COMPUTER-AIDED DIAGNOSIS SYSTEM FOR BONE FRACTURE DETECTION USING MACHINE LEARNING ALGORITHMS

AMIRKHASHAYAR NADERIAN

A thesis submitted to Nottingham Trent University in candidature of the Degree of Doctor of Philosophy

School of Science and Technology
Department of Computer Science
Nottingham Trent University

November 2022

Amirkhashayar Naderian: *Computer-aided diagnosis system for bone fracture detection using machine learning algorithms,* A thesis submitted to Nottingham Trent University in candidature of the Degree of Doctor of Philosophy, © November 2022

SUPERVISORS:
Prof. Saeid Sanei
Prof. Eiman Kanjo

DECLARATION

I hereby declare that the this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Nottingham, UK, November 2022

Amirkhashayar Naderian

ABSTRACT

Diagnostic imaging technology has revolutionized the healthcare industry by allowing more accurate and earlier diagnosis of diseases. This technology reduces the need for invasive procedures such as surgery and enhances the quality of patient care. Several machine learning algorithms like SVM, K-mean clustering and UNET have been demonstrated to be capable of solving classification, detection, and segmentation problems in medical imaging, as well as being used for super-resolution techniques. The purpose of this thesis is to examine machine learning and image processing methods for four key challenges in medical image analysis.

The first one is the segmentation of medical images. The second challenge involves implementing super-resolution techniques for medical images. Third, using image processing methods in order to diagnose the abnormalities. The fourth contribution is to enrich the image information by mapping of medical images between different modalities using deep neural models. In this research, all contributions aim at developing an end-to-end model that can detect fractures automatically or be used as a clinical assistant to reduce errors. As the first contribution, the thesis presents a multi-stage novel approach for bone segmentation in X-ray images using faster region-based convolutional neural network (R-CNN) and distance regularized level set evolution (DRLSE) algorithms. A hybrid model utilizing deep neural network (DNN) and image processing techniques are proposed to segment the bones in two stages. Our model is more robust to the changes in X-ray images, as well as applicable to bones that are misplaced. Additionally, we have used transfer learning to reduce the amount of time and effort required to collect and label the data. As the second contribution, DNN models are used to enhance the resolution of medical images. CNN and generative adversarial network have been used as super-resolution techniques to achieve high-resolution medical images. The analysis includes subjective and objective evaluations of different models on regions with or without fractures to compare them with our model. The third contribution involves applying different image analysis methods to X-ray images in order to detect fractures with the minimum amount of human intervention. By using entropy and intensity, we have also attempted to identify regions of interest that have a higher probability of having fractures. We also evaluate the effect of super-resolution technique on the saliency map with and without fractures. Lastly, we present image-to-image mapping by using variational autoencoders and generative adversarial networks to reduce the cost of diagnosis and medical images retrievals. We have attempted to map X-ray images to MRIs in this section in order to fuse the high diagnostic information existing in MRIs, for enhancing the matched X-ray images.

PUBLICATIONS

1. Zhao, Min, Amirkhashayar Naderian, and Saeid Sanei. "Generative Adversarial Networks for Medical Image Super-resolution." 2021 International Conference on e-Health and Bioengineering (EHB). IEEE, 2021.

2. Amirkhashayar Naderian, and Saeid Sanei. "Automated Bone Segmentation Using a Hybrid Model of DRLSE and Faster R-CNN" Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA) . IEEE, 2022.

ACKNOWLEDGEMENTS

It is my pleasure to acknowledge and extend my sincere gratitude to all those who supported me during my PhD research.

Ashkan you did nothing for my PhD but I mentioned your name to be remembered.

There may be times when you can do less for those you love the most. Your unconditional love and support will always be remembered by me. Rest in peace Mamanbozorg. It was you who gave me my first book, and I promise to never stop learning.

