PEOPLE'S DECMOCRATIC REPUBLIC OF ALGERIA
Ministery og higher Education and Scientific Research
University Mohamed Khider – BISKRA
Faculty of Exact Sciences, Natural and Life Sciences
Computer science departement

Order N° : IA06/M2/2023

Dissertation

Presented to obtain the academic master's degree in

Computer Science

Option : **Artificial Intelligence (AI)**

Bone fractures detection in X-ray images

By :

MEFTAH MOHAMED

Defended in 19/06/2023 before the members of the jury composed of :

TELLI ABDELMOUTIA	MAA	President
SAOULI RACHIDA	PR	Supervisor
MOUAKI BENNANI NAWEL	MAA	Examiner

Academic year 2022-2023

Acknowledgments

I would like to express my sincere gratitude and appreciation to all those who have supported me throughout the journey of completing this master's thesis.

First and foremost, I want to thank myself for the tremendous and hard work that I put into realizing this project. I am also deeply indebted to my supervisor, Prof. Saouli Rachida, and Dr. Youkana Imane for their guidance and encouragement. Their expertise and unwavering commitment have played a vital role in shaping this project and significantly enhanced my understanding of the subject matter. I am grateful for their unwavering support and for pushing me to strive for excellence.

My heartfelt thanks go to my family and friends for their unwavering support and encouragement. they have been the driving force behind my perseverance during challenging times. Finally, in alignment with the teachings of the Quran, which emphasize the pursuit of knowledge and encourage seeking wisdom, I am reminded of the verse from Surah Al-Baqarah (2:269): "Allah grants wisdom to whoever He wills. And whoever is granted wisdom is certainly blessed with a great privilege. But none will be mindful of this except people of reason." This divine wisdom has been a guiding light throughout my academic journey, and I am humbled by the opportunity to contribute to the reservoir of knowledge.

MEFTAH Mohamed

Abstract

Bone fractures are a prevalent medical condition that has significant implications for patient care and recovery. The accurate and timely detection of fractures plays a vital role in treatment planning and achieving optimal patient outcomes. However, fracture detection can pose challenges, often requiring the expertise of radiologists and specialized imaging techniques. The application of deep learning methods has shown promising potential in the healthcare field. In this work, we have developed a YOLOv7 model specifically designed for automatic fracture detection in X-ray images which is designed for the use in the emergency department. The dataset used is the GAZ dataset, which comprises 20,327 wrist X-ray images. Through our work, we achieved a precision/recall of approximately 0.91, highlighting the effectiveness of our model in the analysis of X-ray images for fracture detection. These obtained results signify the potential of deep learning techniques in enhancing the efficiency and accuracy of fracture detection, ultimately improving patient care and treatment outcomes. and paving the way for further advancements and applications in medical imaging.

Key-words: Bone fracture, Deep learning, YOLO, X-ray.

Résumé

Les fractures osseuses sont une affection médicale prévalente qui a des implications significatives pour les soins aux patients et leur rétablissement. La détection précise et rapide des fractures joue un rôle vital dans la planification du traitement et l'obtention de résultats optimaux pour les patients. Cependant, la détection des fractures peut poser des défis, nécessitant souvent l'expertise des radiologues et des techniques d'imagerie spécialisées. L'application des méthodes d'apprentissage profond a montré un potentiel prometteur dans le domaine des soins de santé. Dans ce travail, nous avons développé un modèle YOLOv7 spécifiquement conçu pour la détection automatique des fractures dans les images radiographiques, destiné à être utilisé dans le service des urgences. La base de données utilisé dans notre étude est GAZ, qui comprend 20 327 images radiographiques du poignet. Grâce à notre travail, nous avons obtenu une précision/recall d'environ 0,91, mettant en évidence l'efficacité de notre modèle dans l'analyse des images radiographiques pour la détection des fractures. Ces résultats témoignent du potentiel des techniques d'apprentissage profond pour améliorer l'efficacité et la précision de la détection des fractures, améliorant ainsi les soins aux patients et les résultats du traitement, et ouvrant la voie à de nouvelles avancées et applications dans l'imagerie médicale.

Mots-clés: Fractures osseuses, Apprentissage profond, YOLO, X-ray.

Contents

General introduction **1**

1 Overview of Bone Fractures **4**

1.1 Introduction 4

1.2 Bone Anatomy notions 4

1.2.1 Musculoskeletal system 4

1.2.2 The Skeleton 6

1.2.3 Bone 6

1.2.3.1 Bone tissue 6

1.2.3.2 Bone tissue types 7

1.2.3.3 Structure 7

1.2.3.4 Bone Types 8

1.3 Notions about bone fracture 9

1.3.1 Bone fracture categories 9

1.3.2 Bone Fracture types 9

1.3.3 Common problems 10

1.3.3.1 Diagnosis 10

1.3.3.2 Healing of bone 10

1.4 Presentation of X-ray images 11

1.4.1 History 11

1.4.2 X-ray 12

1.4.2.1 X-ray functioning 12

1.4.2.2 Applications of X-ray imaging 13
1.4.3 Diagnostic challenges in fracture detection 14
1.5 Conclusion 15

2 Methods for bone fracture detection 16
2.1 Introduction 16
2.2 Machine learning 16
2.2.1 Types of machine learning 17
2.2.2 Classification 18
2.2.3 Regression 19
2.2.4 K-Nearest Neighbours(KNN) 19
2.2.5 Support Vector Machine (SVM) 20
2.3 Deep Learning 21
2.3.1 Artificial neural networks 21
2.3.1.1 Biological neuron 21
2.3.1.2 Artificial neural 22
2.3.2 Naive Bayes 23
2.4 Activation functions 23
2.5 Transfer Learning 25
2.5.1 Understanding Transfer Learning 25
2.5.2 Strategies for transfer learning 26
2.6 Convolution Neural Network(CNN) 27
2.6.1 Convolutional Neural Network Architecture 27
2.6.1.1 Convolutional Layer 28
2.6.1.2 Pooling Layer 29
2.6.1.3 Fully-connected layer: 30
2.7 Popular CNN architectures 30
2.7.1 VGGNET 30
2.7.2 Densely Connected Convolutional Networks (DenseNet) 31
2.7.3 Residual Network (ResNet) 31

2.7.4 R-CNN 32
2.8 Related work for Bone fractures detection 33
2.9 Conclusion 36

3 Design and implementation of Bone fractures detection system 37
3.1 Introduction 37
3.2 Design of our Bone fractures detection system 37
3.2.1 General architecture 37
3.2.2 Detailed architecture 38
3.2.3 YOLO architecture 39
3.2.3.1 Extended Efficient Layer Aggregation(E-LAN) 40
3.2.3.2 Auxiliary Head Coarse-to-Fine 41
3.2.3.3 Loss functions 42
3.2.4 Dataset preparation 43
3.2.4.1 Collecting data 43
3.2.4.2 Pre-processing 47
3.2.4.3 Data loading 48
3.2.4.4 Detection 48
3.2.4.5 Evaluation of the YOLO model 48
3.3 Implementation of our deep learning model 50
3.3.1 Hardware configuration 50
3.3.2 Frameworks, tools, and libraries 50
3.3.3 Dataset preparation and preprocessing 52
3.3.4 Building our YOLO model 54
3.4 Conclusion 55

4 Experimental Results and Discussion 56
4.1 Introduction 56
4.2 The obtained results for fractures detection 56
4.2.1 Before class removal 56

4.2.2 After class removal 58
4.2.2.1 Precision and recall 58
4.2.2.2 F1 score 60
4.2.2.3 Confusion_matrix 60
4.2.2.4 Loss 62
4.2.3 Model results on Testing data 63
4.3 Results comparison 64
4.4 System interface 65
4.5 Discussion 66
4.6 Conclusion 69

List of Figures

1.1 Musculoskeletal system [10]. 5
1.2 Bone tissue types [13]. 7
1.3 Bone types [16]. 8
1.4 Types of fracture [19]. 10
1.5 First X-ray [22]. 12
1.6 X-ray machine [24]. 13
1.7 Discreet fracture [27]. 15
1.8 Normal bone structure [27]. 15

2.1 Main varieties of machine learning. The main methods are clustering under unsupervised learning and classification and regression under supervised learning. Through interaction with the environment, reinforcement learning improves the model's performance. The training data is represented by colored triangles and dots. The fresh data that the trained model can predict is represented by yellow stars [30]. 17
2.2 supervised learning [31]. 18
2.3 KNN [37]. 20
2.4 Support Vector Machine (SVM) [39]. 20
2.5 Biological Neuron [24]. 22
2.6 The Perceptron [42]. 22
2.7 Transfer learning [48]. 26
2.8 Convolutional Neural Network Architecture [52]. 28
2.9 Convolutional Layer representation [53]. 29

2.10 The basic building block of VGG network: Convolution (Conv) and FC for fully connected layers [56]. 30
2.11 A 5-layer dense block with a growth rate of k = 4 [57]. 31
2.12 Residual learning: a building block [58]. 32
2.13 R-CNN architecture [59]. 32

3.1 The general architecture of our system. 38
3.2 Detailed architecture of our system. 39
3.3 YOLO Architecture [68]. 40
3.4 Extended Efficient Layer Aggregation Network (E-ELAN) [68]. 41
3.5 coarse-to-fine lead head guided assigned [68]. 42
3.6 Wrist bones [73]. 44
3.7 Dataset classes. 45
3.8 The quantity and proportion of annotated objects. Each image may be assigned to many objects [78]. 47
3.9 File in the YOLO format. 48
3.10 Pseudo code for histogram equalization(CLAHE). 52
3.11 Applying histogram equalization(CLAHE) filter and normalizing the picture. 52
3.12 pseudo code for Normalization. 53
3.13 Pseudo code for deleting labels of classes that we want to remove. 53
3.14 pseudo code for reordering the rest of classes labels. 53
3.15 Pseudo code for Dataset splitting. 54
3.16 Spliting result. 54
3.17 command to download YOLOV7 pre trained weights. 54
3.18 Model summary. 55

4.1 Precision/Recall curve. 58
4.2 Precision and recall performance of the model. 59
4.3 Precision/Recall curve. 59
4.4 F1 score curve. 60

4.5 Confusion matrix. 62
4.6 Training loss. 63
4.7 Validation loss. 63
4.8 Result of our model on testing data. 63
4.9 testing example number. 64
4.10 The interface of our bone fractures system. 65
4.11 The interface of our bone fractures system. 66

General introduction

One of the most prevalent medical conditions is orthopaedic fractures, which affect about 11.67 men and 10.65 women globally per 1,000 people each year [1]. Bones are skeletal system organs that provide the body its form, mechanical support, protection, and ease of movement. Additionally, bones play a role in the body's balance of minerals and have recently been linked to the endocrine regulation of energy metabolism [2]. A rupture in the continuity of bone tissue, whether total or partial is known as a bone fracture. In the human body, every bone can fracture. It can occur from minor trauma damage from a medical condition that weakens bones, a high-force impact, stress, or both [3]. The number of orthopaedic fractures is increasing along with the ageing population and advances in diagnosis. The primary tool for fracture diagnosis is radiological detection, nevertheless, radiologists may make mistakes, which can be brought on by inexperience, diminished performance under heavy workloads, such as during night shifts, and the limitations of human vision. Missed fracture diagnoses can cause a variety of issues with patient care and nursing. The use of computer-assisted imaging diagnostics is practical from a clinical standpoint [4].

Bone fracture is a problem that can affect people of all ages because it can result in complications like chronic pain, reduced mobility, and deformities, bone fractures can be dangerous. Fractures can often be fatal if they happen in vital places like the skull or spine because they might harm the surrounding tissues and organs as well. Because it can ensure that the fracture is treated quickly and successfully, early diagnosis of bone fractures is essential. Prolonged healing times, persistent discomfort, and long-term impairment might result from a delayed or inaccurate diagnosis. Additionally, early fracture diagnosis can

reduce the risk of complications like infections, nerve damage, and blood clots. Early fracture detection allows medical professionals to create a customized treatment plan that caters to the particular requirements of each patient, potentially improving results and hastening recovery [5].

In the other hand, the difficult process of identifying bone fractures in X-ray pictures needs the knowledge of competent medical professionals, that's why in healthcare, machine learning techniques are the methods established as a crucial tool for illness detection and treatment and are more often utilized to give clinical decision support and the creation of clinical care recommendations. The support vector machine (SVM) [6] and K-Nearest Neighbours (KNN) [7], are widely used methods in machine learning .

Artificial neural networks are used in deep learning, to model and resolve complicated issues. The capacity of deep learning to reliably and swiftly evaluate massive datasets in healthcare is one of its main advantages. This is especially helpful in the field of medical imaging, where deep learning algorithms can quickly and accurately evaluate hundreds of photos to find patterns and abnormalities that could be hard or impossible for a human expert to notice[8]. Deep learning algorithms can be applied, for instance, to X-ray image analysis to find bone fractures. This can aid medical professionals in developing more precise diagnoses and individualized treatment regimens that are catered to the particular requirements of each patient. [8].

The primary goal in our work is to develop a robust system capable for accurately detecting fractures in X-ray images, with a specific focus on wrist radiographs. In order to achieve this objective, we chose to leverage the power of deep learning and employ the YOLO (You Only Look Once) model. By adopting the YOLO framework, we anticipate achieving a solution that is not only reliable but also highly efficient in detecting fractures within wrist radiographs. Our system aims to contribute the improvement of fracture diagnosis and enhance the overall efficiency of radiology practices. This automatic fracture detection process allows us for reducing the time required for diagnosis, enabling timely treatment decisions, and improving patient outcomes.

Dissertation Outline

The remaining sections of the dissertation are organized as follows:

- The first chapter consists of two main parts. We will start by providing a brief overview of bone anatomy and a notion about bone fracture, after that we will present X-ray imaging. Then, we will provide some problems that face a nan expert in detecting fractures in X-ray images.

- In the second chapter, we provide some background information on our work. We focus on convolution neural networks (CNN), Machine learning and deep learning. Then, we will provide the CNN structure and its various layers and give some popular CNN architectures, On the other hand, we are also going to talk about transfer learning. In the end, a state of art about deep learning and bone fracture is presented.

- In the third chapter, the design of our suggested system as well as its actual implementation are both discussed. Following the outline of the general and detailed architectures, Then, we will move on to discuss the tools that were necessary and the code sources that were utilized in the implementation of our work.

- In the last chapter, we present a summary of our results, Conduct an analysis of the outcomes of the various plans and include a comparison section that outlines the key difference between this work and prior ones and finally provide an example of the system interface.

Chapter 1

Overview of Bone Fractures

1.1 Introduction

The human skeleton is composed of many different structures the most important are the bones, which serve as a support to the body, protect the organs found inside it and allow for movement. As a direct consequence of this, the condition of our skeletal system is of paramount importance to both our general health and our well-being. When a bone is subjected to more stress than it is able to handle this can result in a bone fracture, which is essentially a break or crack in the bone. A bone fracture can be brought on by a number of different things, such as a fall, an accident or an injury sustained while playing sports. There are a variety of clinical methods, Such as diagnostic exams, computer-assisted diagnosis etc., that can be utilized in order to identify these fractures.

In this chapter, we first briefly introduce the anatomy of the bone, bone fracture and presentation of X-ray images. Then, we will talk about problems that face a nan expert in detecting fractures in X-ray images.

1.2 Bone Anatomy notions

1.2.1 Musculoskeletal system

The musculoskeletal system is a component of the human body that gives our body

mobility, stability, form and support. In general it is separated into two systems [9]:

- **Muscular system:** It contains all bodily muscle types. The muscles that move within our joints in particular the skeletal muscles do so. Tendons which connect muscles to bones are also part of the muscular system.
- **Skeletal system:** Bone is its major part. Our bodies are equipped with a robust but dynamic skeleton thanks to the way bones articulate with one another and create the joints. Articular cartilage, ligaments and bursae are skeletal system accessory structures that maintain the integrity and functionality of the bones and joints.

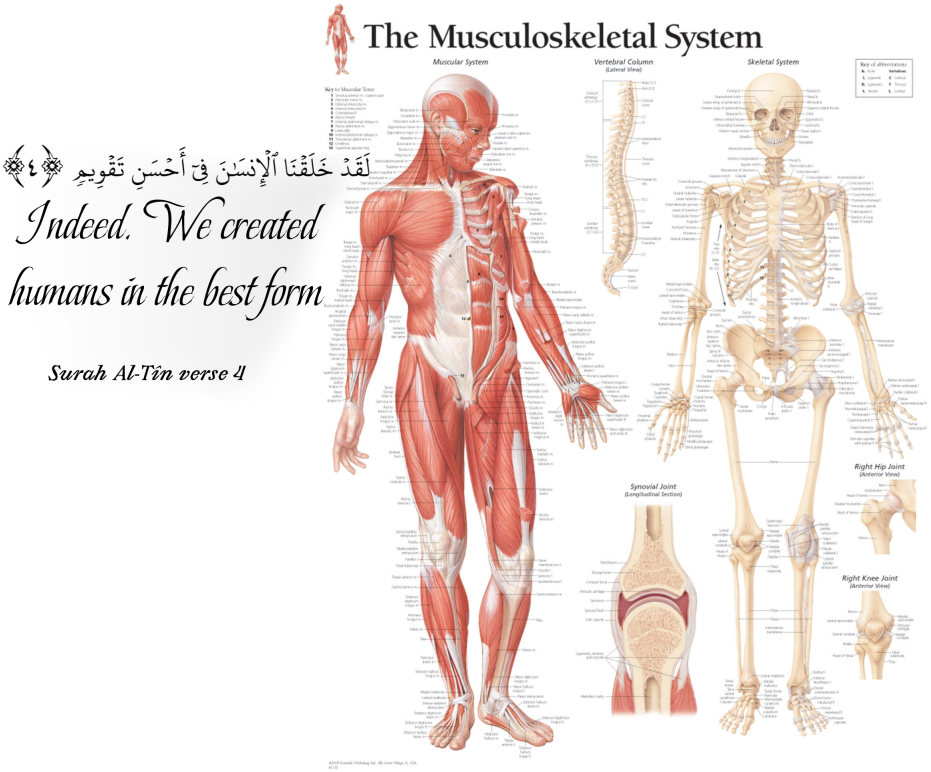


Figure 1.1: Musculoskeletal system [10].

The musculoskeletal system serves many other purposes besides giving the body stability and mobility, the skeletal portion is crucial for maintaining other homeostatic processes like the storage of minerals (such as calcium) and hematopoiesis, whereas the muscular system stores the majority of the body's carbohydrates as glycogen [9].

1.2.2 The Skeleton

The skeleton acts as a system of structural support. It includes features that allow it to expand and contract in order to adapt to different mechanical forces. It aids in the detoxification of heavy metals as well as the calcium/phosphate balance. Throughout life, bone tissue is continually created and altered. This is essential because given the constant twisting and repeated stress it experiences, it would otherwise exceed its tolerance limit. Initially, growth (increase in size) and subsequent modelling give the bone its increased size and structure. The skeleton continuously renews in late childhood and adulthood through a process known as remodelling. For modelling and remodelling to be effective, Bone resorption and bone creation must both take place at the same time. This condition is referred to as "coupling" [11].

1.2.3 Bone

In most vertebrate animals, a bone is a hard organ that is a component of the skeleton. The body's other organs are shielded by bones, which also manufacture red and white blood cells, store minerals, provide the body structure and support and permit motion. Bones contain intricate internal and exterior structures and exist in a range of sizes and forms. They have a number of uses and are both light and sturdy [12].

