 6          | 0.0969            | 0.1839 | 174 | 5          |
| Dynamic R-CNN     | 0.2978               | 0.5033 | 139 | 8          | 0.2664            | 0.4605 | 156 | 10         |
| Faster R-CNN      | 0.0873               | 0.1573 | 126 | 10         | 0.0864            | 0.1634 | 137 | 12         |
| FSAF              | 0.3142               | 0.5419 | 130 | 7          | 0.2605            | 0.4718 | 128 | 8          |
| RetinaNet         | 0.3103               | 0.4987 | 120 | 16         | 0.348             | 0.576  | 120 | 8          |
| Libra RetinaNet   | 0.5033               | 0.7278 | 131 | 8          | 0.5374            | 0.7813 | 130 | 6          |
| PAA               | 0.3256               | 0.8387 | 127 | 6          | 0.33              | 0.834  | 104 | 7          |
| RegNet RetinaNet  | 0.315                | 0.543  | 256 | 8          | 0.313             | 0.539  | 225 | 7          |
| SABL Faster R-CNN | 0.0536               | 0.2244 | 243 | 8          | 0.0501            | 0.2140 | 212 | 12         |
| SABL RetinaNet    | 0.1034               | 0.3817 | 132 | 21         | 0.1857            | 0.6559 | 123 | 6          |

Figure 2.26: Illustration of Training loss of Validation at an AP of 50

In this table, we observed that the number of epochs with the highest validation accuracy varies between 6–21 in models without augmentation and 5–12 in models with augmentation and that the lowest loss values were achieved from different training times in models both with and without augmentation are achieved in SABL Faster R-CNN for train bbox loss and in DCN Faster R-CNN model for train loss.



## Validation of AP50 and AR results of detection models

| Models            | Without Augmentation |        |     |            | With Augmentation |        |     |            |
|-------------------|----------------------|--------|-----|------------|-------------------|--------|-----|------------|
|                   | TB_Loss              | T_Loss | TT  | Best Epoch | TB_Loss           | T_Loss | TT  | Best Epoch |
| DCN Faster R-CNN  | 0.0938               | 0.1821 | 174 | 6          | 0.0969            | 0.1839 | 174 | 5          |
| Dynamic R-CNN     | 0.2978               | 0.5033 | 139 | 8          | 0.2664            | 0.4605 | 156 | 10         |
| Faster R-CNN      | 0.0873               | 0.1573 | 126 | 10         | 0.0864            | 0.1634 | 137 | 12         |
| FSAF              | 0.3142               | 0.5419 | 130 | 7          | 0.2605            | 0.4718 | 128 | 8          |
| RetinaNet         | 0.3103               | 0.4987 | 120 | 16         | 0.348             | 0.576  | 120 | 8          |
| Libra RetinaNet   | 0.5033               | 0.7278 | 131 | 8          | 0.5374            | 0.7813 | 130 | 6          |
| PAA               | 0.3256               | 0.8387 | 127 | 6          | 0.33              | 0.834  | 104 | 7          |
| RegNet RetinaNet  | 0.315                | 0.543  | 256 | 8          | 0.313             | 0.539  | 225 | 7          |
| SABL Faster R-CNN | 0.0536               | 0.2244 | 243 | 8          | 0.0501            | 0.2140 | 212 | 12         |
| SABL RetinaNet    | 0.1034               | 0.3817 | 132 | 21         | 0.1857            | 0.6559 | 123 | 6          |

Figure 2.27: Illustration of Training loss of validation at an AP of 50

This suggests that the best AP50 scores in validation were obtained in Dynamic R-CNN models with/without augmentation among the models used for detection.

## Test AP50 and AR results of detection models

| Models            | Without Augmentation |       | With Augmentation |       |
|-------------------|----------------------|-------|-------------------|-------|
|                   | AP50                 | AR    | AP50              | AR    |
| DCN Faster R-CNN  | 0.547                | 0.391 | 0.577             | 0.323 |
| Dynamic R-CNN     | 0.63                 | 0.341 | 0.654             | 0.323 |
| Faster R-CNN      | 0.617                | 0.343 | 0.624             | 0.395 |
| FSAF              | 0.739                | 0.398 | 0.746             | 0.412 |
| RetinaNet         | 0.621                | 0.377 | 0.652             | 0.338 |
| Libra RetinaNet   | 0.695                | 0.432 | 0.715             | 0.445 |
| PAA               | 0.666                | 0.491 | 0.754             | 0.496 |
| RegNet RetinaNet  | 0.713                | 0.452 | 0.74              | 0.468 |
| SABL Faster R-CNN | 0.622                | 0.427 | 0.595             | 0.402 |
| SABL RetinaNet    | 0.613                | 0.386 | 0.65              | 0.404 |

Figure 2.28: Test AP at 50 of Detection models

In this table among all the models used for fracture detection in wrist X-ray images, the highest AP50 score obtained on the test data was 0.754 in the PAA model with augmentation.

The Precision-recall curve of PAA (best score of 20 models).

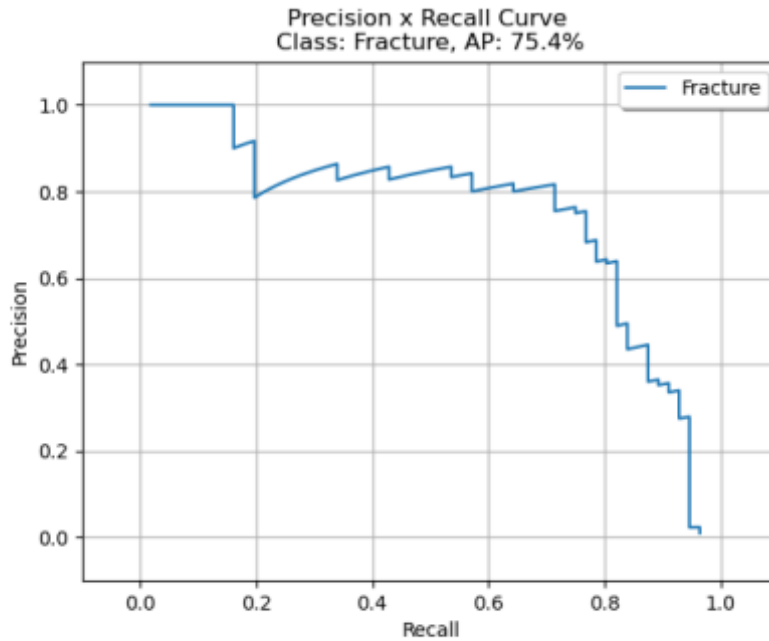


Figure 2.29: Precision and Recall curve

#### Fracture Detection Results of Proposed Models

Based on the results of 20 models based on deep learning in which fracture detection was performed in wrist X-ray images ensemble models were developed, thus leading to an improvement in the detection results.