Lastly, I would like to express my gratitude to my parents, who have supported me generously and, more importantly, paid a tremendous amount of money for me to become a doctor.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A and E	accident and emergency
AE	autoencoder
AI	artificial intelligence
ANN	artificial neural network
AVN	avascular necrosis
BN	batch normalization
BP	backpropagation
CAD	computer aided diagnosis
CGAN	conditional generative adversarial network
CNN	convolutional neural network
CT	computer tomography
D	domain
DL	deep learning
DRLSE	distance regularized level set evolution
ED	emergency department
Faster R-CNN	faster regional convolutional neural network
FC	fully connected
GAN	generative adversarial network
HR	high resolution
IoU	intersection-over-union
LOG	laplacian gausian filter
LR	low resolution
LSF	level set function
MLP	multi-layer perceptron

MOS	mean opinion score
MRI	magnetic resonance image
NGTDM	neighborhood grey-tone difference matrices
NN	neural network
PA	postero-anterior
PET	positron emission tomography
PReLU	parametric rectified linear unit
PSNR	peak signal-to-noise ratio
R-CNN	region-based convolutional neural network
ReLU	rectified linear unit
ResNet	residual network
RF	radio frequancy
RF	random forest
RNN	recurrent neural network
ROI	region of interest
RPN	region proposal network
SGLD	spatial gray level dependence
SIFT	scale invariant feature transform
SISR	single image super resolution
SPADE	spatially-adaptive denormalization
SR	super resolution
SRCNN	super resolution convolutional neural networks
SRGAN	super resolution generative adversarial network
SSIM	structural similarity index measure
SVM	support vector machines
T	task

TA	texture analysis
VAE	variational autoencoder

1 INTRODUCTION

Over two-year period (2002-2004), 5879 patients visited the Emergency Department (ED) with injuries. Approximately 1% of all visits to the ED result in an incorrect fracture diagnosis, and 3.1% of fractures are not diagnosed at the time of the first visit [1]. The majority of early diagnosis is based on X-ray images [2]. It has been reported that missed fractures are the most common type of interpretational error made by physicians on musculoskeletal radiographs [3]. This may lead to treatment delays, resulting in malunions and arthritis with associated morbidity, and is one of the most common causes of malpractice claims against physicians [4, 5].

Artificial intelligence (AI) refers to the use of computers to model intelligent behaviour with minimal human involvement. Recent advances in artificial intelligence, particularly deep learning, have enabled machines to better interpret and represent complex data. The term deep learning refers to a subset of AI that is composed of layers of artificial neurons. There has been a significant increase in the use of deep learning in recent years. A number of studies have been conducted in the field of orthopaedics and traumatology that have used deep learning to detect fractures on radiographs. Studies using deep learning to detect and classify fractures from computed tomography (CT) and specifically plain X-ray images are limited.

In order to diagnose bone fractures, clinicians often use plain 2D X-ray images. Although it is effective for many types of obvious fractures, there is a limitation in the use of X-rays for microfractures or some specific fractures like those found in the Scaphoid bone or Calcaneal bone due to the resolution, bone overlaps and also the position of the fractures. Therefore, it is necessary to use MRI or high resolution images. In comparison to X-rays, MRIs are more expensive and less accessible. The development of a neural network (NN) model capable of reconstructing a single MRI from one or many low resolution X-ray images that contains crucial details related to fracture that are less visible in X-ray images may help reduce errors and make diagnosis more affordable. Reconstructed MRI images do not have to be exactly like MRI images and do not have to contain all the slices of MRIs or even all the details of MRIs, but they can contain some details that are related to fractures, which were not visible on X-rays.

1.1 MEDICAL IMAGING OVERVIEW

Medical imaging is a set of techniques that allows visualisation of the human anatomy using a non-invasive method. Through the use of these techniques, we are able to gain a better understanding of the internal structure of an organ, as well as to make accurate clinical diagnoses or treatments when a disease is present. Medical imaging was invented at the end of 19th century [6], and it has helped public health since then. Most medical imaging techniques are non-invasive, meaning there is no need to cut the skin, have open surgery, or even insert anything into the patient's body. However, some involve hazardous radiation such as X-ray and Positron emission tomography (PET).

Over the past century, a great deal of progress has been made in X-ray imaging and today, images are more detailed and contain a greater level of information that can be analysed by clinicians or artificial NNs. Since the advent of digital medical imaging, image analysis by computer has become increasingly popular, and it has become increasingly important to make an accurate diagnosis. The use of machine learning methods can be beneficial not only for clinicians, but also for reducing the demand for expensive medical images, such as MRI. The purpose of the image analysis is primarily to detect patterns between a limited number of images in order to assist clinicians in making more accurate decisions during diagnosis, surgery, or treatment.