1.2.3.1 Bone tissue

Is a subset of specialized connective tissue and is also known as osseous tissue or bone in technical meaning. Inside, the bone possesses a honeycomb-like matrix that contributes to the bone's stiffness. Several kinds of bone cells make up bone tissue. The development and mineralization of bone are mediated by osteoblasts and osteocytes, whereas the resorption of bone tissue is mediated by osteoclasts. The lining cells that create a protective coating on the surface of the bone are transformed (flattened) osteoblasts. Ostein, an organic component of the mineralized matrix of bone tissue, is mostly composed of collagen, whereas bone mineral, an inorganic component, is composed of different salts. Cortical bone and cancellous bone are two different forms of mineralized tissue that make up bone.

Other forms of tissue in bones include bone tissue [12].

1.2.3.2 Bone tissue types

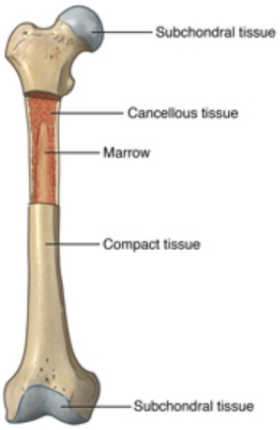


Figure 1.2: Bone tissue types [13].

Bone is living tissue that makes up the body's skeleton. There are 3 types of bone tissue [14]:

- **Compact tissue**: the denser, outer bone tissue.
- **Cancellous tissue**: tissue in bones that resembles a sponge.
- **Subchondral tissue**: the smooth tissue that covers cartilage, another kind of tissue near the ends of bones.

1.2.3.3 Structure

A group of specialized bone cells weaves a complicated web of bonded minerals and a flexible matrix, which together make up around 70% of bone and prevent it from being evenly solid. Bone's distinctive make-up and structure enable them to be reasonably strong and hard while still being light. The elastic collagen fibres also known as ossein, make up 90–95% of the bone matrix, with the remaining material being crushed up. Collagen's flexibility increases fracture resistance. The binding of the inorganic mineral salt calcium

phosphate results in calcium hydroxylapatite, a chemical structure that hardens the matrix. The stiffness of bones is a result of bone mineralization. During the course of life, certain bone cells called osteoblasts and osteoclasts actively build and repair bone. The tissue is intertwined into two basic patterns called cortical and cancellous bone, each with a distinct look and properties within any given bone [15].

1.2.3.4 Bone Types

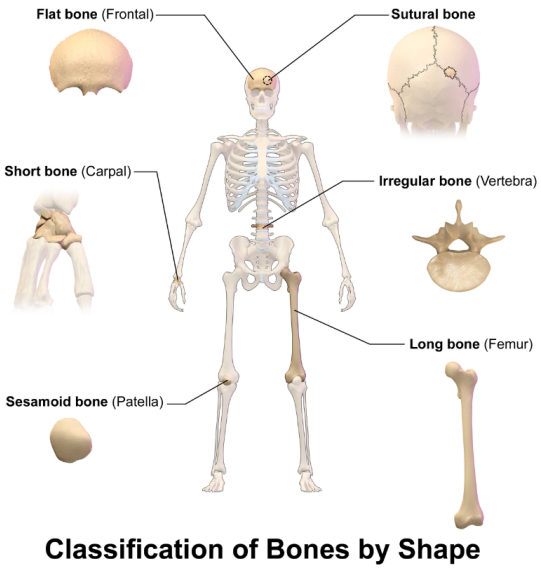


Figure 1.3: Bone types [16].

Five types of bones are present in the human body: Long, short, flat, irregular and sesamoid [17]:

- **Long bones**. The majority of their structure is composed of solid bone with smaller amounts of marrow found in the medullary cavity and regions of spongy, cancellous bone near the extremities of the bones. The majority of limb bones, such as those in the fingers and toes are long bones.

- **Short bones** Are generally cube-shaped and contain a spongy interior surrounded by a thin shell of compact bone. The wrist and ankle bones are small bones.

- **Flat bones** They are frequently bent and slender. The bulk of the bones in the cranium including the sternum have a flat form.

- **Sesamoid bones** Are tendons that encase bones. They function to keep the tendon farther away from the joint, increasing the angle of the tendon and as a result, the muscle's leverage. The patella and the pisiform are two examples of sesamoid bones.

- **Irregular bones** Do not fall within the aforementioned headings. They have a spongy interior surrounded by thin layers of dense bone. Its namesake suggests that their forms are complex and asymmetrical. They frequently have several sites of ossification or have bone sinuses, which account for their uneven form. Uneven bones include those in the spine, pelvis and portions of the skull. The ethmoid and sphenoid bones are two examples.

1.3 Notions about bone fracture

1.3.1 Bone fracture categories

There are several sorts and patterns of fractures and each one must be repaired using a unique approach. Following are some categories of bone fractures [18]:

- **Displaced Fracture**: Occurs when it splits into two or more pieces and shifts.

- **Non-displaced fracture**: Occurs when a bone breaks without moving out of position.

- **Closed Fracture**: The skin is intact.

- **Open Fracture**: The skin has been pierced by the bone.

1.3.2 Bone Fracture types

There are several types of bone fracture [18] :

- **Comminuted Fracture**: Many portions of the bone are broken off.

- **Greenstick Fracture**: A broken bone that is only partially broken and most frequently affects youngsters.
- **Spiral Fracture**: At the place of fracture, one portion of the bone has been twisted.
- **Oblique Fracture**: The design of the break is curving or sloping.
- **Transverse Fracture**: The fractured bone is at a straight angle to the axis of the bone.
- **Linear Fracture**: The fracture runs parallel to the long axis of the bone.

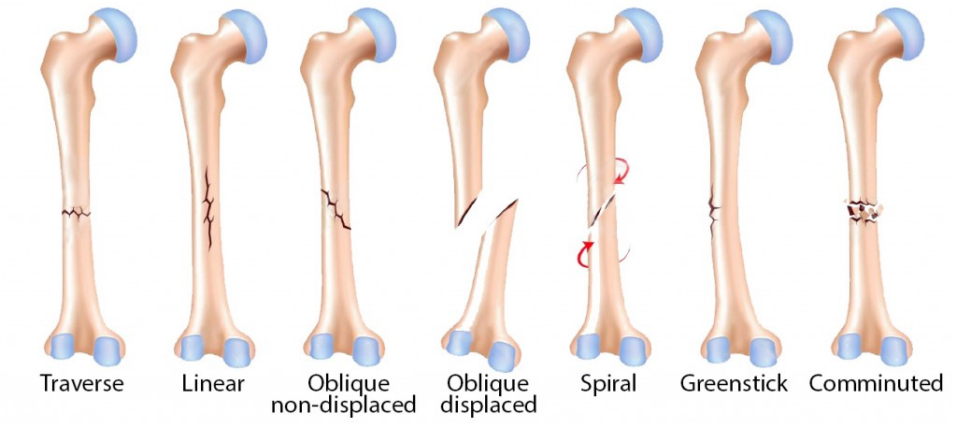


Figure 1.4: Types of fracture [19].

1.3.3 Common problems

1.3.3.1 Diagnosis

In all areas of medicine, diagnostic mistakes are crucial since they are a sign of poor patient care. Studies showed that emergency X-rays can be misread in different ways, with up to 26% of the time being wrong [20].

1.3.3.2 Healing of bone

Several diverse outcomes can result from traumatic disturbance of bone, such as fractures or surgical osteotomies. The mode of disruption, the kind of fixation used, the pattern of fracture or osteotomy and the mechanical and biological environment to which

the bone is exposed all affect the outcome. For instance, fractures that occur in semirigid circumstances develop an exterior and interior callus and are then brought back together using a cartilage model. Several adverse circumstances such as intervening soft tissues, avascularity and Excessive motion. The presence of foreign material, tumour or infection can cause this process to become disturbed and lead to "nonunion." There might not form any transitional cartilage or an exterior callus under tightly specified circumstances. A separate series of processes occurs if tensile pressures are present as in callotasis or distraction bone development. The cellular processes that function in these various circumstances are just now starting to be understood [11].

1.4 Presentation of X-ray images

1.4.1 History

100 years ago, Rontgen discovered X-rays. Physicists and engineers were primarily the professions at the forefront of technological advancements in X-ray apparatus in the late 1890s. As a result of their fascination with pictures, photographers and "medical men" as they were then known in the scientific literature also contributed. These authors first focused primarily on the uses of the new rays in surgery referring to the location of foreign substances [21].

Eventually, medical professionals made several contributions to physics including suggestions for measuring scales, beam quality gauges and equipment attachments like x-ray tube shields and diaphragms. Nonetheless, it is especially fitting that Physics in Medicine and Biology commemorates the centenary with a study that highlights the contributions of physics and engineering in the early years after the discovery of x-rays.

Wilhelm Conrad Rontgen made the discovery of X-rays on November 8th 1895 at the University of Wurzburg's Physics Institute. When an unexpected observation was made he was experimenting with different physics contributions to x-ray diagnostics between 1895 and 1915 Lenard and Crookes tubes. On a piece of thin cardboard that was wrapped in black light-tight paper and placed next to one of the stimulated tubes, some platino-barium

cyanide fluorescent material was visible glowing. Rontgen quickly realized that these X-rays could pierce anything other than black paper, such as a hardwood plank, a thick hook and metal sheets. But more importantly, he discovered that "Strangest of all while the flesh was quite transparent bones were opaque" according to his biographer Glasser (1931, 1933) [21].

Figure 1.5: First X-ray [22].

1.4.2 X-ray

Like visible light, X-rays are electromagnetic radiation. In contrast to light X-rays have higher energy and can penetrate most materials including the human body. To create pictures of the tissues and structures inside the body, medical X-rays are employed. A picture of the "shadows" cast by the things inside the body will be created if X-rays passing through the body also pass through an X-ray detector on the patient's opposite side.

Photographic film is one sort of X-ray detector but there are many more that are used to create digital pictures. Radiographs are the X-ray pictures produced by this method [23].

1.4.2.1 X-ray functioning

A patient is positioned such that the body portion being scanned is situated between an X-ray source and an X-ray detector to generate a radiograph. When the equipment is turned on, X-rays enter the body and depending on the radiological density of the tissues

they pass through tehy are absorbed in various amounts by various tissues. The density and atomic number (the number of protons in an atom's nucleus) of the substance being photographed are both factors that affect radiological density. For instance, calcium found in human bones has a greater atomic number than the majority of other tissues. This characteristic makes bones easily absorb X-rays, which results in strong contrast on the X-ray detector. As a result, against a radiograph's dark backdrop, the bone structure stands out as being whiter than other tissues. In contrast, fat, muscle and air-filled cavities like the lungs allow X-rays to pass through more easily because they are less radiologically dense than other types of tissue. On a radiograph, these structures are seen in various hues of grey [23].

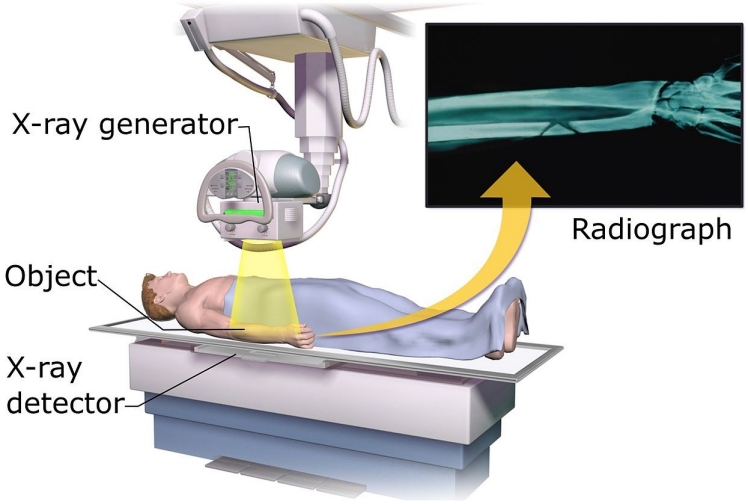


Figure 1.6: X-ray machine [24].

1.4.2.2 Applications of X-ray imaging

X-ray imaging is widely used in many medical specialities including radiology and orthopaedics. X-ray imaging is the most commonly used for the reasons listed below [25]:

- **Diagnosis**: a variety of problems including fractures, dislocations, infections, cancers, lung diseases, gastrointestinal disorders and cardiovascular ailments can be diagnosed with X-ray imaging. X-ray scans include important data that may be used to make a precise diagnosis and outline a course of action.

- **Monitoring**: in order to follow the healing of fractures, check the efficacy of chemotherapy or radiation therapy for cancer or gauge the success of joint replacements, X-ray imaging is employed for monitoring the course of treatment.

- **Interventional procedures**: interventional operations such as biopsies, abscess drainage, Catheter installation and joint injections are guided and monitored using X-ray imaging. To achieve precise needle placement and reduce problems, real-time supervision during these procedures frequently uses fluoroscopy.

1.4.3 Diagnostic challenges in fracture detection

Detecting fractures in X-ray images is a complex task that can present challenges we state blow some of them :

- **Time constraints:** speed is essential in the emergency room and there might be time restrictions for looking over and interpreting X-ray images. To help with timely patient diagnosis and treatment, quick turnaround times are needed. Due to time constraints, it is more likely that one will overlook minor fractures or interpret something incorrectly [26].

- **Large volume of cases:** since emergency rooms frequently see a large number of patients, X-rays must be interpreted quickly while maintaining accuracy. The workload may be heavy which could make one tired and make them overlook fractures or other abnormalities [26].

- **Skeletal immaturity:** fracture detection presents unique difficulties in youngsters because their bones are still growing. Children's growth plates may resemble fracture lines and their fractures can be discreet. Pediatric radiology knowledge is necessary to distinguish between fractures and normal variants in pediatric patients [26].

- **Discreet fractures:** if a fracture is tiny, situated in a complicated location or covered by other bodily structures it might not always be apparent on an X-ray [27].

Figure 1.7: Discreet fracture [27].

- **Complex anatomy:** The necessity to distinguish between true fractures and typical alterations in bone structure. Someone without specialized experience could misinterpret a bone's normal fluctuations in form or density for a fracture [27].

Figure 1.8: Normal bone structure [27].

1.5 Conclusion

In this chapter, we provided a brief overview about the essential concepts and knowledge related to bone anatomy, Bone fractures, X-ray images and the challenges faced by nan experts in detecting fractures in X-ray images. In the following chapter, we will try to go into more detail covering everything from artificial neural networks to CNN networks. This includes the key modules and functions that are used to build architectures.

Chapter 2

Methods for bone fracture detection

2.1 Introduction

Bone fractures are a common injury that needs a precise and prompt diagnosis for the right course of care. The conventional method of detecting bone fractures depends on human skill in interpreting diagnostic pictures, including X-rays or CT scans. However, with the development of artificial intelligence in medical imaging, there is rising interest in applying AI techniques for bone fracture diagnosis to enhance diagnostic precision, effectiveness and patient outcomes [28].

This chapter gives an overview about the deep learning methods that we will use to develop our work. our main focus is on the Convolutional Neural Network (CNN) architecture, in which we begin by discussing machine learning and the various types of algorithms that it employs. After that, we move on to discussing deep learning algorithms. In addition, we will provide a presentation of the most common CNN architectures. In conclusion, an overview of the current state of the art about bone fracture detection using deep learning is presented.

2.2 Machine learning

Using algorithms and statistical models, machine learning is a subset of artificial intelligence (AI) that enables computer systems to learn from data and make predictions

or judgments based on that data without being explicitly programmed. It is a process through which a computer system, without being specifically configured for that purpose naturally improves its performance on a task via experience [29]. Indeed, the fast rise in the amount of patient data available has now put ML in the headlines for creating data-oriented approaches to precision medicine [30].

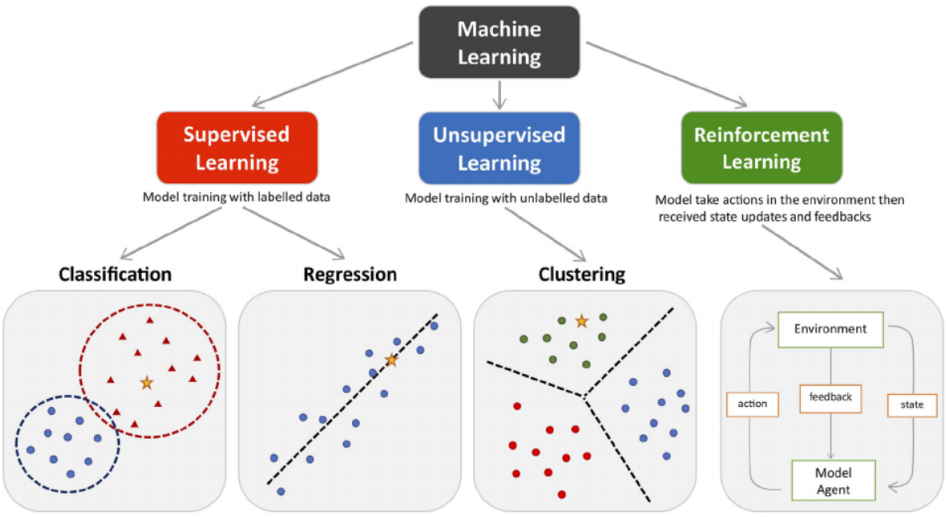


Figure 2.1: Main varieties of machine learning. The main methods are clustering under unsupervised learning and classification and regression under supervised learning. Through interaction with the environment, reinforcement learning improves the model's performance. The training data is represented by colored triangles and dots. The fresh data that the trained model can predict is represented by yellow stars [30].

2.2.1 Types of machine learning

In machine learning, we can distinguish three types of learning as following:

- **Supervised learning**: this is a learning method for identifying relationships between independent qualities and a chosen dependent attribute (the label). Using input and output values, supervised learning consumes a training dataset to create a prediction model. The model may then forecast the values of the output for a fresh dataset. In order to gain better generalization and stronger prediction ability for new datasets, the performance of models constructed via supervised learning depends on the size

and variance of the training dataset. The majority of induction algorithms fall under the heading of supervised learning [31].

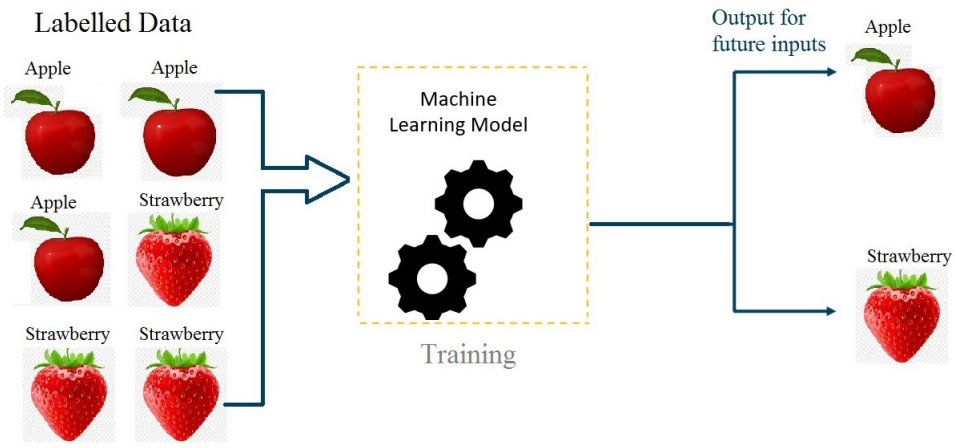


Figure 2.2: supervised learning [31].