The precision-recall curve of WFD-C (best score of ensemble models).

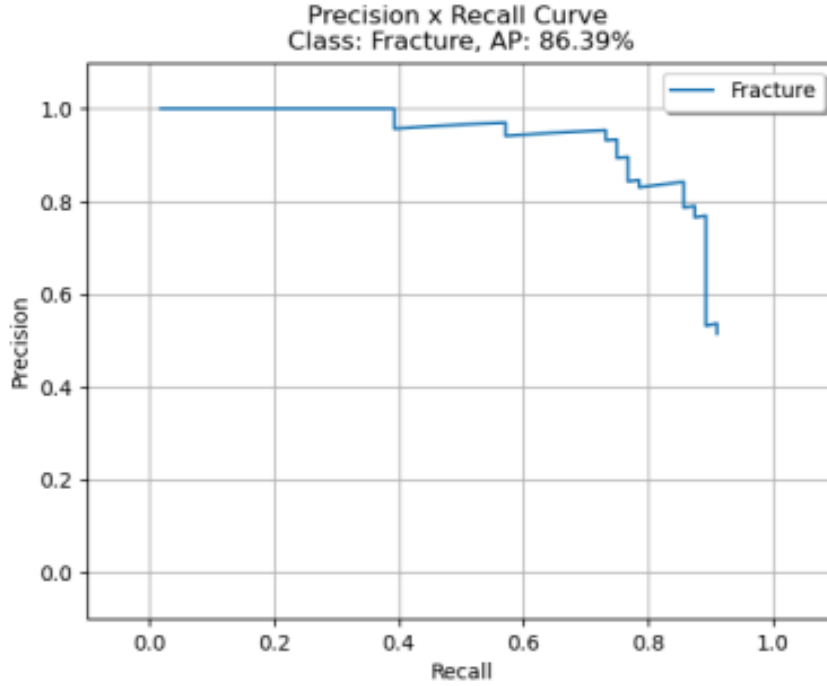


Figure 2.30: Precision and Recall curve

For the results of detection carried out with the ensemble models, the precision-recall curve of the WFD-C ensemble mode with the highest AP score is shown above.

Comparison with various amounts of wrist test data sets.

| Model          | Input         | Dataset | Amount             | AP50   | AR    | LRP <sub>t</sub> | oLRP <sub>t</sub> |
|----------------|---------------|---------|--------------------|--------|-------|------------------|-------------------|
| PAA            | 800 × 800 × 3 | Gazi    | 54 test            | 0.754  | 0.496 | 0.56             | 0.310             |
| YOLOv3         | 750 × 750 × 3 | Gazi    | 54 test            | 0.531  | 0.298 | 0.164            | 0.378             |
| Proposed WFD-C | 800 × 800 × 3 | Gazi    | 54 test            | 0.8639 | 0.33  | 0.357            | 0.349             |
| PAA            | 800 × 800 × 3 | Gazi    | 54 test + 54 valid | 0.629  | 0.499 | 0.552            | 0.304             |
| YOLOv3         | 750 × 750 × 3 | Gazi    | 54 test + 54 valid | 0.516  | 0.286 | 0.164            | 0.364             |
| Proposed WFD-C | 800 × 800 × 3 | Gazi    | 54 test + 54 valid | 0.709  | 0.344 | 0.454            | 0.315             |

Figure 2.31: Comparison of various Wrist Test Data-sets

WFD-C ensemble model developed had the highest AP score

### Drawbacks of the system

- Fracture labeling: other small bone (trapezoid, trapezium, scaphoid, capitate, hamate, triquetrum, pisiform, lunate) fractures in the Wrist were not studied and were ignored.
- YOLO and other deep learning models that do not support  $800 \times 800 \times 3$  sizes were not used.
- In the distribution of the data set, the number of fractures per image was not considered to be equal in the train, validation, and test data set.

[14]

# Chapter 3

## Methodology

### 3.1 Introduction

This chapter describes in detail the entire processing of development, training, and deployment of the deep learning model for the detection of fractures in x-ray images. The process involved the use of different tools and software such as Anaconda, and Google colab among others as will be elaborated.

### 3.2 Data Set Development

#### 3.2.1 Obtaining Data-set

We downloaded the 1548 hand fractured X-ray images from the [universe.roboflow.org](https://universe.roboflow.org), and the images were already labeled in YOLO format. The images we downloaded had no missing annotations and no null examples meaning that all images in the data-set had fractures. The data set had 1846 annotations across the 1548 images with an average of 1.2 annotations per image. Some images from the data-set



Figure 3.1: Images from the Dataset

### 3.3 Exploratory Data Analysis

EDA is used to understand and summarize the contents of a data-set, usually to investigate a specific question or to prepare for more advanced modeling. EDA typically relies heavily on visualizing the data to assess patterns and identify data characteristics that the analyst would not otherwise know how to look for.

### 3.3.1 Random Sampling

| sample | image        | Location         | Fractured | Annotations |
|--------|--------------|------------------|-----------|-------------|
| 1      | 0_jpg        | wrist            | Yes       | Yes         |
| 2      | 3_jpg        | Ulnar and Radius | Yes       | Yes         |
| 3      | 23_jpg       | Ulnar and Radius | Yes       | Yes         |
| 4      | 45_jpg       | Wrist            | Yes       | Yes         |
| 5      | 220_jpg      | Ulna and Radius  | Yes       | Yes         |
| 6      | X_ray_30_jpg | Humerous         | Yes       | Yes         |
| 7      | X_ray_57_jpg | Ulnar and Radius | Yes       | Yes         |
| 8      | X_ray_28_jpg | Ulnar and Radius | Yes       | Yes         |
| 9      | X_ray_83_jpg | Humerous         | Yes       | Yes         |
| 10     | X_ray_90_jpg | Humerous         | Yes       | Yes         |

Figure 3.2: Sampling Table for the EDA

### 3.3.2 Data-set Distribution

To access the data-set distribution we plotted a histogram.