Computers can serve as clinical assistants, increase image resolution, or even perform diagnoses on their own. Our research aims to improve segmentation of regions of interest (ROI) for clinicians, increase resolution by super-resolution techniques, make bone fracture diagnoses, and reconstruct MRI images from X-rays using machine learning and image processing methods.
In this thesis, X-ray images and MRIs are the most commonly used modalities of medical images for bone fracture detection.

1.1.1 *X-Radiography*

An X-ray is a form of electromagnetic radiation that has enough energy to dislodge the electrons from the atoms when it interacts with them. Medical imaging utilises X-rays with energies ranging from 10 keV to 150 keV depending on the region of the body being imaged. The following equation can be used to calculate the energy of X-rays:

$$E = h\nu \tag{1.1}$$

In this case, ν is frequency of radiation and h is Planck's constant $h = 4.136 \times 10^{-15}\,\text{eV/Hz}$ [7].

Radiography uses X-ray beams to visualise the structure of internal organs [8]. In radiography, X-ray beams are projected onto the body, and depending on the density and structure of the region, various tissues, and organs, such as bones, vessels and skin, absorb different amounts of energy. On the other side of the body, an analogue or digital sensor array records the amount of energy absorbed, which gives us a 2D picture of the structure in that area.

Radiography can be divided into two types: fluoroscopy and projected X-ray. Fluoroscopy uses contrast agents like Barium to visualise vessels or moving body structure, while X-rays (Figure 1.1) are commonly used to visualise bones. Even though radiography is the oldest method of obtaining medical images, it is still the most prevalent method due to the low cost of taking radiographs as well as the wide availability of these images. Because the purpose of this thesis is to diagnose bone fractures, only projected X-rays are used rather than fluoroscopy.

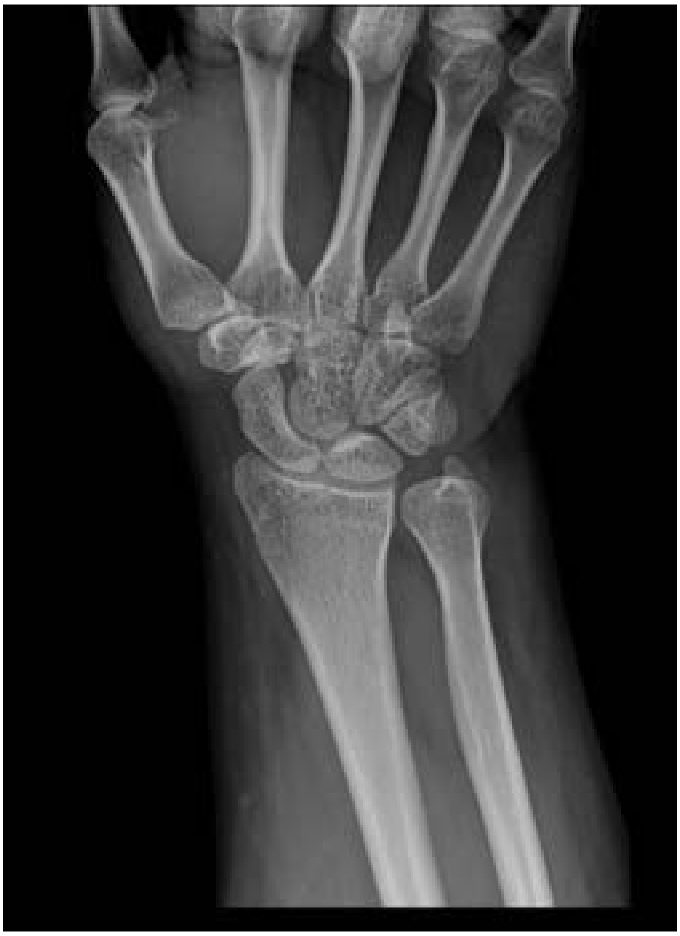


Figure 1.1: An example of X-ray image of the right wrist.

1.1.2 *MRI*

In spite of the fact that there are many different medical imaging modalities available for identifying the structure of organs, MRI (Figure 1.2) is able to provide the highest level of contrast in soft tissues. MRI has many advantages, but it also requires a considerable amount of post-processing and scan time and it is more expensive than other imaging techniques. Even though MRI plays an important role in clinical diagnosis, it is susceptible to motion caused by body movement or cardiorespiratory system.

In MRI, radio frequancy (RF) waves and magnetic fields are used to create cross-sectional images of the organs within a patient's body [9]. As opposed to radiography, there are no X-ray beams involved, so this procedure is much safer.