- **Unsupervised learning**: this is a learning strategy that combines instances without requiring a dependent attribute to be declared. This method often entails discarding pure unstructured noise and discovering organized patterns in the data. Algorithms for dimensionality reduction and clustering are often unsupervised. K-means and the Apriori Algorithm for Learning Association Rules are two commonly used algorithms in unsupervised learning [32].

- **Reinforcement Learning**: involves an intelligent agent's (RL- agent's) investigation of an adaptive series of actions or behaviours in a predetermined environment with the goal of maximizing the cumulative reward. The environment changes in a way that can be seen as a result of the intelligent agent's behaviour. The learning approach trains itself for a given set of experimental actions and observed reactions to the condition of the environment to create an adaption model [29].

2.2.2 Classification

The process of finding a model to assist divide data into different categorical classes, Data simplification, and prediction are within the classification category. Such data sets

may be summarized using the classification process, which also aids in identifying the data set's significant links and structure. If separate classes of items are discovered to exist, they can be named, their attributes may be enumerated, and information can be organized and retrieved more effectively. This also makes it easier to assign new objects to the system [33].

2.2.3 Regression

A common statistical tool is regression modelling, which offers an easy way to build a functional connection between variables. For quantitative response variables, there are univariate and multivariate methods. For predictor variables, there are simple and multiple methods, for data that can be transformed linearly or nonlinearly, there are linear and nonlinear methods, for qualitative variable predictors there is an analysis of variance and analysis of covariance methods and for qualitative response variables, there are logistic methods [34]. The Least Squares Method for regression was invented by Legendre and Gauss. By adding the squares of each equation's residual, this approach in Linear Regression creates approximations that best match the data. One of the most used types of regression analysis is linear regression. Despite the fact that each independent variable is not necessarily linear, in this method the parameters are defined as a linear combination. Similar to basic linear regression, but with additional independent variables is multiple linear regression. When the parameters are not linear, nonlinear regression must be used. This iteratively minimizes the function by applying the sum of squares method [35].

2.2.4 K-Nearest Neighbours(KNN)

In data-mining applications including classification, regression and missing value imputation, the KNN approach has been effectively developed. The main goal of a conventional KNN approach is to predict a test data point's label using the majority rule or more specifically, by using the major class of the test data point's k most comparable training data points in the feature space [36]. Using a conventional KNN technique, we can perform classification, regression, and missing value imputation.

i.e. It is fair to use k = 3 and k = 2 for the left test data point and the right one, respectively, for a comparable regression (or missing value imputation) case in Figure 2.3. This scenario also suggests that in actual KNN prediction applications, various test data points should use varying numbers of nearest neighbours. It claims that in actual classification applications, using k as a fixed constant for all test data points (the whole issue space) can frequently result in low prediction rates[37].

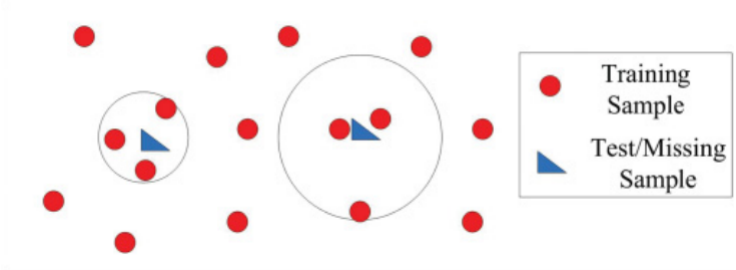


Figure 2.3: KNN [37].

2.2.5 Support Vector Machine (SVM)

Support vector machine (SVM) algorithm, a supervised machine learning technique, has shown good performance in resolving classification issues in several biological domains, particularly in bioinformatics [38]. SVMs may effectively do non-linear classification in addition to linear classification by implicitly translating their inputs into high-dimensional feature spaces. This technique is known as the kernel trick. In essence, it draws boundaries between the classes. The margins are designed to have the shortest possible distance between them and the classes, which minimizes classification error[39].

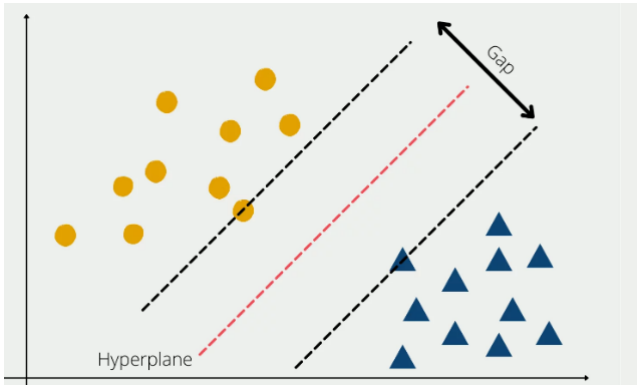


Figure 2.4: Support Vector Machine (SVM) [39].

2.3 Deep Learning

2.3.1 Artificial neural networks

Artificial neural network (ANN), which mimics the biological neural network in the human brain that regulates thought and behaviour, is a form of supervised learning strategy in machine learning utilizing labelled inputs. Learning in-depth information about the makeup of biological neuron networks is made possible by simulating the anatomy of the brain, including neurons and neuronal activity. Each neuron, or nerve cell, in the human brain is connected to several other neurons via synapses, generating very intricate neural networks. Hence, a synaptic function is crucial. Modern neuroscience has a solid foundation because synapses allow neurons to communicate with one another. Since neural networks feature several parallel computing and dispersed information processing processes, humans are capable of self-learning. Scientists want to use current computer-based programs to replicate the human brain by utilizing people's capacity for self-learning. Hence, the parallel computing model of the human nervous system is used to describe the structure of an ANN. An artificial neural network (ANN) is a type of technology that simulates how the brain and nervous system process information. Hence, the parallel computing model of the human nervous system is used to describe the structure of an ANN. An artificial neural network (ANN) is a type of technology that simulates how the brain and nervous system process information.[40]

2.3.1.1 Biological neuron

The brain's building blocks are neurons. This means that studying the brain requires a thorough understanding of neurons. Each neuron in the normal human brain has at least 10,000 connections to other neurons, making up the brain's estimated 100 billion neurons. In neurons, electrical signals travel through axon branches that finish on the dendrites or cell bodies of potentially thousands of other neurons before emanating from the dendrites or cell body in response to stimulus from other neurons. Synapses are specialized structures that connect the terminals of axons to the dendrites or cell bodies of other neurons [41].

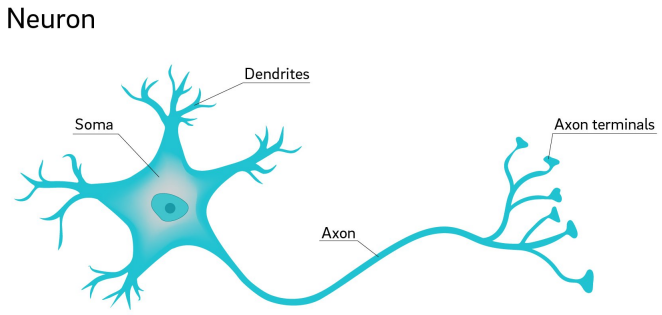


Figure 2.5: Biological Neuron [24].

2.3.1.2 Artificial neural

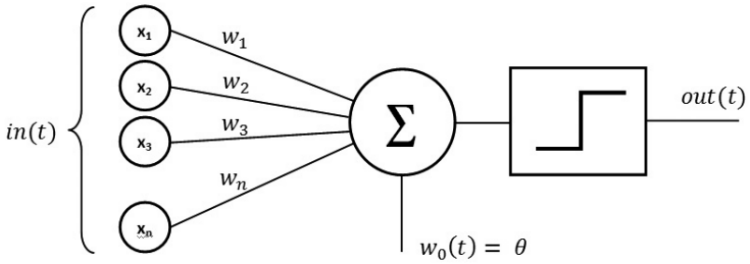


Figure 2.6: The Perceptron [42].

One artificial neuron makes up a perceptron, a neural network with an input layer and a series of connections connecting the input units to the output unit as illustrated in the figure 2.6 above. A perceptron's objective is to categorize patterns that are given to input units. The output unit's fundamental function is to add the values of each input (xn) multiplied by the weight, or connection strength, of each input (wn), to the output unit. In the picture above, the threshold is determined by comparing the weighted sum of the inputs $\sum_{i=1}^{n} w_i * x_i$, which is then processed through a step function. If the total is larger than the threshold, "1" is the output; if not, "0" is the output. As an illustration, the input may be the pixel intensities in an image or, more generally, characteristics that are extracted from the original picture, such as the contours of objects. One image at a moment is shown, and the perceptron determines whether or not it belongs to a category,

such as the category of cats. Only one of two possible states "off" if the picture is not in the category or "on" if it is, and can be selected for the output. The binary values for "on" and "off" are 1 and 0, respectively [42].

2.3.2 Naive Bayes

The well-known Naive Bayes classifier uses the Bayes theorem to divide data into groups based on basic training criteria. By assuming that characteristics are independent of class, the naive Bayes classifier dramatically simplifies learning. Despite the fact that independence is often a bad assumption, in reality, naive Bayes frequently outperforms more advanced classifiers. A given example that has been characterized by its feature vector is given the most likely class using a Bayesian classifier. Naive Bayes has shown to be useful in many real-world applications, including text classification, medical diagnosis, and system performance management. Learning such classifiers may be substantially streamlined by assuming that characteristics are independent of a given class [41]. For some issues classes with a high degree of feature dependencies, including disjunctive and conjunctive ideas, demonstrate naive Bayes optimality [43].

2.4 Activation functions

It determines whether or not a neuron should be activated. It will determine whether or not the neuron's input is significant throughout the prediction process. This decision is made using more basic mathematical procedures. The two fundamental categories of activation functions are: Linear and Non-linear Activation Functions. In the following, we will present the most common activation function types because there are many of them and each one has a particular application [44].

- **ReLU (Rectified Linear Unit) :** nowadays, the ReLU is the activation function that is employed the most globally. Considering that practically all convolutional neural networks and deep learning employ it to convert all input values to positive

numbers [44] as determined by the following equation:

$$Relu(z) = max(0, z) \tag{2.1}$$

- **Leaky ReLU (Rectified Linear Unit) :** when the input is negative, the Leaky ReLU activation function, a variation of the normal ReLU activation function, permits a small, non-zero gradient. where the slope of the function for negative inputs is controlled by the tiny positive constant (for example, 0.1). and it is defined by the following equation :

$$f(x) = x, if x > 0, \alpha x, if x <= 0, \tag{2.2}$$

Due to its capacity to address the "dying ReLU" issue, the Leaky ReLU activation function is preferred in YOLO and many other deep learning architectures. This issue arises when the ReLU function sets all negative values to zero, rendering the corresponding neurons inactive and preventing them from continuing to learn. The Leaky ReLU resolves this problem and guarantees that all neurons in the network continue to update their weights, enhancing the learning capability of the model. It does this by introducing a small gradient for negative inputs [45].

- **ELU (Exponential Linear Unit) :** it is a function that travels toward cost convergence to zero more quickly and produces more precise results [44] as determined by the following equation:

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha \cdot (\exp(x) - 1), & \text{if } x < 0 \end{cases}$$

- **Sigmoid :** we chose the sigmoid function primarily because it can be found between 0 and 1. As a result, it is particularly utilized for models whose output is a probability prediction. Since anything's probability only exists between 0 and 1, the sigmoid is the best option [44] as determined by the following equation:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

- **Softmax :** It is mostly utilized for classification problems. The softmax function returns the probability distribution for each class of the model when given a vector of inputs. The distribution's total values add up to one. Typically, the output of a neural network's last layer serves as the input to the softmax function. Consequently, it is often added at the network's end. With the help of the softmax function, the input vector is transformed into a probability distribution whose exponential is proportional to the input values. This indicates that the input vector's members may be positive, negative, or zero. However, the output numbers would always range from 0 to 1. It is therefore simple to interpret as determined by the following equation [46]:

$$(zi) = eziPKj = 1ezjfori = 1, 2, ..., K \tag{2.4}$$

2.5 Transfer Learning

Transfer learning is the process of using a machine learning model that has already been trained to solve a separate but connected problem. For instance, if you trained a straightforward classifier to identify the presence of a backpack in a picture, you might utilize the information the model learned during training to identify additional items like sunglasses. The primary idea behind transfer learning is to use what has been learned in one activity to enhance generalization in another. We move the weights that a network has picked up at "task A" to a fresh "task B". The main concept is to apply what a model has learned from a task with a lot of labeled training data to a new task with little to no training data. We begin the learning process using patterns discovered while completing a comparable activity, as opposed to starting from scratch [47].

2.5.1 Understanding Transfer Learning

Neural networks, for instance, frequently attempt to identify edges in the earlier layers, forms in the middle layer, and certain task-specific properties in the latter layers of computer vision. The early and intermediate layers are utilized in transfer learning, whereas

the latter layers are just retrained. It aids in utilizing the labelled data from the first job it was trained on. Transfer learning is the process of transferring as much information as feasible from the task the model was trained on to the current task. Depending on the issue and the information, this knowledge might take many different forms. For instance, it can be the way models are put together, which makes it simpler for humans to recognize fresh items [48].

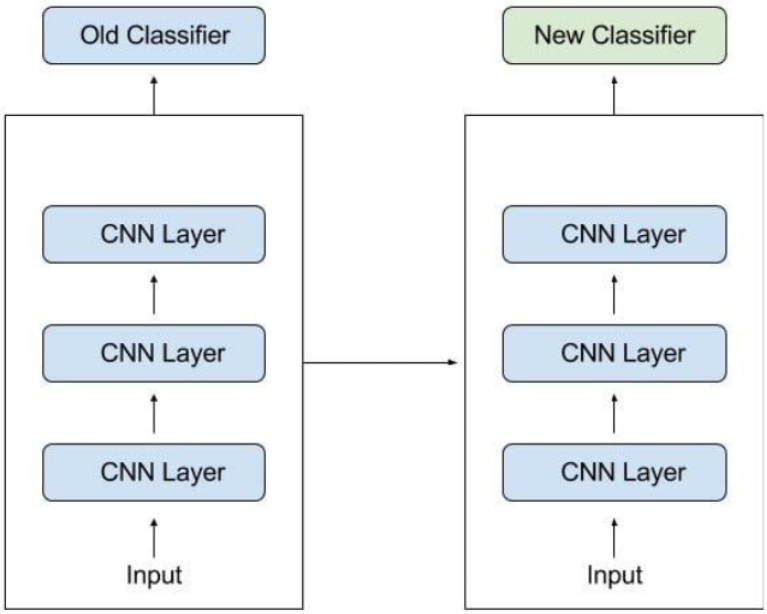


Figure 2.7: Transfer learning [48].

2.5.2 Strategies for transfer learning

The following are some strategies for effective transfer learning [49]:

- **Directly use pre-trained model:** for a comparable job, the pre-trained model may be employed right away. For instance, you may predict the categories of photographs using Google's InceptionV3 model. The excellent accuracy of these models has previously been shown .

- **Fixed features:** the features created for the data points using the information from one model can then be given to new models as fixed features. The output of any layer from a pre-trained ConvNet may be utilized as a feature vector for this picture, for instance, if you run the fresh photos through it. For the required condition, a

classifier may be created using the characteristics that were so established. The word vectors can also be used directly in the text classification model.

- **Fine-tuning the model:** this method allows for network fine-tuning while using the pre-trained network as your model. You may feed your photos to the InceptionV3 model, for instance, and utilize the pre-trained weights as an initialization (instead of random initialization), for the image classification model. A significantly lesser amount of user-provided data will be used to train the model. The benefit of this approach is that weights can attain the global minimum with little training and data. You may also fix certain layers (often the first few) and then solely fine-tune the rest.

- **Combining models:** you can replace the top few layers of a pre-trained model with a new classifier, train this merged network, and leave the pre-trained component unchanged rather than having to retrain the model's top few layers.

2.6 Convolution Neural Network(CNN)

The discipline of image processing and computer vision has been completely transformed by convolutional neural networks (CNNs), a type of deep learning method. CNNs are made to automatically identify important details in images and understand intricate patterns by training on massive amounts of data. CNNs have attained state-of-the-art performance in a variety of image-related tasks such as image classification, object detection, and image generation. thanks to its hierarchical architecture and capacity to capture local and global dependencies in images [50].

2.6.1 Convolutional Neural Network Architecture

Convolutional neural networks have three main types of layers, which are:

- Convolutional layer
- Pooling layer
- Fully-connected layer

A convolutional network's initial layer is the convolutional layer. The fully-connected layer is the last layer, even though convolutional layers, further convolutional layers, or pooling layers, might come after it. The CNN becomes more complicated with each layer, detecting larger areas of the picture. Early layers emphasize basic elements like colours and borders. The bigger features or forms of the item are first recognized when the visual data moves through the CNN layers, and eventually, the desired object is recognized [51].

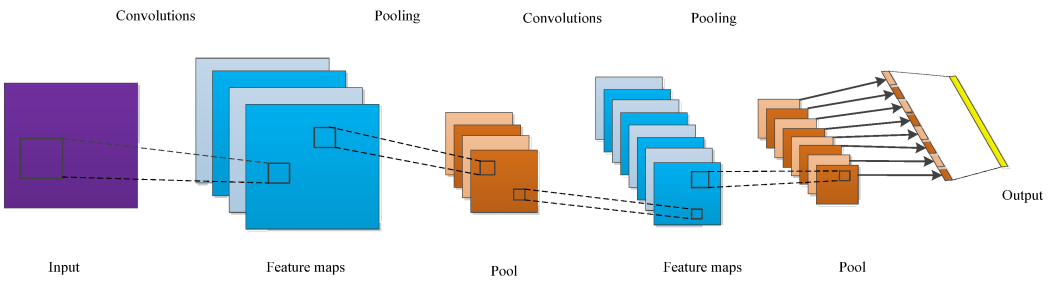


Figure 2.8: Convolutional Neural Network Architecture [52].

2.6.1.1 Convolutional Layer

The foundational layer of a CNN is the convolutional layer. In order to extract local characteristics like edges, corners, and textures, which is also where the majority of computation takes place. It needs input data, a filter, and a feature map, among other things. Assume that the input is a color image that is composed of a 3D pixel matrix.As a result, the input will have three dimensions height, width, and depth that are related to RGB in an image. Additionally, we have a feature detector, also referred to as a kernel or filter, which will move through the image's receptive fields and determine whether the feature is there. his process is known as a convolution. A section of the image is represented by a two-dimensional (2-D) array of weights acting as the feature detector. Normally a 3x3 matrix, the filter size also determines the size of the receptive field but can vary in size. The dot product between the input pixels and the filter is calculated after the filter has been applied to a section of the picture. This dot product is then supplied to the output array. The filter moves by a stride and repeats the process when the kernel has gone over the entire picture. A feature map, activation map, or convolved feature is what the succession of dot products from the input and filter ultimately produce [53].