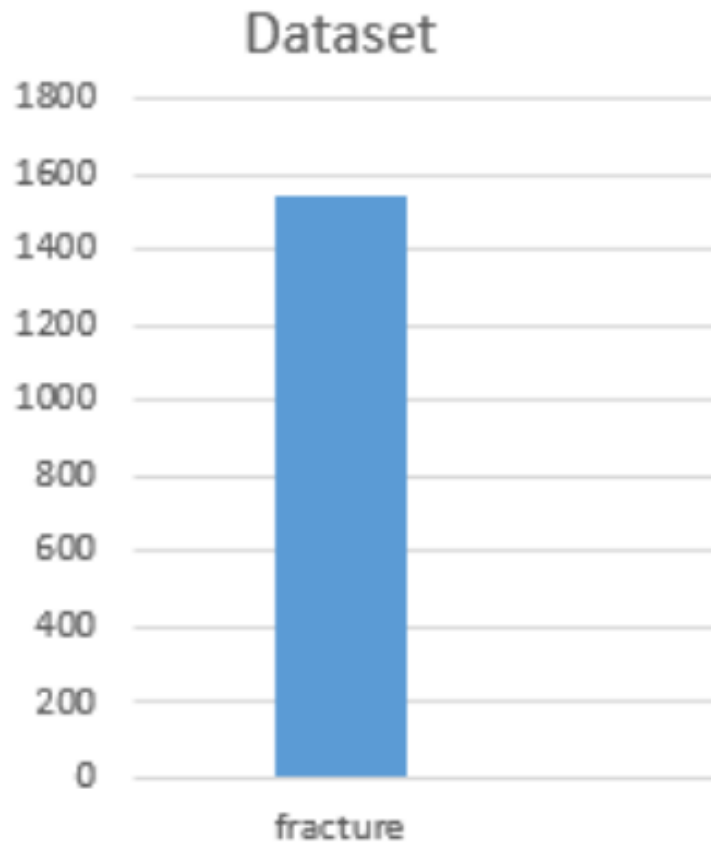


Figure 3.3: Data-set distribution

### 3.3.3 Anatomy of the Data-set

#### Bone Anatomy

Most bones develop from cartilaginous ossification centers which form either a diaphysis (shaft) or an epiphysis (end). During bone growth, the diaphysis and epiphysis are separated by the growth plate (also known as the epiphyseal line or physis) which fuses later in life. The zone adjacent to the growth plate on the diaphyseal side is called the meta-physis.

A sesamoid bone is a bone that ossifies within a tendon. The largest is the patella. Sesamoid bones are also present at the first metatarsophalangeal joint (big toe) and the first metacarpophalangeal joint (thumb). Bone Anatomy Example for the Metacarpal.



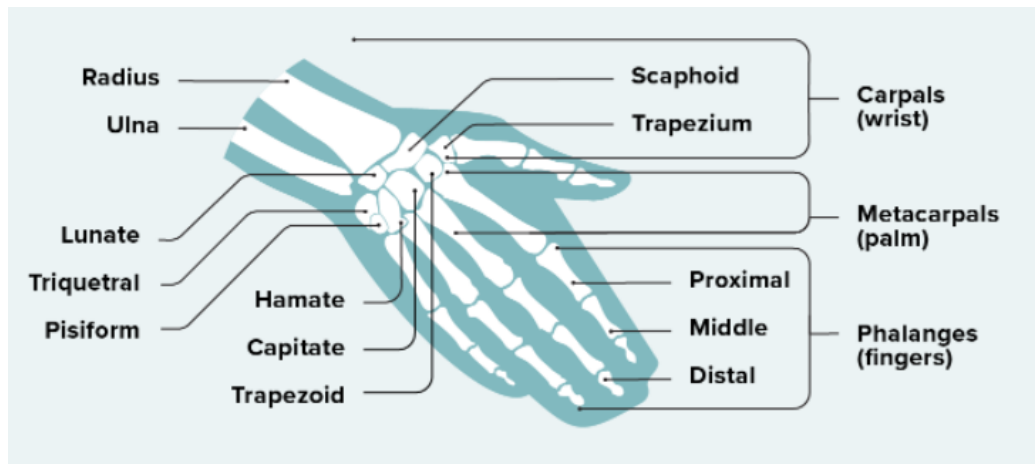


Figure 3.4: Bones of Wrist and Hand

The metacarpals are the five long hand bones between your wrist and fingers. They make up the palm of your hand and are visible through the skin on the back of your hand. Each metacarpal bone corresponds to a digit (finger) and consists of a base, shaft or body, and head.

- First (thumb) metacarpal: thickest, shortest metacarpal bone, moves along with the trapezium.
- Second (index) metacarpal: longest metacarpal bone with the largest base that connects to the trapezium, trapezoid, and capitate.
- Third (middle) metacarpal: articulates with the capitate
- Fourth (ring) metacarpal: articulates with the capitate and hamate.
- Fifth (pinky) metacarpal: smallest metacarpal bone, articulates with the hamate

Bone Anatomy Example Clavicle / Scapula / Humerus



Figure 3.5: Clavicle Scapula and Humerus Image

The clavicles, scapulae, and humeri were often clearly seen in our data-set of X-ray images.

### **Ease of Interpretation**

Is a way to analyze and help people make sense of the data that has been analyzed and collected.

- Identify and mitigate bias. We were able to easily identify and solve the biases of the data-set having more images of the wrist than other parts and also having more male x-ray images of patients.
- Accounting for the context of the problem. Our data-set was able to fall in the context of our project scope of developing an automatic bone fracture system since the majority of the images were fractured.
- Generalization and performance. Our data set was general in terms of having more fractured x-ray images hence improving model performance and inference.

### **3.3.4 Data Preprocessing**

All images were resized to 640x640 using [app.roboflow.com](https://app.roboflow.com) auto-resizing tool, since this size is suitable for the YOLOv5s used during training. Image augmentation was not conducted because the number of images was above 1500 and the YOLO family models generate good precision with data sets of images above 1500.

### 3.3.5 Model Architecture

The model design was implemented using python programming language and TensorFlow libraries and PyTorch and run on local personal computer virtual environments as well as google Col laboratory GPUs for additional processing power. First, the transfer learning technique was exploited on a convolutional neural network architecture yolov5 an established single-stage detector that has been demonstrated to be successful on various image types in object detection tasks. This yolov5 Ultralytics default model was pretrained on a coco data-set which was composed of 80 classes of images ie the bird class. Models are composed of two main parts: the backbone layers which serve as a feature extractor, and the head layers which compute the output predictions. To further compensate for a small dataset size, we'll use the same backbone as the pretrained COCO model, and only train the model's head with fracture detection. YOLOv5 backbone consists of 12 layers, which can be fixed by the 'freeze' argument. Inputs were fed into the model head which resulted in 7038508 trainable parameters without freezing any.