An MRI uses a strong magnetic field to align the core spin axis of hydrogen atoms [10]. RF waves are then used to perturb these atoms by using RF waves. As soon as the RF propagation stops, hydrogen atoms emit RF signals to return to their stable form. An MRI scanner captures these emitted RF signals in order to reconstruct an image through the use of Fourier transformation.

Several types of tissues can be distinguished by their relaxation times, T1, which is the amount of time it takes for excited atoms to realign with external magnetic forces again, and T2, which is the time it takes for perturbed atoms to lose phase coherence. Other than T1-weighted and T2-weighted MRIs, successive pulse sequences (TR) and echo times (TE) may also be used to generate images [11].

Depending on the amount of water in the tissue, each tissue has a different T1 or T2 time, which can be used to reconstruct the images, including organ structures.

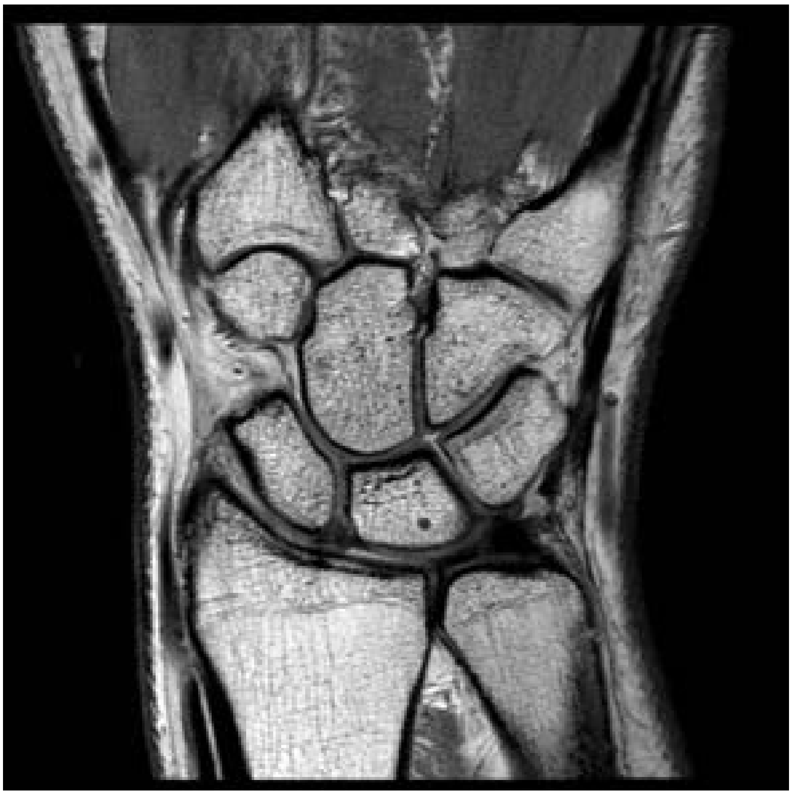


Figure 1.2: An example of MRI T1 image taken from the wrist.

1.2 OBJECTIVES AND CHALLENGES

Our challenges can be categorised into three groups related to “quality of medical images”, “labelling of medical images” and how much our models can be useful in “clinical practice”. Based on how we approach these challenges, we have categorised them accordingly.

1.2.1 *The Quality of Medical Images*

There are three major challenges in the image analysis due to the nature of images. First of all, the quality of some acquired images may be poor. Due to the safety of the patient, the amount of radiation should be reduced, resulting in a low peak signal-to-noise ratio (PSNR). Additionally, clinical images are often degraded by artefacts, which may occur as a result of patient movement or operator error. Medical images with low PSNR and artefacts adversely affect the accuracy of machine learning models for image analysis.

Secondly, there is a great deal of diversity in medical images. Being available in many different modes, medical images can also be captured on many different types of organs. For instance, bone segmentation requires a different approach for each part of the body, such as the chest or wrist. It may also requires a different approach when using MRI or X-rays. The images are also subject to variation in intensity, noise, and the quality of imaging systems. As a result of this diversity, it would be impossible to develop one universal machine learning models for all diagnostic problems.

In medical images, the human anatomical structure varies in position size, and shape and may appear very complex. There are some anatomies that look like a group of pixels such as fat mass [12], or like a group of small patches, such as brain haemorrhages [13].