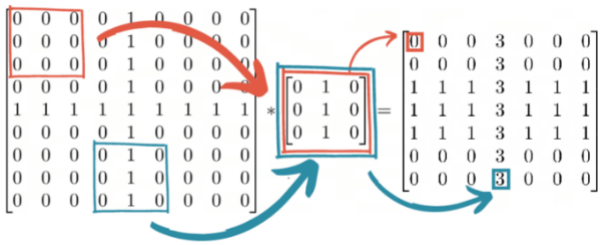


Figure 2.9: Convolutional Layer representation [53].

A second convolution layer may come after the first one, as was previously described. Because the subsequent layers will be able to see the pixels in the previous layers' receptive fields when this happens, the CNN's structure may then become hierarchical. Use the example of attempting to identify whether a bicycle is there in a picture. You may think of the bicycle as being made up of many parts. It is made of a frame, pedals, wheels, and handlebars. The bicycle's individual parts, each of which represents a lower-level pattern in the neural network, and the bicycle as a whole, a higher-level pattern, produce a feature hierarchy inside the CNN [53].

2.6.1.2 Pooling Layer

This layer down-samples the retrieved features along the spatial dimensions in an effort to reduce the number of features [54] . With the exception of the fact that this filter doesn't have weights, the pooling operation applies a filter to the entire input comparable to how the convolutional layer does. Instead, the output array is filled with values from the receptive field by the kernel using an aggregation function [55]. The following are the two main pooling types [54]:

- Max pooling: as the filter moves over the input, it selects the pixel with the greatest value to transmit to the output array.
- Average pooling: as it traverses the input, the filter computes the average value inside the receptive field and delivers that value to the output array.

2.6.1.3 Fully-connected layer:

Typically, convolution layers and pooling layers are put before dense layers. This layer's objective is to calculate either the class scores or the hidden activations.[54]

2.7 Popular CNN architectures

2.7.1 VGGNET

The 2014 ILSVRC's runner-up was The Visual Geometry Group (VGG). This work's key contribution is that it demonstrates how important network depth is for improving recognition or classification accuracy in CNNs. Two convolutional layers make up the VGG architecture, and both of them employ the ReLU activation function. A single max pooling layer and numerous fully linked layers that also use a ReLU activation function come after the activation function. A Softmax layer for categorization makes up the model's top layer. Convolution filter size is altered in VGG-E to a 3 3 filter with a 2 stride. Three VGG-E models were proposed, with the models having 11, 16, and 19 layers, respectively. In Figure 2.10, the VGG network model is displayed [56].

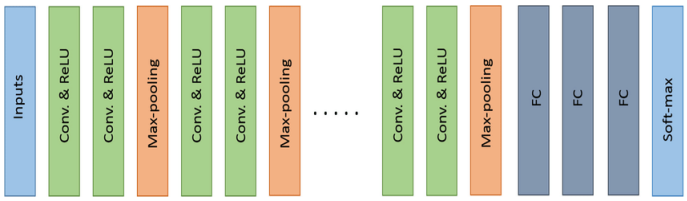


Figure 2.10: The basic building block of VGG network: Convolution (Conv) and FC for fully connected layers [56].

The VGG-E models all have three completely linked layers as their final result. However, there were various numbers of convolution layers. Convolution layers made up 8 of the layers in VGG-11, 13 of the layers in VGG-16, and 16 of the layers in VGG-19. VGG-19 [56].

2.7.2 Densely Connected Convolutional Networks (DenseNet)

The outputs of each layer are coupled with all succeeding layers in a dense block in the DenseNet system, which Gao et al. designed in 2017. As a result, it has a dense network of connections between its layers, earning it the moniker DenseNet. By effectively reusing features, network parameters are drastically decreased. The transition blocks between two adjacent dense blocks make up the majority of the DenseNet [57].

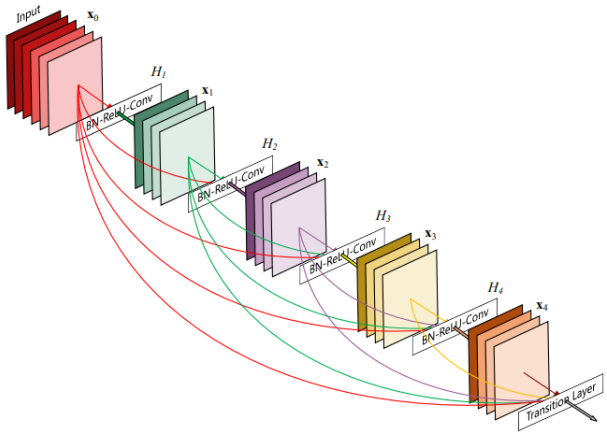


Figure 2.11: A 5-layer dense block with a growth rate of k = 4 [57].

All of the previous feature maps are fed into each layer. deconstructed each layer was given as input all of the feature maps from earlier layers, the figure 2.11 above illustrated it.

2.7.3 Residual Network (ResNet)

Is a type of neural network that has been frequently used for computer vision problems. Since its initial release in 2015[58], it has gained popularity as a solution for picture recognition problems. Deep neural networks struggle with vanishing gradients, but ResNet enables the model to skip layers without sacrificing performance. For a number of years, the ImageNet Large Scale Visual Recognition Challenge has been won by the ResNet architecture. ResNet has a wide range of variations, and scientists are constantly looking for methods to make it function better. ResNet has a number of important features, such as batch normalization, skip connections and residual blocks. ResNet has also been used in

other fields, including speech recognition and natural language processing. Overall, ResNet has made a substantial contribution to the field of computer vision and is a potent tool for deep learning practitioners [58], the figure 2.12 below shows Residual learning.

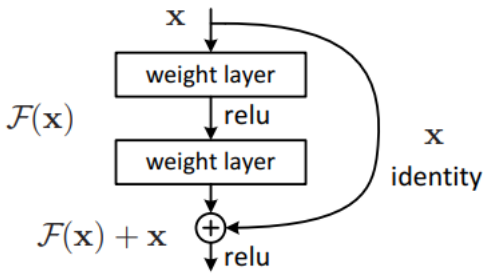


Figure 2.12: Residual learning: a building block [58].

Figure 2.18 displays the ResNet architecture's fundamental block diagram. A conventional feedforward network with a residual link is called ResNet. Based on the outputs of (l - 1)'th, which originate from the preceding layer denoted by xl - 1, it is possible to define the output of a residual layer. The final result of numerous procedures is F(xl - 1).

2.7.4 R-CNN

R-CNN stands for Region Convolutional Neural Networks is A two-stage detection algorithm, unlike YOLO which is a one-stage detection algorithm. A subset of areas in an image that potentially contain an item is found in the first stage. The item is classified in each region in the second stage. R-CNN is a slow object detection algorithm because it must process each image twice. The first pass extracts features from region proposals, and the second pass trains a classifier and regressor to predict the object class and bounding box [59]. The figure 2.13 below illustrates the R-CNN architecture.

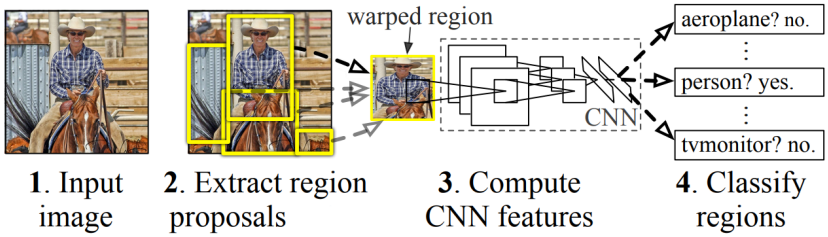


Figure 2.13: R-CNN architecture [59].

2.8 Related work for Bone fractures detection

In recent times, there has been a lot of focus placed on the application of deep learning in the process of detecting bone fractures. Several different deep learning strategies have been suggested as possible aids for medical professionals in diagnosing and arriving at accurate treatment choices. in this section, we show some of the works that are closely linked to our study.

- **Pranav Rajpurkar, et al.**[60]: They employed a huge dataset of musculoskeletal radiographs consisting of 40,895 pictures of the upper extremity derived from 14,982 studies where each study has been individually categorized by radiologists as either normal or abnormal. Pranav Rajpurkar, et al trained a 169-layer highly connected convolutional network on this dataset in order to identify and locate abnormalities. The performance obtained was an accuracy of 85 %.

- **Seok Won CHUNG, et al.** [61]: The total dataset for this study consisted of 1,891 plain shoulders radiographs from 1,891 patients (591 men, 1,300 women; 1,083 from Konkuk University Medical Center, 209 from Kyungpook National University Hospital, 165 from Myungji Hospital, 203 from Kangwon National University Hospital, 41 from the National Police Hospital, 25 from Seoul Saint Mary's Hospital, and 165 from Wonkwang University Sanbon Hospital). This study was conducted in South Korea. Seok Won CHUNG, et al used ResNet 152 in order to train the dataset, and as a result, they obtained an accuracy of around 96 %.

- **Takaaki Urakawa, et al** [62]: The patient's legs were turned internally when the radiographs of their hips in the anterior view were obtained. The dimensions of the field of vision were 429 mm by 352 mm. Using a Digital Imaging and Communications in Medicine viewer (View R; Yokogawa, Tokyo, Japan), the images of 1773 patients were examined by a single board-certified orthopedic surgeon (T.U.). Plain radiographs of the hips taken from these individuals were cropped so that they displayed only proximal fractured and non-fractured femurs. This resulted in a total of 3346 hip pictures, 1773 of which were fractured and 1573 of which were non-fractured. for

the purpose of classifying hip pictures. According to their results, the best performance was accomplished by utilizing one of the CNNs known as the Visual Geometry Group 16-layer (VGG_16) network and the end result had an accuracy of 83% for diagnosing fractures.

- **Rui-Yang Ju and Weiming Cai [63]:** On 21 April 2023, Wang et al presented a new method for detecting wrist fractures in X-ray images. The authors propose a YOLOv8-based model that is trained on the GRAZPEDWRI-DX dataset which has 20,327 X-ray pictures of wrist injuries. At the Department of Pediatric Surgery at the University Hospital Graz, many pediatric radiologists collected these photos from 6,091 patients between 2008 and 2018. By adding bounding boxes to the photos, 9 distinct types of annotation are added. The model was able to achieve a mean average precision (mAP) of 0.947.

- **Kaifeng GAN, et al** [64]: Two senior orthopedists with more than ten years of professional experience in the field of orthopedics conducted a retrospective review of 2,359 plain wrist radiographs along with diagnostic reports from 2,359 adult patients who underwent radiological examinations at the Medical Center of Ningbo City, Lihuili Hospital, of the Ningbo University School of Medicine between January 2010 and September 2017 in order to verify that each case had a correct diagnosis. This review was carried out in order to determine whether or not each patient had, After consulting with a senior orthopedist who had 22 years of professional experience in the orthopaedic field, a consensus was reached. 2,340 radiographs of the wrist were taken, including 1,491 abnormal instances and 849 normal wrists. In recent image classification competitions, Inception-v4, the diagnostic model that Kaifeng GAN et al employed, have achieved state-of-the-art results, making it an ideal choice for Kaifeng GAN et al. They achieved an accuracy of 90 %.

- **Balaji, et al** [65]: This research makes use of a database consisting of 175 femur X-rays that were taken with a Philips X-ray equipment at the Raja Muthaiah Medical College Hospital, which is affiliated with Annamalai University. The database was

obtained from the Department of Orthopedics. Each image has a size of 600 by 800 pixels, and out of the total number of photographs, 100 are considered normal while the remaining 75 have fractures. in this study, Balaji et al used the baseline CNN model, and the end result had an accuracy of 90.7% for diagnosing fractures.

- **Gang Sha, et al** [66]: The data on spinal lesions that Gang Sha et al collects come from Xijing Hospital, which is the Second Affiliated Hospital of Shaanxi University of Traditional Chinese Medicine. These are true clinical data that have been gathered in the hospitals that were mentioned previously. We marked the location and size of spinal lesions from CT images by labelling, then distinguished lesions to (cervical fracture, cfracture), (thoracic fracture, tfracture), and (lumbar fracture, lfracture), 40 cases spinal fracture lesions data consisting of 5134 images. Gang Sha et al. used the YOLO-tiny network. The data was labeled with the assistance of the attending doctors in the Department of bone tumour and Orthopedics. the results of the study show the mean average precision (mAP) of the proposed method is 86.63%.

- **Waseem Abbas, et al** [67]: The dataset that was utilized in this study was obtained from Bahawal Victoria Hospital in Bahawalpur. It includes a total of 50 X-ray pictures of human lower leg bone fractures. Waseem Abbas et al. used 30 photos for training and 20 images for validation in their work. In this study, they suggested a transfer learning FasterRCNN deep learning model, and they were able to reach an mAP (mean Average Precision) of 96%.

Our analysis of various works in the field of fractures detection has revealed significant variations in datasets, their sizes, and their availability. We observed that different studies utilize diverse datasets for example [65] used 175 femur X-rays and [64] used 2,359 plain wrist radiographs, making it challenging to establish a unified benchmark. Furthermore, the accessibility of certain datasets remains limited for example the GAZ and MURA datasets are publicly available while the rest are not available for public use. Additionally, we noted that different classes within these datasets exhibit varying levels of complexity for example we have the hips [62] and wrist [63], requiring specialized approaches for

accurate detection. These findings underscore the need for standardized datasets and comprehensive evaluation metrics to facilitate fair comparisons and advancements in fracture detection algorithms. We may draw the conclusion from all of these studies that the use of YOLO was successful in achieving the desired results for the detection of bone fracture. The consistent utilization of YOLO across multiple studies underscores its effectiveness and establishes it as a reliable framework for bone fracture detection. Because of this, we deemed it necessary to base our study on it.

2.9 Conclusion

In this chapter, we went over the basics of machine learning methodologies as well as the basics of deep learning, After that, we discussed transfer learning and the various strategies associated with it, and then we went over the convolutional neural network and the various layers it possesses. In addition, we discussed the architecture of a few well-known CNNs. In conclusion, we provided an overview of the current state of the art. This review includes pertinent publications on the methods utilized for bone fracture detection as well as the results obtained by using those methods.
In the following chapter, the presentation of our bone fracture system and its implementation will be made.

Chapter 3

Design and implementation of Bone fractures detection system

3.1 Introduction

In our project, we aim to build a convolutional neural network that can detect bone fractures in X-ray images. In this chapter, we discuss our system's comprehensive design development and implementation. The first section presents the overall and detailed design of our detection system. Whereas the second one is about the implementation details which are common tools, frameworks, and libraries used to realize our system As well as how we implemented it.

3.2 Design of our Bone fractures detection system

Both the conceptual study and the design of our system are presented in this section. To begin, we will describe the overall architecture of our system, focusing on the primary functions that it provides. After that, we will present the detailed version, in which each step of the process will be explained separately.

3.2.1 General architecture

Our system will follow certain steps in order to construct a deep-learning model for

detecting fractures. The system begins by collecting the dataset, applying preprocessing and splitting it. After that we input the split dataset to the YOLO model, resulting in a model that is accurate. Then, our model can detect the fracture on a given image as shown in the following Figure 3.1.

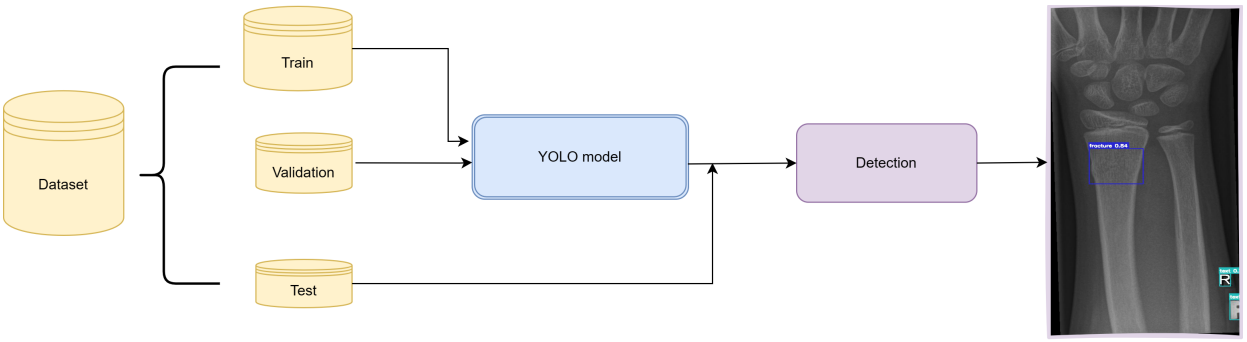


Figure 3.1: The general architecture of our system.

3.2.2 Detailed architecture

The architecture of our bone fractures detection system involves several key steps to effectively identify fractures in X-ray images. Firstly, we gather a comprehensive dataset comprising a diverse range of X-ray images. This dataset serves as the foundation to train our model. Next, we apply preprocessing techniques to enhance image quality and normalize the data, ensuring optimal input for the subsequent stages. Then, we class removal to remove class with a small number of data so it does not affect our overall model performance.

To evaluate the performance of our model, we split the dataset into training, validation and testing subsets, enabling us to train and fine-tune the model while also assessing its capabilities. The split dataset is then fed into the YOLO (You Only Look Once) model, a state-of-the-art object detection algorithm. YOLO analyzes the X-ray images at a holistic level, detecting and localizing potential fracture regions accurately and efficiently. This detection stage enables the identification of fractures within the X-ray images, forming a crucial component of our bone fracture detection system. as shown in the following figure 3.2.

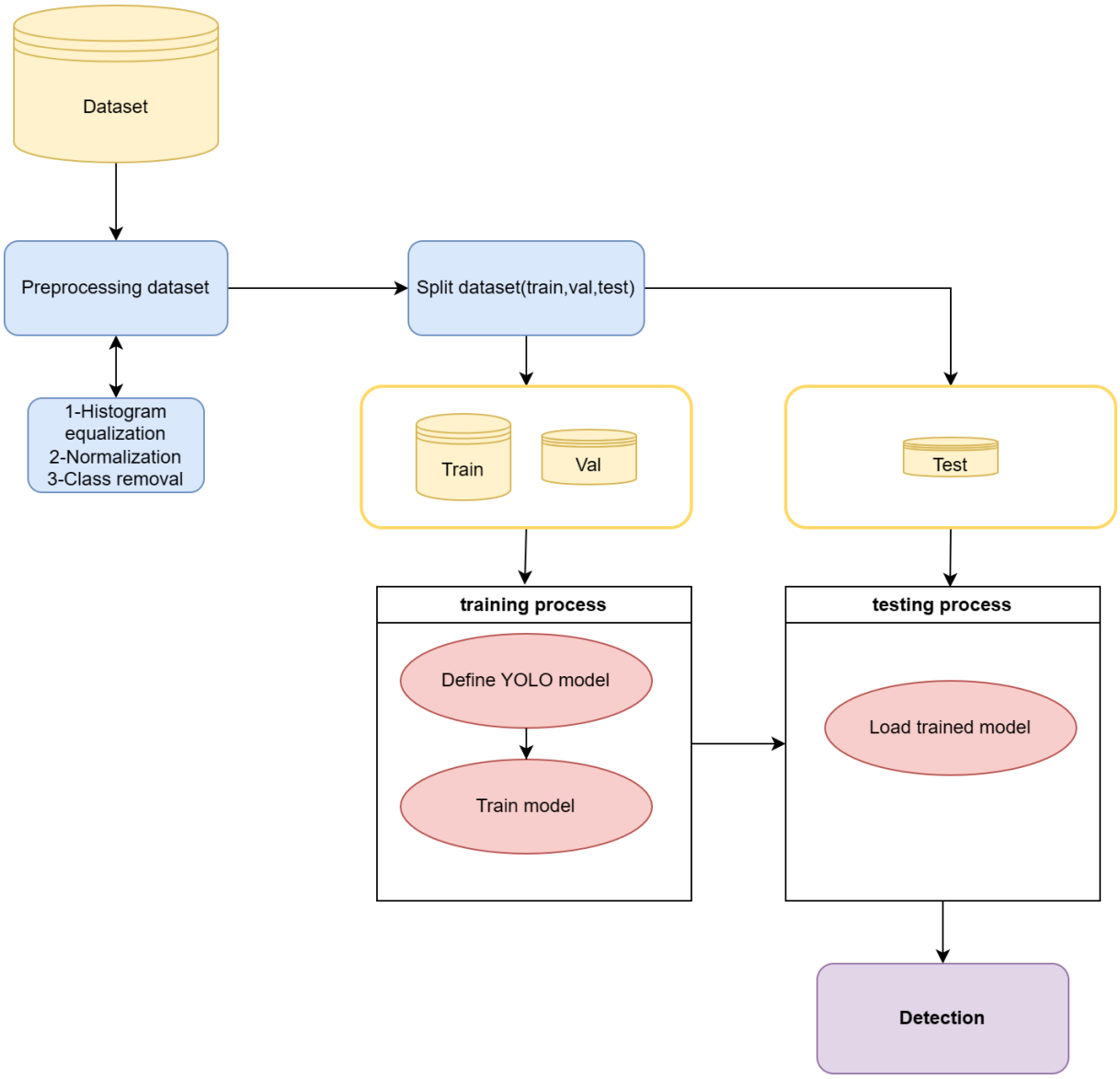


Figure 3.2: Detailed architecture of our system.