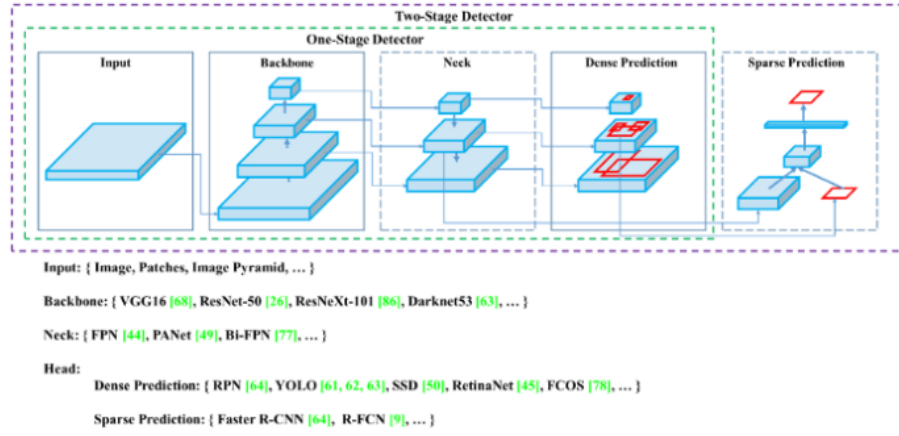


Figure 3.6: YOLOv5 Architecture

All images resized to  $224 \times 224$  using the roboflow API were fed through the yolov5 model. During training, no layers were frozen and there was no fine-tuning which meant we remained with 7038508 trainable parameters. The model is trained on GIOU loss function and optimized with SDG with a learning rate of 0.001 and a decay rate of 0.005.

### 3.3.6 Model Training

We divided the downloaded images into three arts that are to say train set, validation set, and test set in the percentages 70%, 20%, 10% respectively

| Class          | Fracture |
|----------------|----------|
| Train set      | 1077     |
| Validation set | 313      |
| Test set       | 158      |
| Total          | 1548     |

Figure 3.7: Table showing Data-set Split

## 3.4 Training results

The following loss curves in which an indication of the relative training process where the model converged at 120 epochs Since our data has one class only, there are no class misidentifications, and the classification error is constantly zero.