Third, since there is no restriction on the position of the organ during the acquisition of the image, each image has a specific angle of view for the same organ, so two images taken from the same organ of one person may look different and this makes detecting anomalous patterns in medical images difficult. Often mis-diagnosis can happen due to the angle of imaging system with respect to the object. This is even more problematic when 2-D images are to be taken.

1.2.2 *Labelling*

There are three levels of labelling that can be applied to medical images: pixel-by-pixel, region-by-region, and image-by-image. For a better understanding, it is useful to consider some examples. For example, if you have 100 X-rays of the wrist, and you ask a specialist only to label the images with or without fracture, it is known as image labelling. Region labelling is the process of having the specialist look through the images and draw a rectangular area around the fracture as the region of interest (ROI) that contains the fracture. As part of this method of labelling, the specialist must not only classify images into two groups of those with fractures and those without fractures, but also determine the coordinates of the four corners of the rectangular ROI. Although the number of corners can be different in a non-rectangular ROI, instead we have used rectangular ROI.

For pixel-to-pixel or pixel-by-pixel labelling, the specialist must classify each pixel according to whether it represents the region of a fracture or a healthy part of the bone.

When we have a dataset consisting of 100 images, we only require a vector of labels with a length of 100 in order to perform image-by-image labelling. An index value of 1 indicates that the image contains fractures, while a value of 0 indicates that the image is healthy.

To perform region labelling, a matrix of size 5×100 is required. The length of 100 represents the number of images we have, the width of 5 represents the four coordinates of ROI and the number of classes of images based on whether the images contain fractures or not. Likewise, for pixel-wise labelling, if the size of our images is 500×500, we will need a matrix with $500 \times 500 \times 100$ size. The cost of human resources and time increases as you progress from image labelling to pixel-by-pixel labelling.

There are two challenges associated with image labelling. It should be noted that only a small portion of images can be labelled, which means that our labelled dataset is limited. Furthermore, pixel-wise labelling cannot be performed accurately. Image labelling and ROI labelling are both susceptible to error, however, pixel-wise labelling results in a greater degree of error.

1.2.3 *Clinical Practice*

It can be challenging to diagnose some fractures due to their varied shapes and sizes or obscured by another dense tissue. At a clinical level, there are four issues that may arise. First, in some places, such as small clinics or towns, there are no devices capable of taking the required modes of images. For example, X-ray images can be captured in some clinics, but MRI or CT imaging devices are not available. Even if MRI devices are available, taking an MRI is more expensive and time consuming, so it would not be easy to obtain an MRI for all patients. Secondly, the lack of resources and the insufficient number of clinicians, nurses, or specialists result in an increase in human error due to the exhaustion of resources. As the third issue, some diseases or fractures cannot be diagnosed immediately. To obtain an accurate diagnosis, the patient requires follow-up from a couple of days to a couple of months. For example, calcaneal fractures and scaphoid fractures require follow-up for approximately eight weeks [14]. For the fourth issue, medical images portray a large view of an organ. For example, a specialist making a diagnosis related to a spinal cord must examine 120 images, which causes fatigue and increasing errors.

1.3 THESIS CONTRIBUTION

To address the challenges above, we propose novel solutions based on machine learning techniques:

Automated Bone Segmentation Using a Hybrid Model of Distance Regularised Level Set Evolution (DRLSE) and Transfer Learning: Most of the current automated bone segmentation methods are based on CT images or MRIs [15, 16]. Since X-ray images may have varied types of noises or levels of sharpness based on the apparatus or operator of the system, segmenting them can be a challenging task. Some research has been done on X-rays [17], but most of them require huge computation costs or require very large datasets. We have developed a model that automatically segments bones without the need for human intervention. A two-stage model is used. In the first stage, Faster region-based convolutional neural network (R-CNN) is used to detect the region around the phalange, and in the second stage, image processing methods such as DRLSE are used to segment the bone perfectly. Our fully automated framework will not only assist clinicians in looking at smaller areas by proposing ROI, but also can used alongside other frameworks for further analysis and fracture diagnosis. Thus, this model has a great deal of potential for use as a computer aided diagnosis (CAD) system. To address the issue of labelling, we have applied transfer learning to reduce the requirement for large datasets. Additionally, using a hybrid model allows us to simplify the process of labelling, since we only require ROI labelling, which can be performed by a non-specialist. The model is also robust to all changes in image angles, artefacts, translations, and deformations. It is also noteworthy that this framework is robust enough to work in those tricky situations where the shape of a bone is completely changed due to big fractures.