3.2.3 YOLO architecture

YOLO stands for You Only Look Once, it is the fastest and most precise real-time object detector. YOLOv7 is a single-stage real-time object detector. By improving upon its previous performance, YOLOv7 created a notable benchmark.

The greatest accuracy of 56.8% AP(average precision) among all real-time object detectors known, YOLOv7 excels all other known object detectors in terms of speed and accuracy in the range of 5 to 160 frames per second. YOLOv7 also outperforms R-CNN by 551% in terms of speed and 0.7% in AP. Without using any additional datasets or

pre-trained weights, YOLOv7 was purely trained on the MS COCO dataset from scratch.

There are three primary components that make up the YOLO framework [68], that are:

- **Backbone:** The Backbone is a pre-trained network used primarily to extract essential features of a picture and transmits them through the Neck to the Head.
- **Neck:** The Neck collects feature maps that the Backbone has retrieved and builds feature pyramids, this makes it easier for the model to generalize to objects of various sizes and scales.
- **Head:** The head contains output layers with final detections. where it is employed to carry out the stage's last operations, it generates the end result, which includes classes, objectness scores, and bounding boxes, by applying anchor boxes to feature maps.

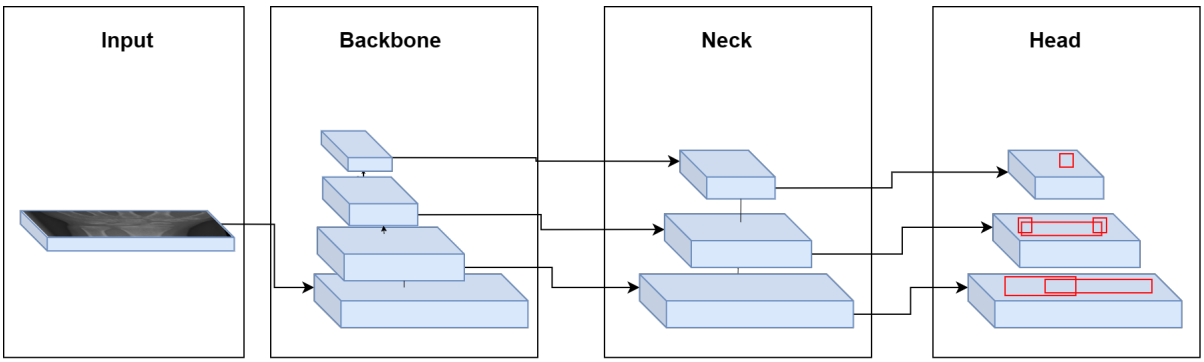


Figure 3.3: YOLO Architecture [68].

3.2.3.1 Extended Efficient Layer Aggregation(E-LAN)

For effective inference speed, the convolutional layers in the backbone of the YOLO network must be efficient. therefore the authors of YOLOv7 extend on previous studies in this area, taking into account both the memory requirements for maintaining layers in memory and the length of time it takes for a gradient to back-propagate across layers. Their network will be able to learn more effectively the smaller the gradient. They settle on E-ELAN, an extended version of the ELAN computational block, as their final layer aggregate.

To put it simply, E-ELAN architecture allows the framework to learn more effectively [68].

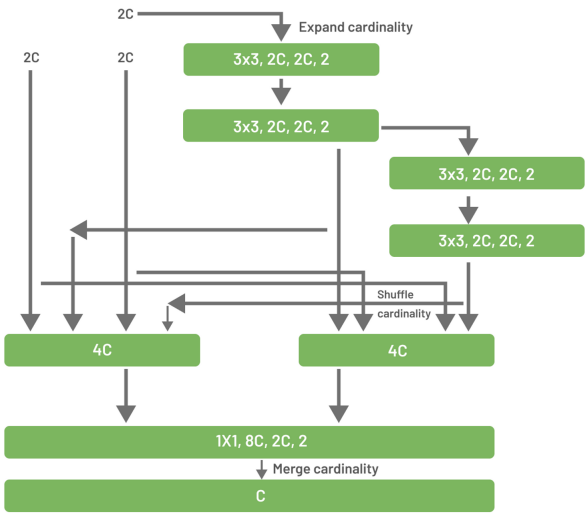


Figure 3.4: Extended Efficient Layer Aggregation Network (E-ELAN) [68].

3.2.3.2 Auxiliary Head Coarse-to-Fine

As we know by now the head contains the predicted outputs, however, because it's so far in the network, it may be useful to add an auxiliary head that is positioned in the centre. The Lead Head is the one in charge of the end prediction. And the Auxiliary Head is the head that helps with middle-layer training. This is being done since the lead head has a rather good capacity for learning. Therefore, the soft label that is produced from it should be a better representation of the distribution and correlation between the target and the source data. The lead head will be more capable to concentrate on acquiring the remaining information that has not yet been learned by enabling the shallower auxiliary head consequently learn the information that the lead head has learned.

To put it in simple terms the "coarse-to-fine" aspect alludes to the object detection process's hierarchical structure. It can be difficult to recognize items effectively using a single-scale technique in object identification since objects might differ considerably in size, shape, and appearance. A coarse-to-fine technique includes examining the picture at various sizes or degrees of detail in order to incrementally improve item detection[68].

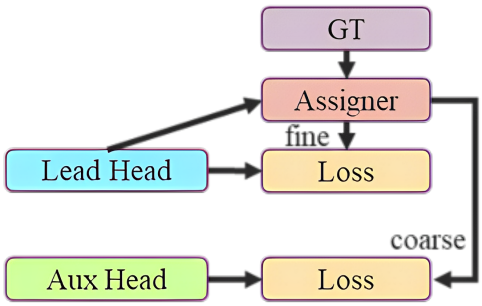


Figure 3.5: coarse-to-fine lead head guided assigned [68].

3.2.3.3 Loss functions

In the YOLOv7 model, several loss functions are utilized to train and optimize the performance of the object detection algorithm. The primary loss functions used in YOLOv7 includes:

- **Box loss:** for evaluating the precision of the predicted bounding box coordinates, YOLOv7 uses localization loss, which is typically calculated with Smooth L1 loss. which is the differences between the predicted bounding box and the actual bounding box are penalized by this loss. The box loss is calculated as follows:

$$loss = smooth\_L1\_loss(box\_pred, box\_gt) \tag{3.1}$$

where box_pred is the predicted bounding box and box_gt is the ground truth bounding box [69].

- **Objectness Loss:** this loss function, which is frequently implemented as binary cross-entropy, assesses how well the model predicts whether an object will be found in a particular grid cell. When the model predicts the presence or absence of an object incorrectly, it is penalized. and it's defined with the following equation :

$$loss = -log(objectness\_pred) \tag{3.2}$$

where objectness_pred is the predicted objectness score. The objectness score is the probability that there is an object in the region. The objectness loss is minimized

when the predicted objectness score is close to 1 for regions that contain objects and close to 0 for regions that do not contain objects [70].

- **Classification Loss:** the categorical cross-entropy method is typically used to calculate the classification loss, which evaluates the precision of object classification within each grid cell. By comparing the predicted class probabilities with the actual class labels, it penalizes incorrect class predictions. The classification loss is calculated as follows:

$$loss = cross\_entropy\_loss(class\_pred, class\_gt) \tag{3.3}$$

where class_pred is the predicted class probabilities and class_gt is the ground truth class labels [71].

3.2.4 Dataset preparation

Before feeding the data to the deep learning model, the initial step is to prepare the data.

3.2.4.1 Collecting data

Collecting data for both training and testing is the initial step in our learning system. The dataset used is an annotated pediatric wrist trauma radiography dataset (GRAZPEDWRI-DX). The acronym is composed of the terms “Graz”, “Pediatric”, “Wrist”, and “Digital X-ray”.

The wrist, which joins the hand to the forearm is a complicated joint. It consists of five metacarpal bones in the hand, two long bones in the forearm (the radius and ulna) and eight carpal bones. Two rows of four bones each make up the carpal bones. The scaphoid, lunate, triquetrum and pisiform bones are found in the proximal row of carpal bones. The trapezium, trapezoid, capitate, and hamate bones make up the distal row of carpal bones [72].

Figure 3.6: Wrist bones [73].

- **Scaphoid:** is a boat-shaped bone in the wrist situated under the thumb [74].
- **Lunate:** is a crescent-shaped bone It is situated between the ulna and radius [75].
- **Trapezium:** a rounded, square-shaped bone located on the thumb side of the wrist [76].
- **Trapezoid:** a trapezoid is a quadrilateral next to a trapezium that has one set of parallel sides [74].
- **Capitate:** is the largest of the carpal bones in the wrist, It is located in the middle of the wrist [74].
- **Hamate:** is a carpal bone located on the ulnar side of the wrist [74].
- **Triquetrum:** is a pyramidal-shaped carpal bone It is on the ulnar side of the hand [77].
- **Pisiform:**is a small, pea-shaped bone that helps stabilize the wrist joints [74].

GRAZPEDWRI-DX dataset was constructed from image data (pediatric wrist radiographs), natural language (report texts) and human expert annotations (bounding boxes, lines, polygons, and image tags). Annotations were established directly based on the X-ray contents and with the aid of the corresponding free text reports, finally combined to the whole dataset [78]. Board-certified pediatric radiologists (S.T., E.N., and

E.S.) with experiences ranging from 6 to 29 years in musculoskeletal radiology performed the image labelling. These radiologists approved all image annotations produced on the Supervisely (Deep Systems LLC, Moscow, Russia) artificial intelligence online platform. The web-based picture database was housed on a dedicated server, which made it possible for several users to contribute labels. Customers connected to the server through the Internet and logged in using a web browser. In addition to the validated radiologists that were stated, local radiologists, visiting colleagues, and medical students all contributed to the advancement of the dataset by labelling different times and shares of the dataset. Every single annotation was carried out during the months of March 2018 and February 2022.

A pool of 20,711 digital wrist radiographs were examined between 2008 and 2018 to determine their eligibility. Due to an improper field of vision that was not focused on the wrist, 384 photos were eliminated. There were 20,327 tagged and annotated images. The annotators did everything by hand, using specialized instruments where required. they used bounding boxes to annotate, among other things, fractures, metal implants, periosteal responses, and bone diseases. Image tags were manually defined to describe the characteristics of each image.

The dataset in question is available in ZIP files, and its total size is 15.2 gigabytes (GB) [78].

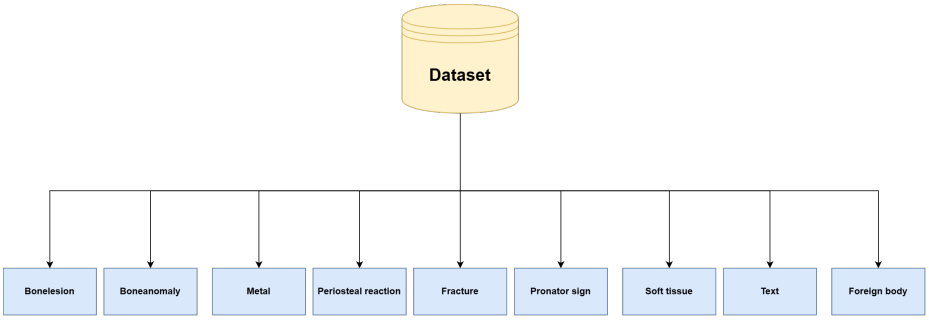


Figure 3.7: Dataset classes.

- **Bonelesion:** any abnormal or unusual region of bone tissue is referred to as a bone lesion. It may show up as a structural change, such as a region of decreased bone density or an area of abnormal bone growth [79].

- **Boneanomaly:** is a general word that describes any variation or abnormality in the growth or structure of wrist bone. It might include a variety of conditions and abnormalities that affect the bone.
- **Metal:** are used in broken bones for internal fixing implants and they are made of titanium and stainless steel [80].

- **Periosteal reaction:** when cortical bone reacts to one of several potential injuries, a periosteal reaction occurs. The periosteum may extend from the cortex and take on different patterns of periosteal response as a result of trauma [81].

- **Fracture:** a rupture in the continuity of bone tissue, whether total or partial is known as a bone fracture [3].
- **Pronator sign:** it can be an unseen indicator of a distal forearm injury. It is predicated on the displacement of the fat pad, which is visible on a lateral wrist radiograph in the following figure, and it's located just superficial to the pronator quadratus muscle[82].

- **Soft tissue:** a wrist injury around a muscle or tendon is referred to as a "soft tissue injury", The following figure shows joint soft-tissue swelling [83].

- **Text:** "L" and "R" letters are used to indicate the left and right sides of the body part being imaged.
- **Foreign body:** when an object or substance is present inside the body that is not ordinarily there, it is referred to as a "foreign body" on X-rays. Small things like metal shards, glass shards, or wood splinters are examples of foreign bodies.

Object	One or more objects present in the image
boneanomaly	n=192 (0.94%)
bonelesion	n=42 (0.21%)
foreignbody	n=8 (0.04%)
fracture	n=13,550 (66.66%)
metal	n=708 (3.48%)
periostealreaction	n=2,235 (11.00%)
pronatorsign	n=566 (2.78%)
softissue	n=439 (2.16%)
Text	n=20,274 (99.74%)

Figure 3.8: The quantity and proportion of annotated objects. Each image may be assigned to many objects [78].

3.2.4.2 Pre-processing

Before providing data to our model, certain image processing methods can be applied in order to prepare the data. In our work, the preprocessing step includes histogram equalization and normalization.

- **Contrast Limited Adaptive Histogram Equalization (CLAHE):** histogram equalization is a computer image processing method used to increase picture contrast. This is achieved by successfully extending the intensity range of the picture and spreading out the most common intensity levels. When the useful data is represented by near-contrast values, this strategy often boosts the overall contrast of the photos. This makes it possible for regions with less local contrast to acquire more contrast [84].

- **Normalization :** it is a method for standardizing a set of independent variables or data properties. normalization of the relative pixel number per grayscale level from

0 to 1 according to the equation 3.4.

$$F(x) = \frac{x - min(x)}{max(x) - min(x)} \tag{3.4}$$

- **class removal:** from 3.8 we excluded 4 classes due to the small number of images that each class has. the classes are Bonelesion, Boneanomaly, Foreign body, and Soft tissue.

3.2.4.3 Data loading

- **Data labelling:** the data we labelled into 9 classes in the YOLO format.

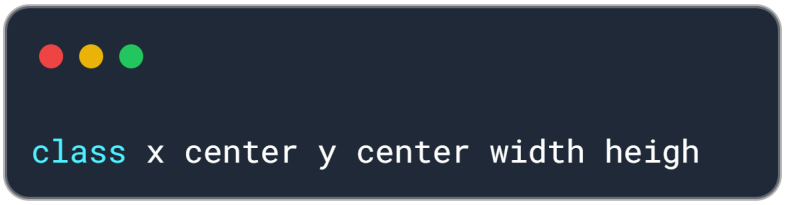


Figure 3.9: File in the YOLO format.

Each row is class, x_center y, _center, width, height format, where class represents the class number and the rest is the box coordinates.

- **Data Splitting :** we divided the dataset into a 7:2:1 ratio, which means 70% of the data is used for training, 20% for validation, and 10% for testing.

3.2.4.4 Detection

After the training phase, our YOLO model is ready to detect fractures, We will utilize the third subset from the dataset splitting to test our YOLO model.

3.2.4.5 Evaluation of the YOLO model

In order to assess model performance we need to compute some metrics, particularly for

multi-class data, performance metrics like precision, recall, and F1-score were employed. Four factors are used to define the performance metrics:

1. true positive (**TP**): a data point that the model correctly labeled as positive and it's truly positive (correct).

2. false positive (**FP**): a data point labeled as positive but is actually negative (incorrect).

3. true negative(**TN**): a data point that the model correctly labeled as negative and it's truly negative (correct).

4. False negative (**FN**): the model labels a data point as negative when it is actually positive.

    - **Precision:** it measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive, It focuses on the accuracy of positive predictions.

$$PRECISION = \frac{TP}{TP + FP} \tag{3.5}$$

    - **Recall:** it determines the proportion of all positive cases that were properly predicted to be positive (true positives). contrary to precision, which only addresses the accurate positive predictions among all positive predictions, Recall indicates any missing positive predictions.

$$RECALL = \frac{TP}{TP + FN} \tag{3.6}$$

    - **Precision/Recall:** illustrates the trade-off between precision and recall for each class.

    - **F1 score:** the overall balance between precision and recall.

$$F1SCORE = \frac{TP}{TP + \frac{1}{2}(FN + FP)} \tag{3.7}$$

3.3 Implementation of our deep learning model

3.3.1 Hardware configuration

Our hardware configuration is a Lenovo Thinkpad laptop of the following characteristics:

— Processor : Intel(R) Core(TM) i7-7600U.

— Processor Frequency: 2.90 GHz.

— RAM: 8 Go.

— Hard drive: 256Go (Solid state drive (SSD)).

— Graphics: HD Graphics 620

3.3.2 Frameworks, tools, and libraries

- **Python :** is now the programming language with the highest rate of growth due to its user-friendliness, quick learning curve, and plenty of high-quality packages for data science and machine learning [85]. It was created to be clear to read and efficient. A Dutch programmer named Guido van Rossum invented Python in 1991 [86].

- **Google Colab:** the Google research project known as Collaboratory, or "Colab" for short, was intended to aid students, data scientists, and researchers working in artificial intelligence (AI) with the teaching and study of machine learning. Google Colab is a free environment for running Jupyter notebooks that does not require any configuration and operates totally in the cloud. With Google Colab, it is possible to develop and execute code, store and share our studies, and have access to powerful computational resources, all without having to download any software or install any plugins. The Google Colab cloud platform is necessary for this activity since the training of a deep learning model often requires a comprehensive CPU and GPU configuration [87].