Figure 3.8: Train Objectness Loss

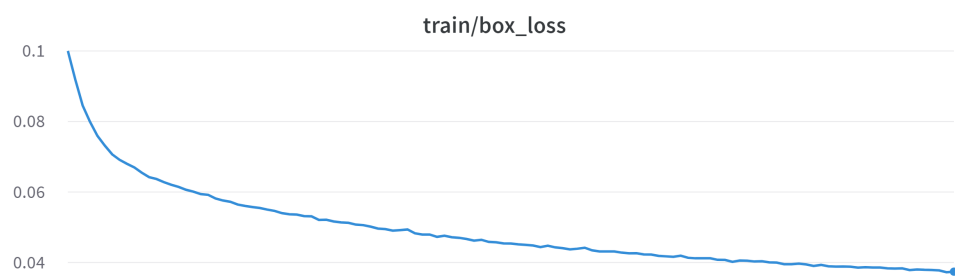


Figure 3.9: Train Bounding Loss

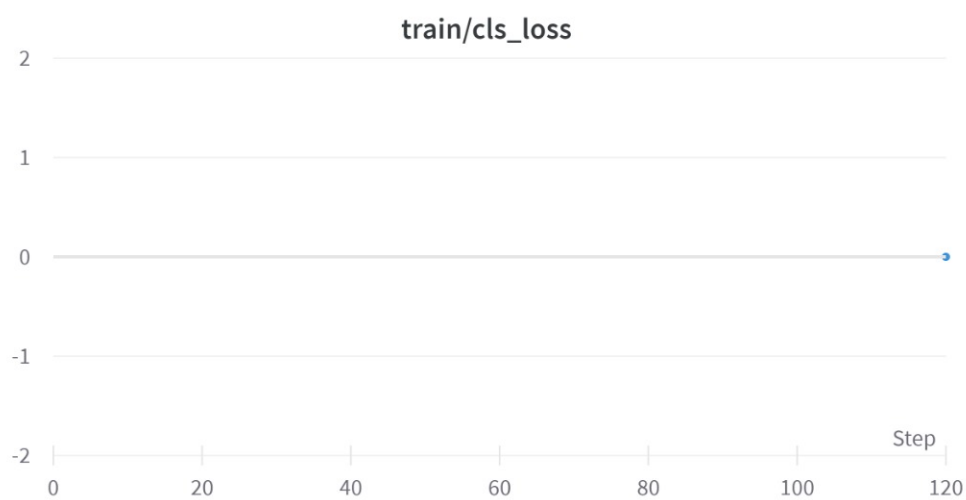


Figure 3.10: Train Classification Loss

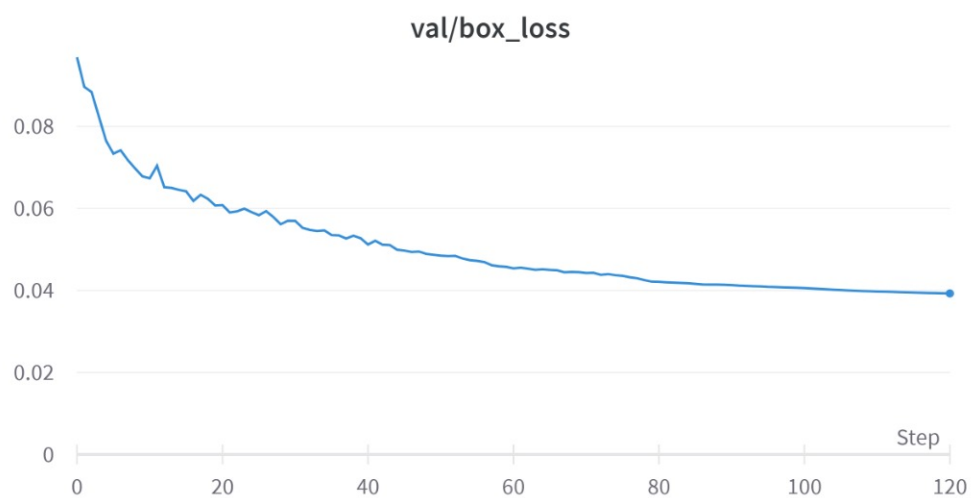


Figure 3.11: Validation Box Loss

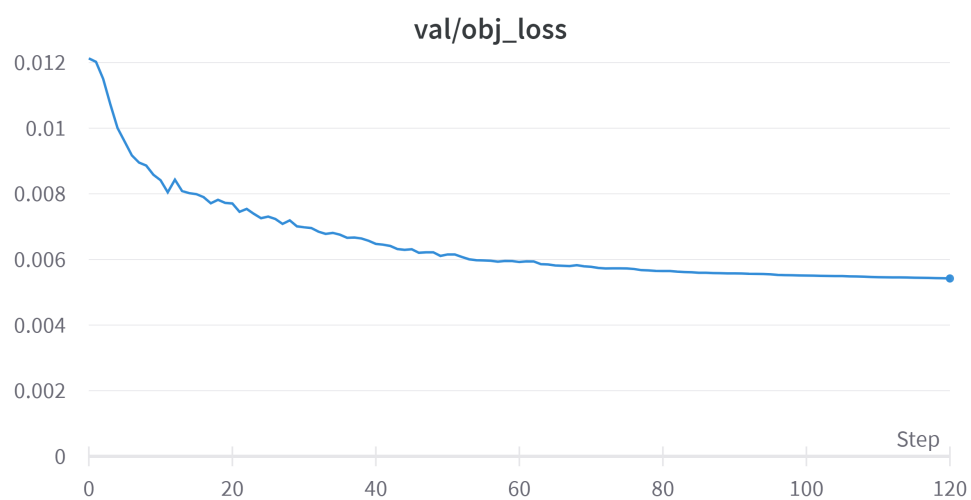


Figure 3.12: Validation Objectness Losss

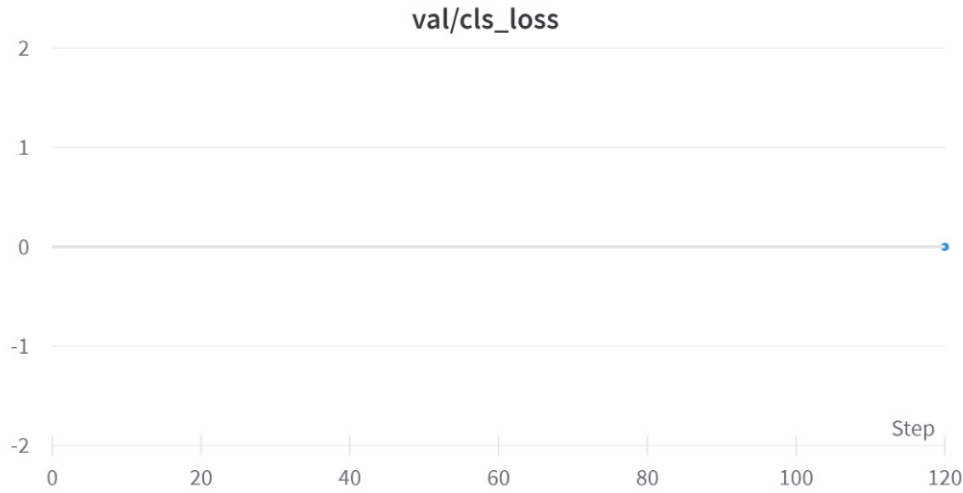


Figure 3.13: Validation Classification Loss

### 3.5 Model Deployment

A web-based application using the Django framework was then developed and the YOLO v5 of the trained model was deployed onto it.

The application code was arranged to fit the three components of the MVC architecture namely;

- **Model** The Model is the part of the web app which acts as a mediator between the website interface and the database. In technical terms, it is the object which implements the logic for the application's data domain. The Model is the component that contains Business Logic in Django architecture.
- **View** This component contains the UI logic in the Django architecture. The view is actually the User Interface of the web application and contains the parts like HTML, CSS, and other front-end technologies.
- **Controller** The controller as the name suggests is the main control component. What that means is, that the controller handles the user interaction and selects a view according to the model. The main task of the controller is to select a view component according to the user interaction and also apply the model component.



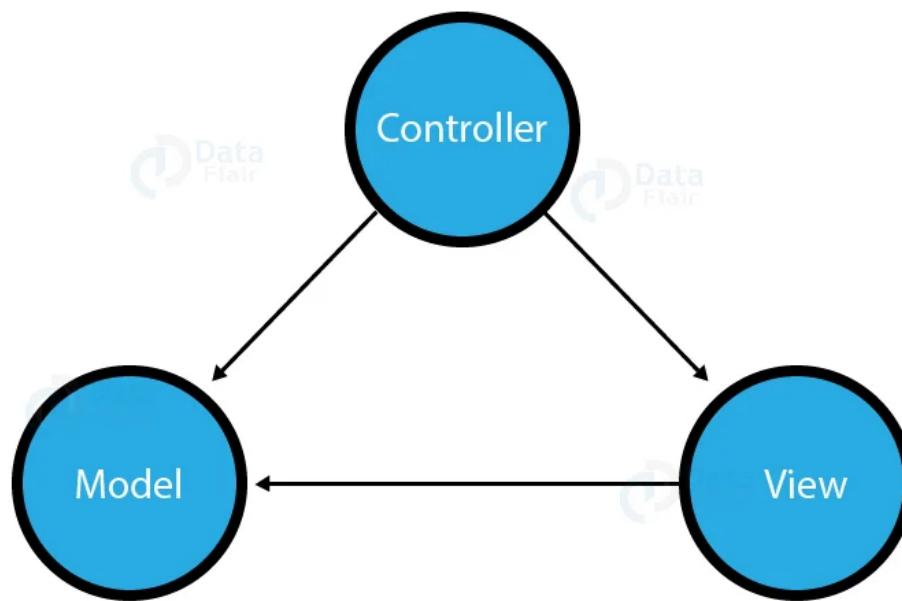


Figure 3.14: MVC Architecture of Django

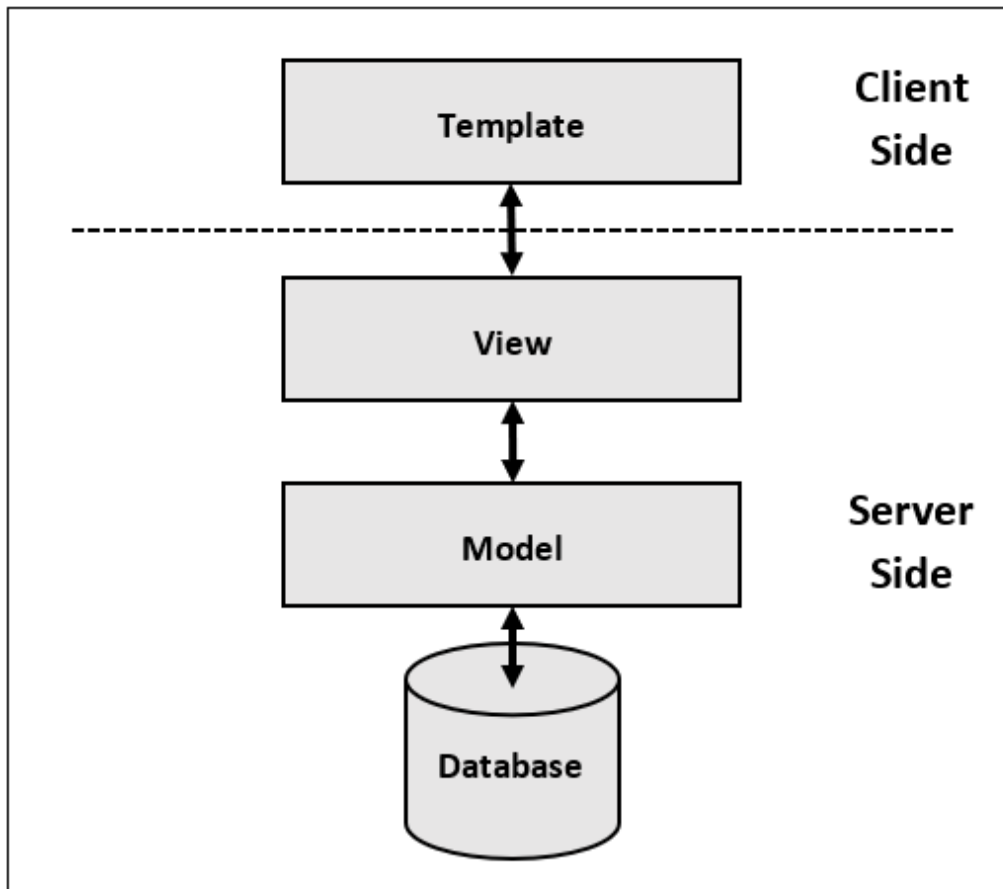


Figure 3.15: Application Structure in Line with the MVC Architecture of Django Framework

# Chapter 4

## Results and discussion

### 4.1 Introduction

In this chapter, we were able to test the model's performance and obtained both qualitative results and quantitative results we also evaluated yolov5 performance with other object detection models.

### 4.2 Qualitative Results

The trained model was tested on the data that was in the test set to obtain inference results. The model was able to detect all the fractures in the test set that appeared in the x-ray images.

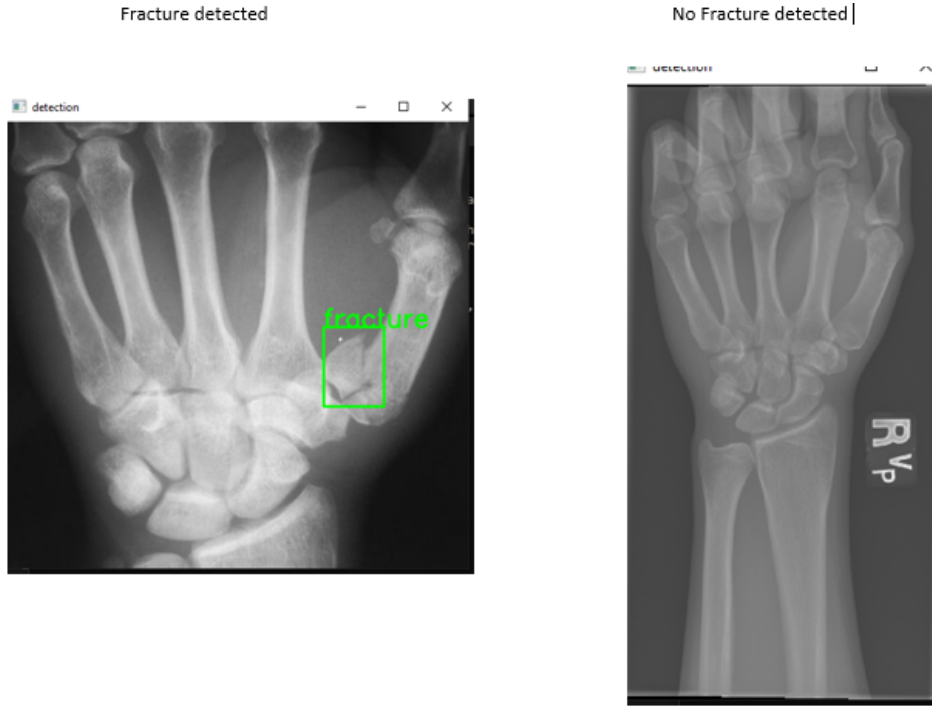


Figure 4.1: Sample of the Model's Inference

### 4.3 Quantitative Results

The key performance metrics used to assess the model's performance were the state-of-the-art dense neural network metrics which are the intersection over Union and with mean average precision being the most of interest since it is the accuracy for object detection models.

The positive signifies 'Fracture detected' while the negative asserts 'No fracture detected'.

Intersection over the union:

is a metric that quantifies the degree of overlap between two regions. IoU metric evaluates the correctness of a prediction. The value ranges from 0 to 1. With the help of the IoU threshold value, we can decide whether a prediction is a True Positive, False Positive, or False Negative.

### Precision

Is a measure of the proportion of the predicted positives that are actually correct. If you are wondering how to calculate precision, it is simply the True Positives out of total detections.  $P = TP / (TP + FP)$  The value ranges from 0 to 1.

### Recall

Is a measure of the proportion of actual positives that were predicted correctly. It is the True Positives out of all Ground Truths. Mathematically, it is defined as follows.

$R = TP / (TP + FN) = TP / \text{Total Predictions}$  The value ranges from 0 to 1.

### Average Precision(AP).

Is the area under the precision-recall curve.