- **Pytorch**: originally created by Meta AI and now a part of the Linux Foundation, PyTorch is a machine learning framework built on the Torch library and used for applications like computer vision and natural language processing. And it is open-

source software [88].

- **Torch:** the torch package implements mathematical operations on multidimensional tensors and provides data structures for these tensors. Additionally, it offers a variety of helpful functions for the effective serialization of arbitrary types and Tensors [88].

- **Matplotlib:** a Python plotting tool called Matplotlib creates publication-quality graphics on a range of platforms and even interactive settings. The Matplotlib Python library provides an object-oriented API for embedding plots in GUI applications. It may be used in scripts, the python and ipython shells, and even web application servers. For our project, we plot graphs using it[89].

- **Numpy:** is a Python module for handling arrays. It contains tools for handling a high-performance multidimensional array object. It's the most crucial Python module for scientific computing [90].

- **Flask:** it is a web framework. This implies that you can create a web application using the tools, frameworks, and technologies offered by Flask [91].

- **OpenCv:** open Source Computer Vision Library is a free computer vision and machine learning software library. OpenCV was developed to assist commercial products incorporate machine perception more rapidly and to offer a standard foundation for computer vision applications [92].

- **scikit-learn:** the scikit-learn is a python open-source machine learning. Researchers, engineers, and data scientists all use it, making it one of the most widely used machine learning libraries in the world. from this library, we used GroupShuffleSplit class. GroupShuffleSplit is a cross-validation splitter that shuffles the data within groups, while preserving the relative ordering of the samples within each group. This can be useful for cases where the data is naturally grouped, such as when the data is collected from different users or different time periods [93].

- **Scikit-image:** scikit-image, also referred to as skimage, is a free and open-source Python tool created for image preprocessing [94].

3.3.3 Dataset preparation and preprocessing

- **Contrast Limited Adaptive Histogram Equalization(CLAHE)**:

  Rescale_intensity return image after stretching or shrinking its intensity levels. A local contrast enhancement approach uses histograms calculated over several tile sections of the image. Therefore, local details can be improved even in areas that are darker or brighter than the majority of the image.

```
1 from skimage import exposure
2 import numpy as np
3 img = exposure.rescale_intensity(img, in_range=(np.percentile(img, intensity_crop),
4                                         np.percentile(img, (100-intensity_crop))))
5 img = exposure.equalize_adapthist(img)
```


Figure 3.10: Pseudo code for histogram equalization(CLAHE).

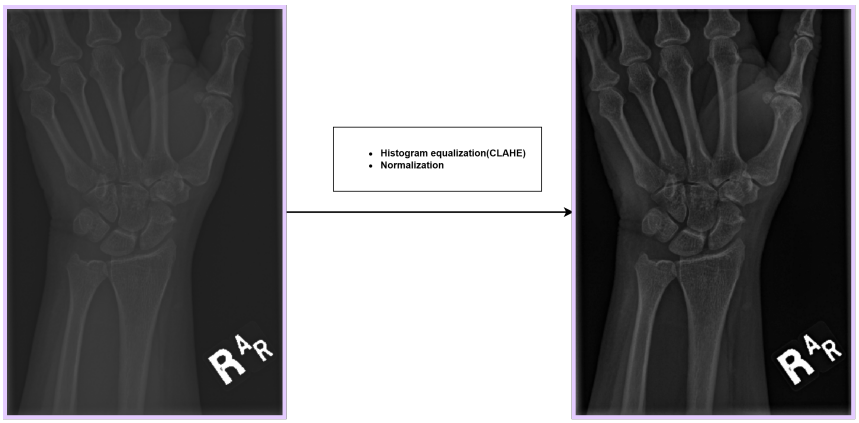


Figure 3.11: Applying histogram equalization(CLAHE) filter and normalizing the picture.

- **Normalization** : normalizing each image using cv2. normalize module.

```python
import cv2
import numpy as np
if outputbitdepth == 8:
 img = (img / 255.0).astype(np.float64)
 img = cv2.normalize(img, dst=None, alpha=0, beta=int((pow(2, outputbitdepth))-1),
                                    norm_type=cv2.NORM_MINMAX).astype(np.uint8)
```

Figure 3.12: pseudo code for Normalization.

- **Class removal:** after training the model we noticed poor results from the following classes: Bonelesion, Boneanomaly, Foreign body and Soft tissue. So we removed the classes by deleting the labels of the corresponding class.

```python
# Filter out labels to delete
filtered_lines = [line for line in lines if line.split()[0] not in labels_to_delete]
# If any labels were deleted, overwrite the label file
if len(filtered_lines) < len(lines):
    with open(label_path, "w") as file:
        file.writelines(filtered_lines)
# List the labels you want to delete
labels_to_delete = ["0", "1", "2","7"]
```

Figure 3.13: Pseudo code for deleting labels of classes that we want to remove.

```python
import os
            # Replace class labels with new numbers
            modified_lines = []
            for line in lines:
                label = line.split()[0]
                if label in label_mapping:
                    new_label = str(label_mapping[label])
                    modified_line = line.replace(label, new_label)
                    modified_lines.append(modified_line)
            # If any labels were modified, overwrite the label file
            if modified_lines:
                with open(label_path, "w") as file:
                    file.writelines(modified_lines)
# Define the mapping of old label numbers to new label numbers
# Example: {"old_label1": 0, "old_label2": 1, "old_label3": 2}
label_mapping = {"3": 0, "4": 1,"5": 2, "6": 3, "8": 4}
```

Figure 3.14: pseudo code for reordering the rest of classes labels.

- **Splitting dataset**: the data was split according to PatientID. To split our dataset into three subsets(Train, validation, testing), we used the GroupShuffleSplit module from the sklearn library.

```python
import pandas as pd
from sklearn.model_selection import GroupShuffleSplit
df = pd.read_csv("/content/dataset.csv")
splitter1 = GroupShuffleSplit(test_size=.3, n_splits=2)
split = splitter1.split(df, groups=df["patient_id"])
train_idxs, valid_idxs = next(split)
train_df = df.iloc[train_idxs]
temp_df = df.iloc[valid_idxs]

splitter2 = GroupShuffleSplit(test_size=.3333, n_splits=2)
split = splitter2.split(temp_df, groups=temp_df["patient_id"])
valid_idxs, test_idxs = next(split)
valid_df = temp_df.iloc[valid_idxs]
test_df = temp_df.iloc[test_idxs]
```

Figure 3.15: Pseudo code for Dataset splitting.

```
100%|██████████| 14041/14041 [00:01<00:00, 12577.69it/s]
100%|██████████| 4185/4185 [00:00<00:00, 13201.84it/s]
100%|██████████| 2101/2101 [00:00<00:00, 12082.38it/s]Data split compleated according to PatientID:
  - 14041 (69.076%) images in the training set.
  - 4185 (20.588%) images in the validation set.
  - 2101 (10.336%) images in the testing set.
```

Figure 3.16: Spliting result.

3.3.4 Building our YOLO model

1. **Download pre-trained weights**

```
!wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt
```

Figure 3.17: command to download YOLOV7 pre trained weights.

2. **Defining our "meta.yaml" file** : in this file, we specify the path to our dataset, the number of classes, and the name of the classes.

```
train: /content/yolov7/yolov5/images/train
val: /content/yolov7/yolov5/images/valid
test: /content/yolov7/yolov5/images/test
# number of classes
nc: 5
# class names
names: ["fracture", "metal", "periostealreaction", "pronatorsign", "text"]
```

3. **Model summary:** this shows the contents and parameters of the model.

```
Model Summary: 415 layers, 37218132 parameters, 37218132 gradients, 105.2 GFLOPS
```

Figure 3.18: Model summary.

4. **Training YOLO model:** here we specified the image size 640*640 so images can be resized automatically.

```
%cd /content/yolov7
!python train.py --batch 16 --cfg yolov5/yolov7_cfg.yaml
--epochs 10 --data yolov5/meta.yaml --img 640 640   --weights yolov7.pt
```

3.4 Conclusion

This chapter provides a comprehensive overview of our system design, delving into the step-by-step process, the tools, libraries, and frameworks employed throughout, as well as the detailed implementation of a large section of our system, including our YOLO model. The following chapter will go over various experiments and their results.

Chapter 4

Experimental Results and Discussion

4.1 Introduction

The preceding chapter covered our system architecture, showed how the dataset we utilized had been structured and supplied the code for each component of our bone fractures detection system. In this chapter, We will explain our experimental results, and also we will provide a comparison and a discussion of our obtained results.

4.2 The obtained results for fractures detection

4.2.1 Before class removal

The evaluation of our model's performance revealed some disappointing results across multiple classes. As seen in the Figure 4.1, the class "boneanomaly" exhibited a particularly low precision/recall value of 0.267, indicating significant challenges in accurately detecting bone anomalies. Similarly, the class "bonelesion" had a precision/recall value of 0.00, suggesting that the model struggled to identify any instances of bone lesions, yielding no positive predictions. The class "softtissue" also fared poorly, with a precision/recall value of 0.146, highlighting a considerable difficulty in detecting soft tissue abnormalities. While some classes showcased relatively better performance, such as "fracture" with a precision/recall value of 0.90 and "metal" with a value of 0.948, there were still areas for

improvement. Additionally, the classes "periostealreaction" and "pronatorsign" had precision/recall values of 0.462 and 0.48, respectively, indicating moderate success but leaving room for enhancement. On a positive note, the class "text" demonstrated a promising precision/recall value of 0.989, indicating the model's strong ability to accurately recognize and interpret textual elements. These results shed light on the areas where the model falls short, emphasizing the need for further refinement and optimization to enhance its overall performance.

The poor performance of the model in certain classes can be attributed to two main factors: the lack of sufficient training data and the complexity of the disease being detected. The number of images available for each class further underscores these challenges. For instance, the class "boneanomaly" with 192 images and "bonelesion" with only 42 images may not provide an ample representation of the wide range of anomalies and lesions that can occur in bones. Similarly, the class "foreignbody" with just 8 images and "softtissue" with 439 images may not adequately cover the diverse variations and appearances of foreign bodies and soft tissue abnormalities. Insufficient training data can limit the model's ability to learn and generalize effectively, leading to a subpar performance in those specific classes. Additionally, the complexity of the disease itself, especially in cases involving bones and soft tissue, can present challenges in accurate detection due to variations in size, location, and appearance. Overcoming these limitations would require a larger and more diverse dataset, encompassing a broader spectrum of cases, to enhance the model's understanding and discriminatory power within these challenging classes. That's why we opted for class removal to remove classes that showed poor results.

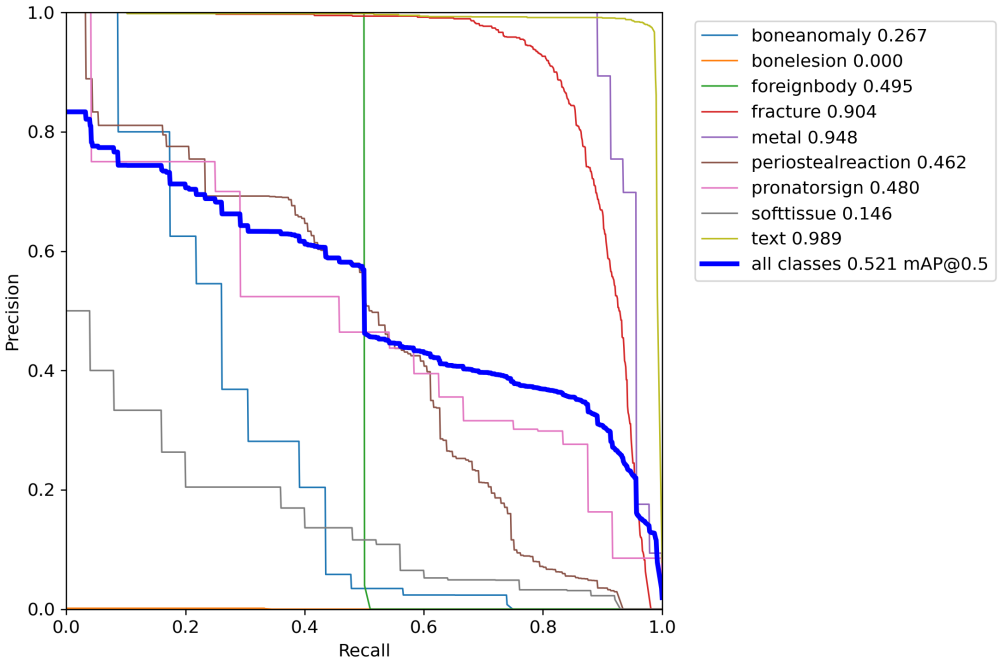
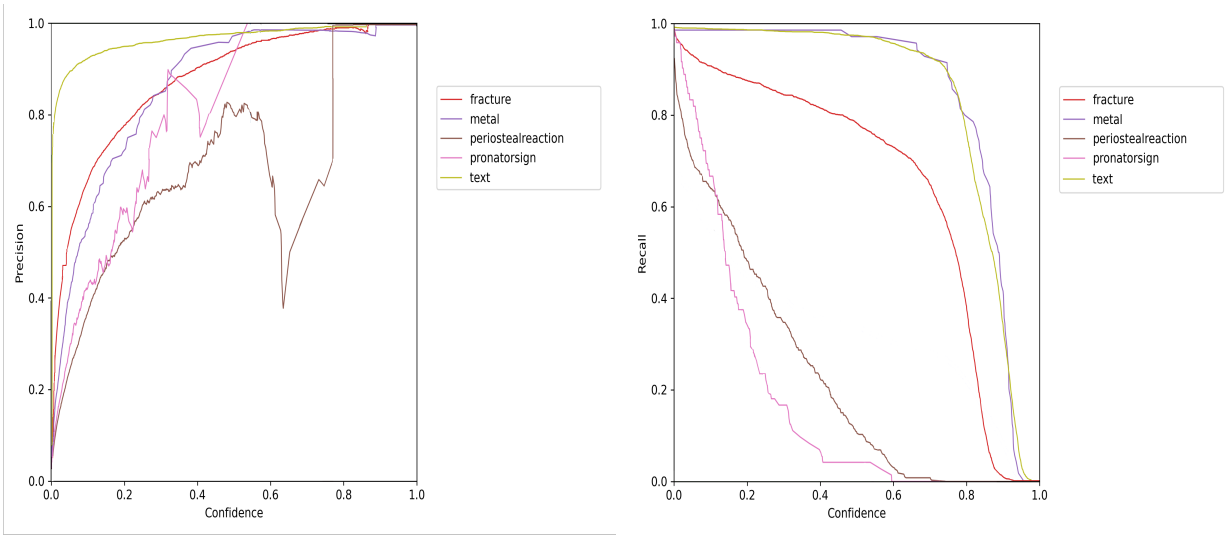


Figure 4.1: Precision/Recall curve.

4.2.2 After class removal

4.2.2.1 Precision and recall

After class removal, as seen in the Figure 4.3, a little enhancement in the model performance. The precision/recall curve analysis for the multi-class detection model yielded insightful results. For the class "fracture" which is the main focus of this project, the model achieved an impressive precision/recall of 0.910, indicating a high level of accuracy in detecting fractures. The class "metal" exhibited good performance also, with a precision/recall of 0.978, showcasing the model's strong ability to identify metallic objects. However, the class "periosteal reaction" had a lower precision/recall of 0.459, suggesting that the model had more difficulty in accurately detecting this particular feature. Similarly, the class "pronator sign" had a precision/recall of 0.511, due to small data. On a positive note, the model excelled in the class "text" with a precision of 0.991. These results from the precision/recall curve provide valuable insights into the model's performance for each class, helping to identify areas of strength.



Precision curve Recall curve

Figure 4.2: Precision and recall performance of the model.

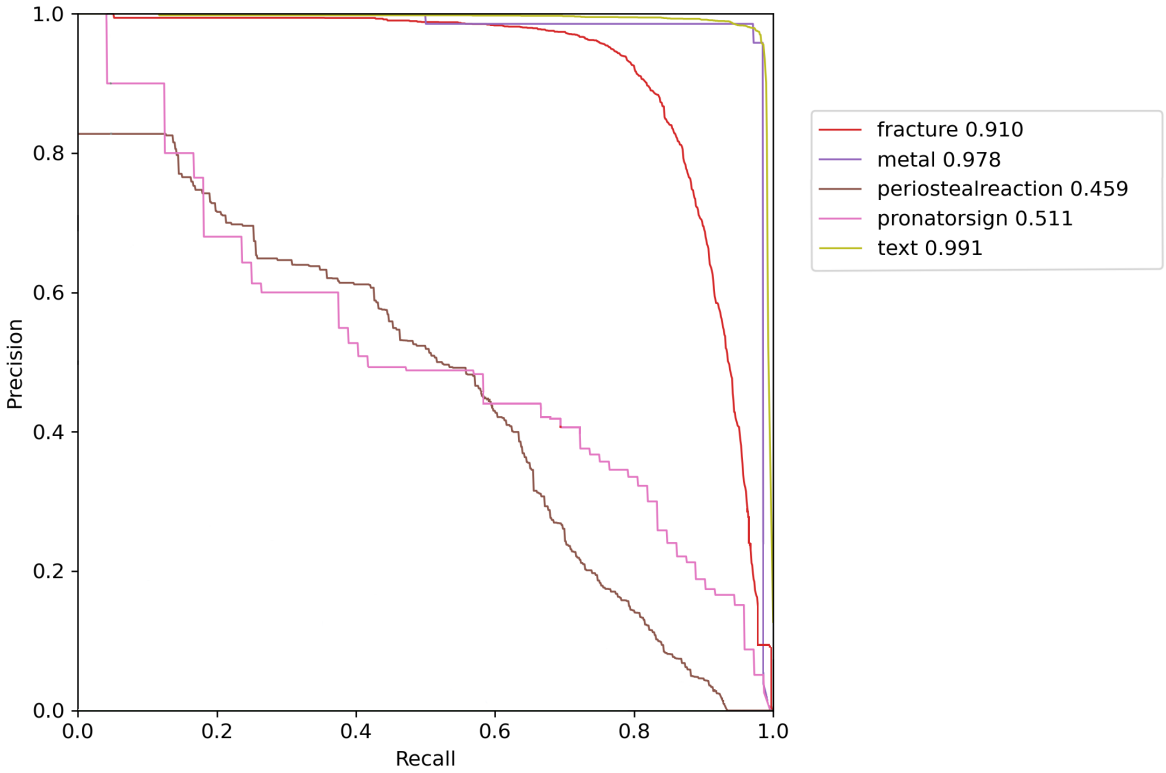


Figure 4.3: Precision/Recall curve.

4.2.2.2 F1 score

The F1 score results obtained by my model shed light on its performance, with a particular emphasis on the main class "fracture." The F1 score for the "fracture" class achieved an impressive value of 0.867, indicating a high level of accuracy in detecting fractures. This highlights the model's proficiency in identifying and localizing fractures within images. Additionally, the class "metal" exhibited excellent performance with an F1 score of 0.97, The F1 scores for "periostealreaction" and "pronatorsign" were 0.52 and 0.53, respectively, indicating moderate success in detecting these specific features. Notably, the class "text" demonstrated outstanding performance, achieving an F1 score of 0.98 (see Figure 4.4). These F1 score results demonstrate the model's effectiveness in various classes, particularly excelling in the "fracture" class, making it a key focus of the model's detection capabilities.