### Mean Average Precision(mAP).

Mean Average Precision or mAP is the average of AP over all detected classes.  $mAP = 1/n \times \sum(AP)$ , where n is the number of classes. where; TP: True Positive FP: False Negative FP: False Positive All the fractures in the x-ray images ("1") test images were correctly detected by the model while the model failed to detect all fractures in x-ray images with multiple fractures. Mean average precision(mAP) of 0.856 at an IOU of 0.5, recall of 84.6 and precision of %90.1

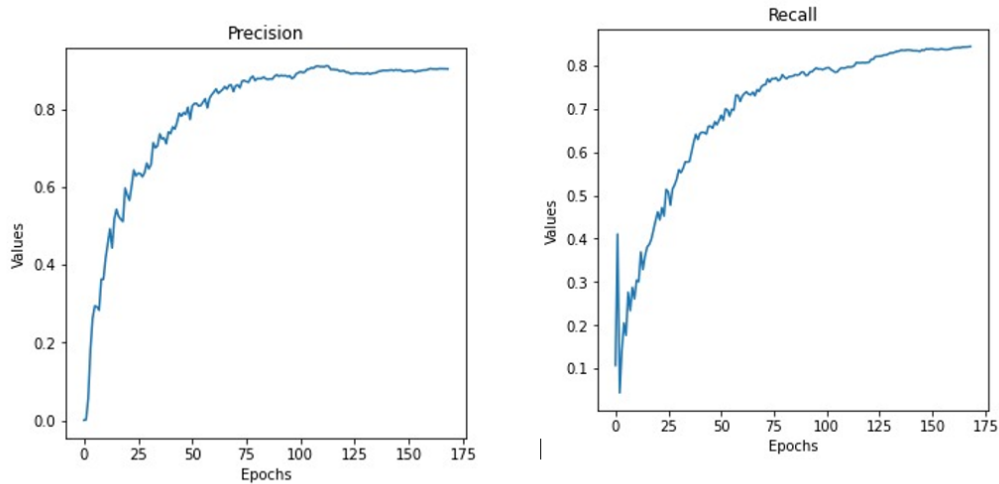


Figure 4.2: Recall and Precision

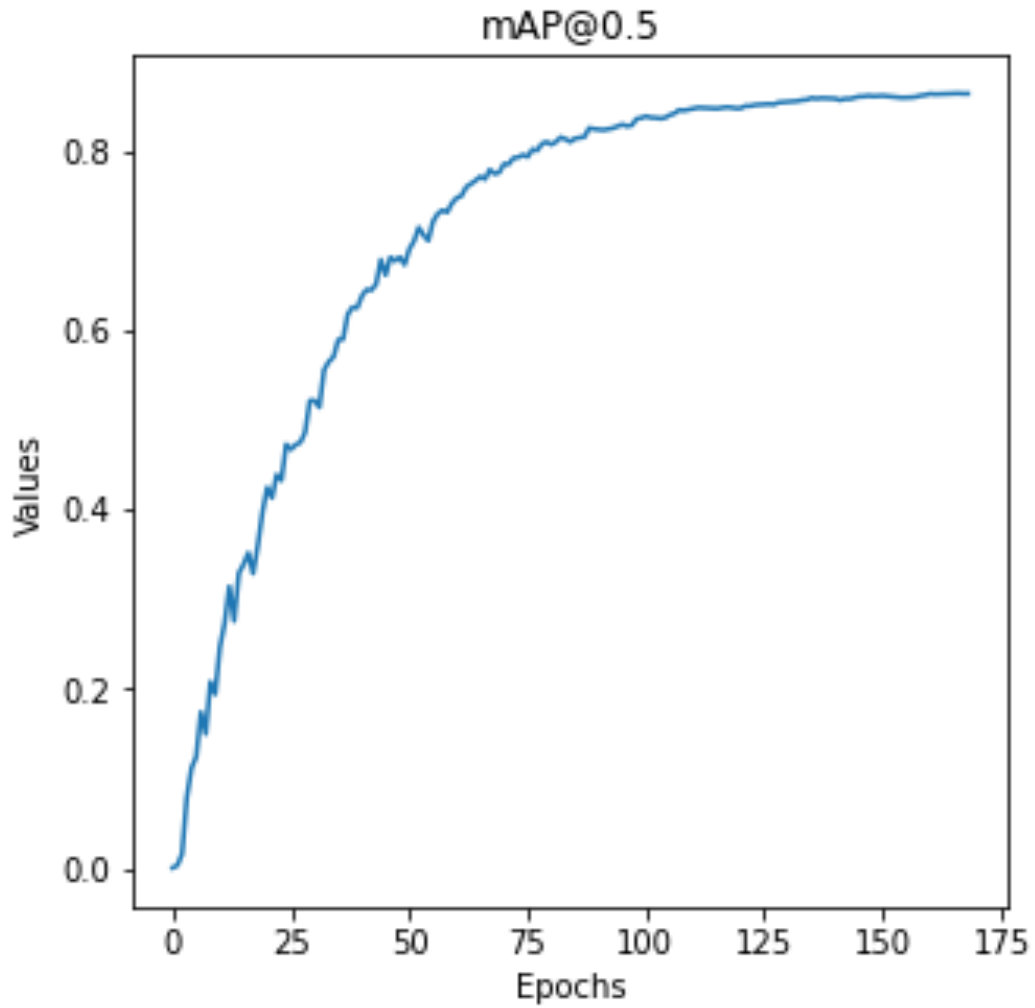


Figure 4.3: Mean Average Precision at 0.5

[13]

#### 4.3.1 Model Evaluation

To demonstrate the effectiveness of the yolov5 model, we compared the results to the performance of an EfficientDet and Detectron 2 model using the same training, validation, and test data set and the following results were obtained at an intersection of the union of 0.5.

| Model         | Metrics                     |        |           |
|---------------|-----------------------------|--------|-----------|
|               | Mean average precision(map) | Recall | Precision |
| Yolo v5       | 86.6                        | 84.6   | 90.1      |
| Detectron 2   | 80.5                        | 81.7   | 80.5      |
| Efficient Det | 78.5                        | 80.5   | 78.5      |

Figure 4.4: Model Results

From the table above, the yolov5 model outperforms the other models.

## 4.4 Model Deployment Results

The different front-end pages such as the features page, log in, image set, and upload pretrained model were designed using HTML, CSS, and JavaScript in VS code after which all the components of the system were integrated, and the model was successfully deployed on the web-based decision support application to allow radiologists to upload of x-ray images for detection of fractures shown in figures below.