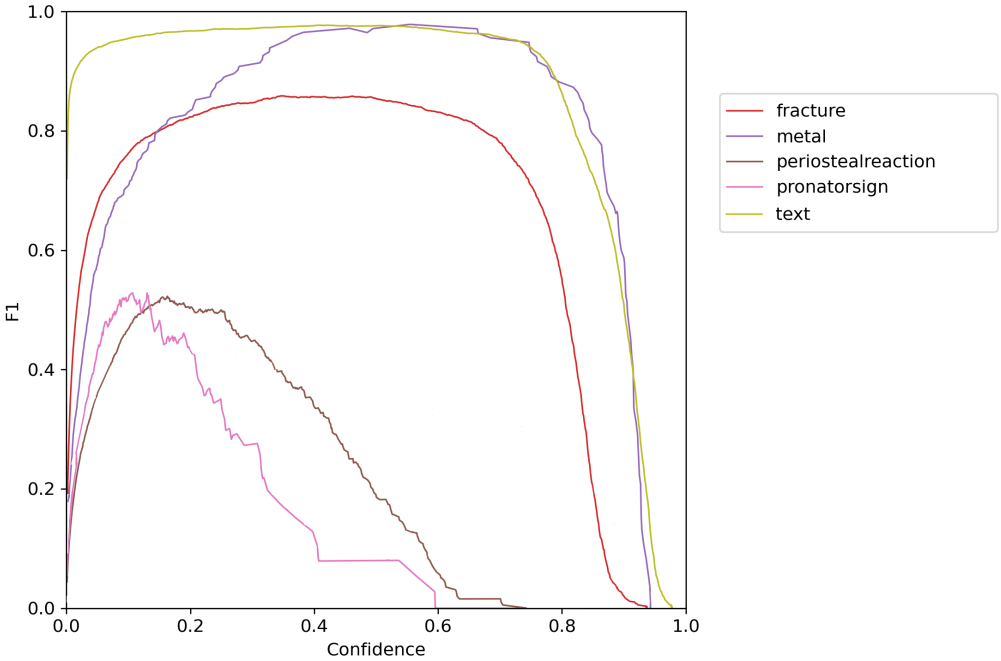


Figure 4.4: F1 score curve.

4.2.2.3 Confusion_matrix

The confusion matrix provides valuable insights into the performance of our model in classifying different classes, with specific attention to the main class, "fracture." Exam-

ining the results for the "fracture" class, we observed a true positive (TP) rate of 0.87, indicating that the model successfully identified 87% of the actual fractures present in the dataset. This demonstrates the model's proficiency in accurately detecting fractures from the X-ray images. Moving on to the "metal" class, we obtained an impressive true positive rate of 0.99, indicating a high level of accuracy in identifying metallic objects in the images. The model excelled in recognizing these objects, which is crucial for identifying potential sources of interference in X-ray scans. For the "periostealreaction" class, we achieved a true positive rate of 0.46, suggesting that the model correctly identified 46% of the instances exhibiting periosteal reactions. While this rate is lower than desired, it still demonstrates the model's ability to detect this specific type of abnormality. The "pronatorsign" class showed a true positive rate of 0.23, indicating that the model recognized 23% of the pronator signs present in the images. Although the performance for this class is relatively lower, it still provides valuable insights and can assist healthcare professionals in their diagnosis. Lastly, the "text" class demonstrated a true positive rate of 0.99, highlighting the model's exceptional capability to identify text elements within the X-ray images. This is particularly useful in scenarios where written information needs to be analyzed, such as identifying labels or annotations on the X-ray scans.

Overall, analyzing the confusion matrix results reinforces the strengths of our model in accurately detecting fractures and metal objects. While there is room for improvement in detecting periosteal reactions and pronator signs, the model's high performance in identifying fractures and text elements signifies its potential value in aiding radiologists and healthcare professionals.

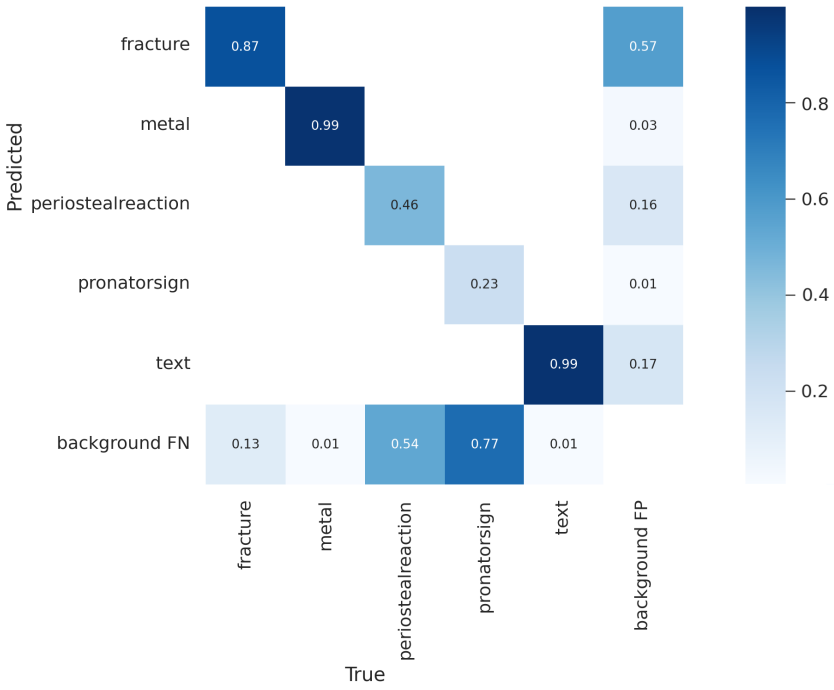


Figure 4.5: Confusion matrix.

4.2.2.4 Loss

The loss results obtained by my model during training and validation are indicative of its performance. During training, the model achieved impressive results with a box loss of less than 0.030, indicating accurate localization of bounding boxes. The objectiveness loss was remarkably low at 0.0059, signifying the model's proficiency in determining the presence or absence of objects within the bounding boxes. The classification loss was also minimal, measuring 0.00062, showcasing the model's ability to accurately classify objects within the detected regions (see Figure 4.6). Similarly, during validation, the model displayed promising performance. The validation box loss remained below 0.049, indicating good localization accuracy. The validation objectiveness loss was measured at 0.0215, demonstrating the model's capability to determine object presence reliably. The validation classification loss was 0.0115, further reinforcing the model's accurate classification skills (see Figure 4.7). These low losses across both training and validation highlight the model's effectiveness in detecting and classifying objects within images, providing confidence in its

overall performance.

Figure 4.6: Training loss.

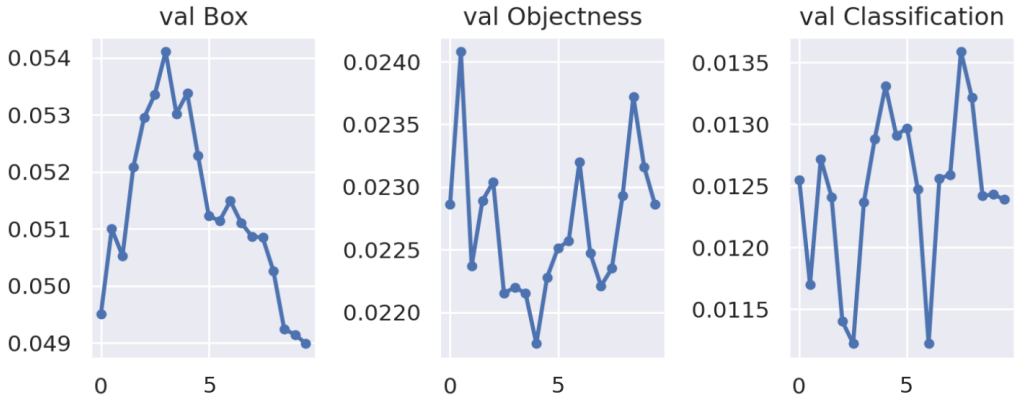


Figure 4.7: Validation loss.

4.2.3 Model results on Testing data

```
test: New cache created: /content/yolov7/yolov5/labels/test.cache
                 Class      Images      Labels           P           R      mAP@.5
               fracture        2030        1832       0.815       0.909       0.937
                  metal        2030          63       0.884       0.952       0.969
  periostealreaction        2030         316       0.619       0.547       0.565
         pronatorsign        2030          58       0.693       0.584       0.688
                  text        2030        2369       0.952       0.988       0.993
Speed: 11.7/2.6/14.4 ms inference/NMS/total per 640x640 image at batch-size 32
```

Figure 4.8: Result of our model on testing data.

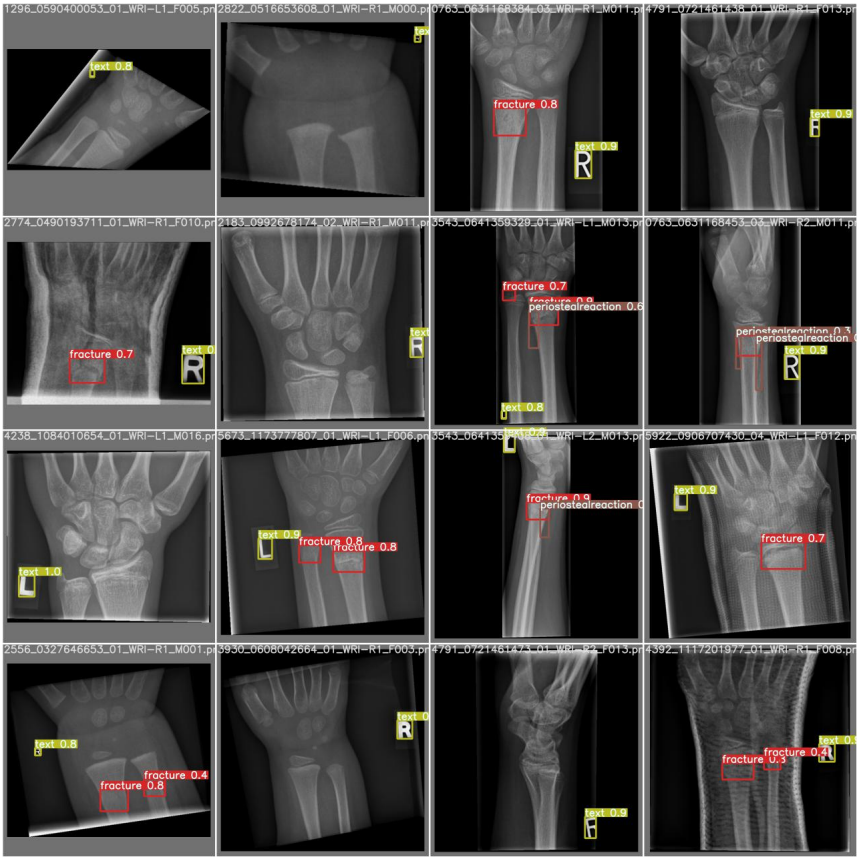


Figure 4.9: testing example number.

4.3 Results comparison

by comparing the results of our work using YOLOv7 with the work of Rui-Yang Ju et al [63] using YOLOv8, the mean average precision (mAP) serves as the metric of comparison. It is worth noting that our model utilizes a 640× 640 input size, while the model developed by Rui-Yang Ju et al [63] employs a larger input size of 1024x1024. Additionally, Rui-Yang Ju et al [63] modified the YOLOv8 architecture and utilized an RTX 3080 ti for training, whereas we conducted our training on Google Colab's free GPU. The results of our model demonstrate the following mAP values: fracture (0.937), metal (0.969), periostealreaction (0.565), pronatorsign (0.688), and text (0.993). On the other hand, Rui-Yang Ju et al [63] 's model achieved the following mAP values: fracture (0.945), metal (0.901), periostealreaction (0.722), pronatorsign (0.663), and text (0.990).

These results indicate that both models perform well in detecting fractures and

other classes of interest. While Rui-Yang Ju et al.'s model achieved slightly higher mAP scores in some classes, it is important to consider the differences in input sizes and hardware resources used for training. Overall, these comparisons showcase the effectiveness of YOLO-based models for fracture detection in X-ray images and highlight the potential for further improvement and optimization in future research endeavours.

4.4 System interface

Figure 4.10: The interface of our bone fractures system.

By clicking choose file we can choose the desired image to do detection on, then after clicking submit the result will be displayed, as shown in the figure 4.11 below.

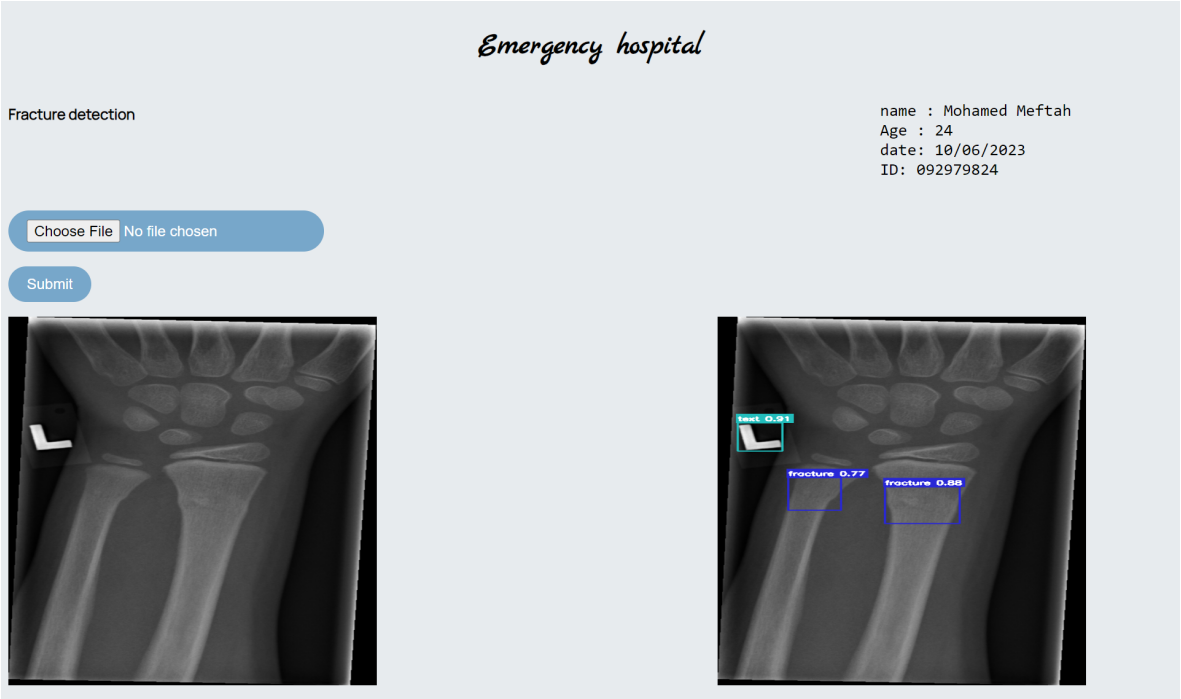


Figure 4.11: The interface of our bone fractures system.

4.5 Discussion

In the realm of fractures detection, there are three prominent publicly available datasets: MURA, LERA, and GAZ. The MURA dataset encompasses seven classes that cover different parts of the arm, including the elbow, finger, hand, humerus, wirst, shoulder, and wrist [95]. The LERA dataset focuses on knee X-rays [96], while the GAZ dataset consists of nine classes, including "boneanomaly," "bonelesion," "foreignbody," "fracture," "metal," "periostealreaction," "pronatorsign," "softtissue," and "text".

During the exploration of the MURA dataset, state-of-the-art performance was achieved using DenseNet169, yielding an impressive accuracy of 86%. It's important to note that both MURA and LERA datasets are primarily designed for classification purposes, distinguishing between normal and abnormal conditions rather than specifically identifying fractures. The term "normal or abnormal" is used as these datasets encompass various abnormalities, such as fractures, as well as other factors like text and metal. However, it was observed that during the training phase on the MURA dataset, the model tended to prioritize features related to text and metal. Since text appears in a significant portion of

the dataset, and metal objects are common in medical images, they garnered substantial attention during training. Consequently, the model's ability to precisely highlight fractures was compromised. During testing, we noticed:

1. Fracture are not precisely highlighted.

2. Text and metal are also highlighted.

In contrast, the GAZ dataset was designed explicitly for fracture detection. It labels fractures, metal, and text with their exact coordinates in the images, eliminating any confusion or ambiguity. Additionally, the inclusion of other classes like "periostealreaction" and "pronatorsign" in the GAZ dataset aids in the detection of fractures by providing additional context and reference points. Considering the characteristics of these datasets, the GAZ dataset proves more suitable for fracture detection due to its dedicated focus on this specific task and the availability of precise annotations. The presence of other relevant classes further enhances the model's ability to accurately identify fractures, making it a valuable resource for advancing fracture detection algorithms.

After removing the classes "boneanomaly," "bonelesion," and "softtissue" from the model, we observed an improvement in overall performance. This improvement can be attributed to several factors. Firstly, by removing these classes, we eliminated the complexity and variability associated with diseases related to bone abnormalities, lesions, and soft tissue abnormalities. These conditions often present intricate patterns and diverse manifestations, making accurate detection and classification challenging. By excluding these classes, the model can now focus on a narrower scope of detection, which allows for better specialization and understanding of the remaining classes. Furthermore, the removal of these classes, particularly "boneanomaly" and "bonelesion," reduces the potential confusion and overlap with the main class of interest, "fracture." Fractures can sometimes exhibit similarities or be misinterpreted as bone anomalies or lesions. By eliminating these confounding classes, the model can prioritize and refine its detection capabilities specifically for fractures, leading to improved accuracy and performance. Additionally, the removal of the "softtissue" class reduces the complexity associated with distinguishing soft tissue abnormalities from fractures, further enhancing the model's focus and precision. Overall,

by removing the classes "boneanomaly," "bonelesion," and "softtissue," the model benefits from a more focused and simplified task, allowing it to concentrate on the detection of fractures with greater accuracy and efficiency. This focused approach reduces confusion, minimizes potential misclassifications, and ultimately leads to better performance in fracture detection.

The classes "periostealreaction," "pronatorsign," and "metal" play crucial roles in enhancing the model's performance for fracture detection, thereby establishing a significant relationship between these classes and fractures. Firstly, the class "periostealreaction" is closely associated with fractures as it represents the body's response to a bone injury. Periosteal reactions often occur adjacent to fractures and manifest as thickening or new bone formation. By including this class in the model, we enable it to detect and identify such reactions, which can provide valuable contextual information and aid in accurate fracture detection. Similarly, the class "pronatorsign" is relevant to fractures as it specifically indicates a wirst bone injury involving the radius and ulna. This class is important for capturing distinctive features and patterns associated with wirst fractures. By incorporating it into the model, we enhance its ability to identify and differentiate fractures in this specific region, thus improving overall fracture detection accuracy. Additionally, the class "metal" is significant for fracture detection due to its relevance in orthopaedic procedures. Metallic implants, such as screws or plates used in fracture fixation, may appear in X-ray images. Including the "metal" class allows the model to differentiate between metallic objects and fractures, reducing potential false positives and enhancing the precision of fracture detection.