Figure 4.5: Web page of login In

Fracture detection app Richard log

### Upload Pre-trained Model

Upload Model Py/Ph File\*  
 No file chosen Allowed extensions are: .pt, .pth

MI Model Classes file\*  
 No file chosen MI Model classes file. Allowed extensions are: .txt, .names, .yaml

Model's description\*

MI Model Version

☐ Public

Figure 4.6: Web page for Uploading a Model

Fracture detection app Richard log

### Upload Pre-trained Model

Upload Model Py/Ph File\*  
 No file chosen Allowed extensions are: .pt, .pth

MI Model Classes file\*  
 No file chosen MI Model classes file. Allowed extensions are: .txt, .names, .yaml

Model's description\*

MI Model Version

☐ Public

Figure 4.7: Web page for uploading an Image



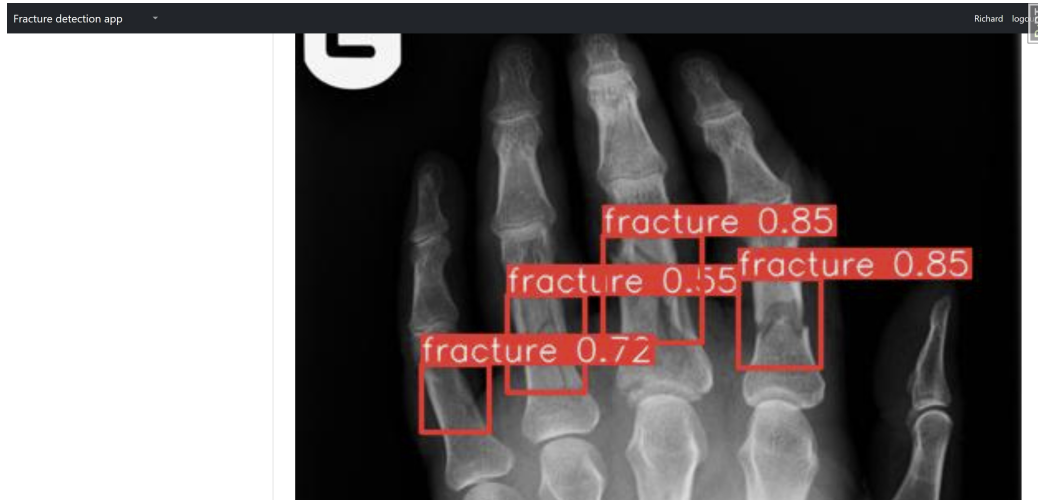


Figure 4.8: Web page for Performing a Prediction

## 4.5 Result Analysis and Discussion

For fracture detection, the model exhibited a high mean average precision in its detection during testing. All the fractured test images were correctly retrieved by the model during testing. Although the model showed very outstanding performance with the detection of single fractures in x-ray images but had a problem detecting multiple fractures in x-ray images due to the non-max suppression in inference.

# Chapter 5

## Conclusion, Challenges and Recommendations

### 5.1 Conclusion

Due to the observer variability among radiologists due to their differences in knowledge and experience, A uniform approach to the detection of bone fractures in X-ray images is needed.

An automatic bone fracture detection system using deep learning was presented in this project.

Open source data was downloaded from roboflow computer vision API and curated, resulting in 1548 fractured images. Three object detection models ie yolov5, efficientDet, detectron 2 were trained, tested, and validated on the same data-set and their hyper-parameters were left in their default setting. Yolov5 outperformed the other models with a mean average precision of 85.6% at an intersection of union of 0.5 and therefore it was deployed on a web-based decision support application.

The application is a decision support tool for radiologists to alleviate the challenge of observer variability during the detection of fractures in x-ray bone images.

It, therefore, becomes more evident that computer vision is highly suitable and goes a long way in automating the detection of fractures in x-ray images to alleviate the challenge of observer variability which can cause image misinterpretation and erroneous prescriptions.

### **5.1.1 Challenges**

The open source data was not large enough for a better model performance with the minimal number of epochs and therefore our results from the model were not the best.

More still, some x-ray images representing some fractures on certain parts of the body were very scarce which slowed down the learning process of the model.

### **5.1.2 Recommendations**

Local bone x-ray bone fracture image data should be collected, pooled, and made more accessible to researchers to ease their work, especially in the area of deep learning solution-based research.

The system should be tested on local clinical data in order for the system to be included in the clinical local setting to solve the problem of backlog.

Lastly, future research can be done in detecting bone fractures for the whole body.

# Bibliography

- [1] N. Fazzalari, “Bone fracture and bone fracture repair,” *Osteoporosis international*, vol. 22, no. 6, pp. 2003–2006, 2011.
- [2] P. V. Saraiya and N. Aygun, “Temporal bone fractures,” *Emergency radiology*, vol. 16, no. 4, pp. 255–265, 2009.
- [3] H. Lule, R. SSebuufu, X. F. Okedi, *et al.*, “Prehospital factors associated with injury severity of motorcycle related femoral fractures at mbarara and kampala international university teaching hospitals in uganda,” 2017.
- [4] S. Dick, “Artificial intelligence,” 2019.
- [5] N. Prandini, E. Lazzeri, B. Rossi, P. Erba, M. G. Parisella, and A. Signore, “Nuclear medicine imaging of bone infections,” *Nuclear medicine communications*, vol. 27, no. 8, pp. 633–644, 2006.
- [6] Y. Ma and Y. Luo, “Bone fracture detection through the two-stage system of crack-sensitive convolutional neural network,” *Informatics in Medicine Unlocked*, vol. 22, p. 100452, 2021.
- [7] Y. L. Thian, Y. Li, P. Jagmohan, D. Sia, V. E. Y. Chan, and R. T. Tan, “Convolutional neural networks for automated fracture detection and localization on wrist radiographs,” *Radiology. Artificial intelligence*, vol. 1, no. 1, 2019.
- [8] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [9] B. Guan, G. Zhang, J. Yao, X. Wang, and M. Wang, “Arm fracture detection in x-rays based on improved deep convolutional neural network,” *Computers & Electrical Engineering*, vol. 81, p. 106530, 2020.

- [10] M. Al-Ayyoub and D. Al-Zghool, “Determining the type of long bone fractures in x-ray images,” *WSEAS Transactions on Information Science and Applications*, vol. 10, no. 8, pp. 261–270, 2013.
- [11] S. L. Greenspan, E. R. Myers, L. A. Maitland, N. M. Resnick, and W. C. Hayes, “Fall severity and bone mineral density as risk factors for hip fracture in ambulatory elderly,” *Jama*, vol. 271, no. 2, pp. 128–133, 1994.
- [12] E. Kironde, P. Sekimpi, I. Kajja, and P. Mubiri, “Prevalence and patterns of traumatic bone loss following open long bone fractures at mulago hospital,” *Ota International*, vol. 2, no. 1, 2019.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [14] G. Moon, S. Kim, W. Kim, Y. Kim, Y. Jeong, and H.-S. Choi, “Computer aided facial bone fracture diagnosis (ca-fbfd) system based on object detection model,” *IEEE Access*, vol. 10, pp. 79061–79070, 2022.