Retaining these classes, even if the model already performs well for fracture detection, as they provide valuable contextual information and distinct features closely associated with fractures. By considering the relationship between fractures and "periostealreaction," "pronatorsign," and "metal," the model can leverage these classes to improve its ability to accurately detect and classify fractures, ultimately enhancing its overall performance in fracture detection tasks.

4.6 Conclusion

In this chapter, we provided an overview and comprehensive explanation of the architecture of our detection model. We also discussed the various tools and packages essential for the successful implementation of our system. Additionally, we presented the results of our experiments in detail, showcasing the remarkable performance achieved in the medical domain. The outcomes obtained from our experiments demonstrate exceptional effectiveness and accuracy, further solidifying the significance of our model in the medical field.

General conclusion

Artificial intelligence (AI) has ushered in a transformative era in the field of medicine, with deep learning playing a pivotal role in revolutionizing healthcare practices. In the context of bone fractures detection in X-ray images, our study aimed to harness the power of AI to improve diagnostic accuracy and treatment outcomes. Employing the YOLO architecture, we designed a robust system tailored specifically for fracture detection. The model was meticulously trained and rigorously validated using X-ray images sourced from the comprehensive GAZ dataset.

The obtained results unequivocally demonstrated the effectiveness of our model in detecting fractures, with an impressive precision/recall score of 91%. This level of performance underscores the reliability and potential of our system in accurately identifying bone fractures in X-ray images. By leveraging AI and deep learning techniques, we have successfully advanced the capabilities of fracture detection, enhancing the speed and accuracy of diagnoses.

The importance of AI in medicine cannot be overstated, as it enables healthcare professionals to make informed decisions, optimize treatment plans, and improve patient care. Our study contributes to this progress by providing a robust and efficient solution for bone fracture detection using X-ray images. The high precision and recall achieved by our model underscore its reliability and its potential to significantly enhance fracture diagnosis.

In conclusion, our research showcases the remarkable impact of AI and deep learning in bone fracture detection. Through the design and implementation of a YOLO-based system, we have successfully demonstrated the effectiveness of our model in accurately detecting fractures in X-ray images. With a precision/recall of 91% compared to an accuracy 85%

90% in [60] [64] respectively our system holds great promise in improving fracture diagnosis and facilitating prompt and appropriate treatment decisions. The integration of AI in the field of medicine continues to shape the future of healthcare, and our work contributes to this transformative journey.

Perspectives

In this regard, our future works will focus more and more on:

- **Expand the dataset:** although the GAZ dataset provided a solid foundation for training and validation, expanding the dataset with a wider variety of X-ray images can further improve the robustness and generalization of the model. Including images from diverse sources, demographics, and varying degrees of fracture severity can help the model become more versatile.

- **Fine-tune the model:** fine-tuning the pre-trained YOLO model specifically for bone fracture detection may yield even better results. By training the model on a larger dataset with fine-grained annotations and carefully tuning hyperparameters, the model's performance can potentially be optimized further.

- **Explore multi-modal approaches:** investigate the integration of other imaging modalities, such as computed tomography (CT) scans or magnetic resonance imaging (MRI), alongside X-ray images. Combining multiple modalities can provide complementary information and improve the accuracy and reliability of fracture detection, especially in complex cases.

- **Address class imbalance:** class imbalance, where certain fracture types or conditions are underrepresented in the dataset, can impact the model's performance. Techniques such as data augmentation, oversampling, or utilizing generative models can help mitigate the effects of class imbalance and improve the model's ability to detect all fracture types accurately.

By addressing these aspects in future work, we can further advance the field of bone fractures detection and contribute to improving patient care, treatment planning, and clinical decision-making processes.

Bibliography

1. Court-Brown, C. M. & Caesar, B. Epidemiology of adult fractures: A review. en. *Injury* **37,** 691–697 (Aug. 2006).
2. Marolt, D., Knezevic, M. & Vunjak-Novakovic, G. Bone tissue engineering with human stem cells. *Stem cell research & therapy* **1,** 1–11 (2010).
3. Fazzalari, N. L. Bone fracture and bone fracture repair. *Osteoporosis International* **22** (2011).
4. Yang, S. *et al.* Diagnostic accuracy of deep learning in orthopaedic fractures: a systematic review and meta-analysis. en. *Clin. Radiol.* **75,** 713.e17–713.e28 (Sept. 2020).
5. Wang, Y. *et al.* Proteomic analysis of the peritrophic matrix from the midgut of third instar larvae, Musca domestica. en. *Biomed. Environ. Sci.* **29,** 56–65 (Jan. 2016).
6. Bagaria, R., Wadhwani, S. & Wadhwani, A. K. Bone fractures detection using support vector machine and error backpropagation neural network. *Optik* **247,** 168021 (2021).
7. Oyeranmi, A., Ronke, B., Mohammed, R. & Edwin, A. Detection of fracture bones in X-ray images categorization. *J Adv Math Comput Sci* **35,** 1–11 (2020).
8. Shen, S. C.-y., Fernández, M. P., Tozzi, G. & Buehler, M. J. Deep learning approach to assess damage mechanics of bone tissue. *Journal of the Mechanical Behavior of Biomedical Materials* **123,** 104761 (2021).
9. MD, G. S. *Musculoskeletal system* 2023. `https://www.kenhub.com/en/library/anatomy/the-musculoskeletal-system`.
10. Publishing, S. *Musculoskeletal, Paper Chart* `https://books.google.dz/books?id=e%5C_7mPQAACAAJ` (Scientific Pub Limited, 2003).

11. Jasvir S. Khurana Edward F. McCarthy, P. J. Z. *Essentials in Bone and Soft-Tissue Pathology* https://link.springer.com/book/10.1007/978-0-387-89845-2#book-header (Springer New York, NY, 2010).

12. De Buffrénil, V., de Ricqlès, A., Zylberberg, L. & Padian, K. *Vertebrate Skeletal Histology and Paleohistology* https://books.google.dz/books?id=tJcwEAAAQBAJ (CRC Press, 2021).

13. Scheuren, A., Wehrle, E. & Flohr, F. Bone mechanobiology in mice: toward single-cell in vivo mechanomics. *Biomechanics and Modeling in Mechanobiology* **16** (Dec. 2017).

14. University, J. H. *Anatomy of the Bone* 2022. https://www.hopkinsmedicine.org/health/wellness-and-prevention/anatomy-of-the-bone.

15. Martinez., C. *Some Bone Cold Facts* https://itjs.github.io/structure.html.

16. Anatomy, B. & Physiology. *Introduction to Bone* https://www.coursehero.com/study-guides/boundless-ap/introduction-to-bone/.

17. Miller, C. *Human Biology* 854–864. https://humanbiology.pressbooks.tru.ca/ (Thompson Rivers University, 2020).

18. Institute, T. O. *TYPES OF FRACTURES* https://www.orthopedic-institute.org/fracture-care/types-of-fractures/.

19. D'Andrea, D. M. M. *Common Fractures* https://mmdchiropractic.ca/common-fractures/.

20. Guly, H. Diagnostic errors in an accident and emergency department. *Emergency Medicine Journal* **18,** 263–269 (2001).

21. Mould, R. F. The early history of X-ray diagnosis with emphasis on the contributions of physics 1895-1915. *Physics in Medicine and Biology* **40,** 1741. https://dx.doi.org/10.1088/0031-9155/40/11/001 (Nov. 1995).

22. Ridolfi, R. *Study of the track reconstruction in the FOOT experiment for Hadrontherapy* PhD thesis (Mar. 2018).

23. Of Biomedical Imaging, N. I. & (NIBIB), B. *X-rays* https://www.nibib.nih.gov/science-education/science-topics/x-rays.

24. Tutorials, S. *Introduction to Biomedical Engineering - Week 6* Accessed: June 11, 2023. 2019. https://sbme-tutorials.github.io/2019/intro-to-BME/notes/6_week6.html.

25. Clinic, M. *X-rays* https://www.mayoclinic.org/tests-procedures/x-ray.

26. Mounts, J., Clingenpeel, J., McGuire, E., Byers, E. & Kireeva, Y. Most frequently missed fractures in the emergency department. *Clinical pediatrics* **50,** 183–186 (2011).

27. Pinto, A. *et al.* Traumatic fractures in adults: missed diagnosis on plain radiographs in the Emergency Department. en. *Acta Biomed.* **89,** 111–123 (Jan. 2018).

28. Shin, H.-C. *et al.* Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging* **35,** 1285–1298 (2016).

29. Awad, M. & Khanna, R. in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (Apress, Berkeley, CA, 2015). https://doi.org/10.1007/978-1-4302-5990-9_1.

30. Peng, J., Jury, E., Dönnes, P. & Ciurtin, C. Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges. *Frontiers in Pharmacology* **12** (Sept. 2021).

31. Hastie, T., Tibshirani, R. & Friedman, J. in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 9–41 (Springer New York, New York, NY, 2009). https://doi.org/10.1007/978-0-387-84858-7_2.

32. Ayodele, T. O. Types of machine learning algorithms. *New advances in machine learning* **3,** 19–48 (2010).

33. Gordon, A. D. *Classification* (CRC Press, 1999).

34. Puntanen, S. Regression analysis by example, by Samprit Chatterjee, Ali S. hadi. *International Statistical Review* **81,** 308–308 (2013).

35. Draper, N. R. & Smith, H. *Applied regression analysis* (John Wiley and Sons, 1998).

36. Dong, Y., Tao, D. & Li, X. Nonnegative multiresolution representation-based texture image classification. *ACM Transactions on Intelligent Systems and Technology (TIST)* **7,** 1–21 (2015).

37. Zhang, S., Li, X., Zong, M., Zhu, X. & Cheng, D. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)* **8,** 1–19 (2017).

38. Ng, K. L. S. & Mishra, S. K. De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. en. *Bioinformatics* **23,** 1321–1330 (June 2007).

39. Mahesh, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]* **9,** 381–386 (2020).

40. Lin, S.-W. Artificial Neural Network Related to Biological Neuron Network. *Advanced Studies in Medical Sciences* **5,** 55–62 (2017).

41. Dayan, P., Abbott, L. F., *et al.* Theoretical neuroscience: computational and mathematical modeling of neural systems. *Journal of Cognitive Neuroscience* **15,** 154–155 (2003).

42. Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. & Maglogiannis, I. *Artificial Neural Networks and Machine Learning – ICANN 2018 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III* (Jan. 2018).

43. Domingos, P. & Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning* **29,** 103–130 (1997).

44. SHARMA, S. *Activation Functions in Neural Networks* `https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6`.

45. Dubey, A. K. & Jain, V. *Comparative study of convolution neural network's relu and leaky-relu activation functions* in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018* (2019), 873–880.

46. Mehta, K. S. *How to Implement the Softmax Function in Python* `https://wandb.ai/krishamehta/softmax/reports/How-to-Implement-the-Softmax-Function-in-Python--VmlldzoxOTUwNTc`.

47. Donges, N. *What Is Transfer Learning? Exploring the Popular Deep Learning Approach* `https://builtin.com/data-science/transfer-learning`.

48. Alzubaidi, L. *et al.* Towards a Better Understanding of Transfer Learning for Medical Imaging: A Case Study. *Applied Sciences* **10,** 4523. ISSN: 2076-3417. `http://dx.doi.org/10.3390/app10134523` (June 2020).

49. Zhuang, F. *et al.* A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE* **109,** 43–76 (2021).

50. Albawi, S., Mohammed, T. A. & Al-Zawi, S. *Understanding of a convolutional neural network* in *2017 International Conference on Engineering and Technology (ICET)* (2017), 1–6.

51. Murphy, J. An overview of convolutional neural network architectures for deep learning. *Microway Inc,* 1–22 (2016).

52. Liu, S., Wang, H. & Li, R. Attention Module Magnetic Flux Leakage Linked Deep Residual Network for Pipeline In-Line Inspection. *Sensors* **22,** 2230. ISSN: 1424-8220. `http://dx.doi.org/10.3390/s22062230` (Mar. 2022).

53. O'Shea, K. & Nash, R. *An Introduction to Convolutional Neural Networks* 2015. arXiv: `1511.08458 [cs.NE]`.

54. Wu, L. & Perin, G. *On the Importance of Pooling Layer Tuning for Profiling Side-Channel Analysis* in *Applied Cryptography and Network Security Workshops* (eds Zhou, J. *et al.*) (Springer International Publishing, Cham, 2021), 114–132.

55. Zafar, A. *et al.* A Comparison of Pooling Methods for Convolutional Neural Networks. *Applied Sciences* **12,** 8643. ISSN: 2076-3417. http://dx.doi.org/10.3390/app12178643 (Aug. 2022).

56. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

57. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. *Densely connected convolutional networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 4700–4708.

58. Shafiq, M. & Gu, Z. Deep Residual Learning for Image Recognition: A Survey. *Applied Sciences* **12.** ISSN: 2076-3417. https://www.mdpi.com/2076-3417/12/18/8972 (2022).

59. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence* **38,** 142–158 (2015).

60. Rajpurkar, P. *et al.* Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint arXiv:1712.06957* (2017).

61. Chung, S. W. *et al.* Automated detection and classification of the proximal humerus fracture by using deep learning algorithm. *Acta Orthopaedica* **89,** 468–473. https://actaorthop.org/actao/article/view/7171 (July 2018).

62. Endo, T. U. . Y. T. . S. G. . H. M. . K. W. . N. ng intertrochanteric hip fractures with orthopedist-level accuracy using a deep convolutional neural network. *Skeletal Radiology* **48** (2019).

63. Ju, R.-Y. & Cai, W. *Fracture Detection in Pediatric Wrist Trauma X-ray Images Using YOLOv8 Algorithm* 2023. arXiv: 2304.05071 [cs.CV].

64. Gan, K. *et al.* Artificial intelligence detection of distal radius fractures: a comparison between the convolutional neural network and professional assessments. *Acta Orthopaedica* **90,** 394–400. https://actaorthop.org/actao/article/view/591 (Apr. 2019).

65. Balaji, G., Subashini, T., Madhavi, P., Bhavani, C. & Manikandarajan, A. *Computer-aided detection and diagnosis of diaphyseal femur fracture* in *Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 1* (2020), 549–559.

66. Sha, G., Wu, J. & Yu, B. *Detection of Spinal Fracture Lesions Based on Improved Yolo-tiny* in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)* (2020), 298–301.

67. Abbas, W. *et al. Lower leg bone fracture detection and classification using faster RCNN for X-rays images* in *2020 IEEE 23rd International Multitopic Conference (INMIC)* (2020), 1–6.

68. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

69. Redmon, J. & Farhadi, A. *YOLOv3: An Incremental Improvement* 2018. arXiv: 1804.02767 [cs.CV].

70. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. *You Only Look Once: Unified, Real-Time Object Detection* 2016. arXiv: 1506.02640 [cs.CV].

71. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* **60,** 84–90 (2017).

72. Eschweiler, J. *et al.* Anatomy, Biomechanics, and Loads of the Wrist Joint. *Life* **12.** ISSN: 2075-1729. https://www.mdpi.com/2075-1729/12/2/188 (2022).

73. Vos, F. *et al. A Statistical Shape Model without Using Landmarks.* in. **3** (Jan. 2004), 714–717.

74. Hooper, G. The anatomy and pathophysiology of the wrist. *Ortopedia, traumatologia, rehabilitacja* **8,** 139–143 (2006).

75. Amarasooriya, M., Jerome, T. J. & Tourret, L. Current Concepts in Scapholunate Instability Without Arthritic Changes. *Indian journal of orthopaedics* **57,** 515–526. https://doi.org/10.1007/s43465-023-00839-0 (2023).

76. Jones, W. & Ghorbal, M. Fractures of the trapezium a report on three cases. *Journal of Hand Surgery* **10,** 227–230 (1985).

77. De Beer, J. D. V. & Hudson, D. Fractures of the triquetrum. *The Journal of Hand Surgery: British & European Volume* **12,** 52–53 (1987).

78. Tschauner, E. N. M. J. F. H. E. S. . S. A pediatric wrist trauma X-ray dataset (GRAZPEDWRI-DX) for machine learning. *Scientific Data.* **9,** 155–168 (2022).

79. Cecava, N. D. M., Kephart, D. A. J. M. & Bui-Mansfield, L. T. M. Bone Lesions of the Hand and Wrist: Systematic Approach to Imaging Evaluation. *Contemporary Diagnostic Radiology* **43,** 1–7 (Feb. 2020).

80. Bronner, F. in *Principles of Bone Biology (Third Edition)* (eds Bilezikian, J. P., Raisz, L. G. & Martin, T. J.) Third Edition, 515–531 (Academic Press, San Diego, 2008). https://www.sciencedirect.com/science/article/pii/B9780123738844000446.

81. Rana, R. S., Wu, J. S. & Eisenberg, R. L. Periosteal Reaction. *American Journal of Roentgenology* **193.** PMID: 19770293, W259–W272. eprint: https://doi.org/10.2214/AJR.09.3300. https://doi.org/10.2214/AJR.09.3300 (2009).

82. Knipe, H., Bell, D., Hacking, C. & et al. *Pronator quadratus sign* Reference article, Radiopaedia.org. Accessed on 17 May 2023. 2023. https://doi.org/10.53347/rID-26112.

83. Awan, R., Ghazanfar, H., Pena, K. & Farkhad, R. An Unusual Case of Acute Chondrocalcinosis in Wrist Joint Presenting as Cellulitis. *Cureus* **9** (Dec. 2017).

84. Bagade, S. S. & Shandilya, V. K. Use of histogram equalization in image processing for image enhancement. *International Journal of Software Engineering Research & Practices* **1,** 6–10 (2011).

85. Vallat, R. Pingouin: statistics in Python. *J. Open Source Softw.* **3,** 1026 (2018).

86. Python, W. Python. *Python Releases Wind* **24** (2021).

87. Gunawan, T. S. *et al.* Development of video-based emotion recognition using deep learning with Google Colab. *TELKOMNIKA (Telecommunication Computing Electronics and Control)* **18,** 2463–2471 (2020).

88. Imambi, S., Prakash, K. B. & Kanagachidambaresan, G. R. in *Programming with TensorFlow: Solution for Edge Computing Applications* (eds Prakash, K. B. & Kanagachidambaresan, G. R.) 87–104 (Springer International Publishing, Cham, 2021). https://doi.org/10.1007/978-3-030-57077-4_10.

89. Faes, D. M. Use of Python programming language in astronomy and science. *arXiv preprint arXiv:1807.04806* (2018).

90. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362 (2020).

91. Vyshnavi, V. R. & Malik, A. Efficient Way of Web Development Using Python and Flask. *Int. J. Recent Res. Asp* **6,** 16–19 (2019).

92. Bradski, G. The openCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* **25,** 120–123 (2000).

93. Makino, M., Odaka, T., Kuroiwa, J., Suwa, I. & Shirai, H. Feature Selection to Win the Point of ATP Tennis Players Using Rally Information. *International Journal of Computer Science in Sport* **19,** 37–50 (July 2020).

94. Van der Walt, S. *et al.* scikit-image: image processing in Python. *PeerJ* **2,** e453 (2014).

95. Rajpurkar, P. *et al. MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs* 2018. arXiv: 1712.06957 [physics.med-ph].

96. For Artificial Intelligence in Medicine, S. C. & Imaging. *LERA: Lower Extremity Radiographs* (2023). https://aimi.stanford.edu/lera-lower-extremity-radiographs.