Generative Adversarial Networks and Convolutional Neural Network for Medical Image Super-resolution: In Chapter 4 we propose the use of generative adversarial networks (GAN) and CNN in order to improve the resolution of X-ray images. Certainly, the enhancement of X-ray images can improve diagnosis. The quality of the results is determined by the PSNR and mean opinion score (MOS) when applied to plain hand X-rays. GANs have been demonstrated to be superior in these applications, particularly in terms of MOS. In addition, we have evaluated the impact of this super-resolution system on the framework presented in the "Automated Bone Segmentation Using a Hybrid Model of DRLSE and Transfer Learning" chapter. The aim of this study is to address the problem of having a large number of low-resolution medical images in clinical trials as well as the possibility of using this model with other methods to increase bone segmentation accuracy. Recently, variety of GAN and CNN architectures have been used for super-resolution medical images. Using our design, our main objective was to compare different architectures on our dataset and evaluate the effect of SRCNN and SRGAN on our automated bone segmentation algorithm. Recently, variety of GAN and CNN architectures have been used for super-resolution medical images [18–20]. The majority of these methods have been implemented on MRIs. We aimed to design an architecture that could be applied to X-ray images. As part of our analysis, we have compared different architectures with different depth and width in order to find the most suitable design, and we have also evaluated the effect of SRCNN and SRGAN on our automated bone segmentation algorithm.

Using Saliency Map for Fracture Diagnosis and Effect of Super-Resolution techniques on the Points Correspond to Fracture: Our objective in Chapter 5 is to determine if it is possible to detect bone fractures automatically using saliency maps of entropy and intensity of X-ray images. We have used these two features because regions correlated to fractures have more entropy than other parts of the bones, however picking regions with high entropy can also select regions correlated to the edges of the bones. Even though edges of bones can have high entropy like fractures, they have a much lower intensity level than fracture regions since half of the regions around the edges lie outside the bones. Thus, by using these two features, we can identify regions where fractures are likely to exist. This model can be used as an independently to identify the region of fracture in X-ray images and assist clinicians system to reduce human error. An error in fracture diagnosis may lead to necrosis or other complications, which will require additional resources. Therefore, it is expected that by following this method such problems are alleviated. Using a saliency map involves the use of image processing methods and does not require any configuration or training. Additionally, this system may be able to detect microfractures that clinicians are unable to detect using X-ray images. Our main objective was to provide clinicians with a region of interest in order to reduce cilinal error in fracture detection, so we did not compare our model with human error.

X-ray to MRI Using The Hybrid Model of Variational Autoencoder and Generative Adversarial Network: While X-ray images can be much cheaper and easier to capture than MRIs, they can also be much noisier, and sometimes X-ray images are not enough to make a diagnosis, so we may need an MRI instead. As part of Chapter 6, we designed a state-of-the-art architecture for translating X-ray images into MRI images. So this algorithm can help us capture X-ray images easily and use the benefits of having MRIs in diagnosis. For the dataset to train our deep model, although we do not need labelling, it requires paired X-ray image and MRI datasets from the same scene. So, we required matched X-ray images and MRIs taken from the same region within a time-frame of less than two weeks to make sure that fractured bone has not been changed dramatically. Thus, by reconstructing an MRI image from an X-ray image, this framework can help detect fractures that cannot be detected by X-rays. As a result, in the event that no MRI device is available or in order to save resources and time, only X-ray images are captured, and our system can transfer the MRI information into X-ray images.

1.4 THESIS LAYOUT

The thesis is organised as follows: Chapter 2 introduces the concept of classical machine learning methods as well as new ones that are relevant to the remaining chapters. Chapter 3 presents an automated bone segmentation method based on a hybrid model of DRLSE and Transfer Learning. In Chapter 4, GANs and CNNs are evaluated for enhancing the resolution of medical images. In Chapter 5, we propose image processing-based models that may be utilised to diagnose fractures using saliency maps. Additionally, we discuss the effect of super-resolution techniques on points in saliency map associated with fractures. In Chapter 6, a framework based on VAE and GAN is proposed for translating X-ray images into MRIs. Each chapter begins with an introduction that discusses the relevant research, followed by an explanation of the basic concept of our model. We then present a methodology section in which we discussed how our model is modified to have more accurate results. We also discuss our novel architecture, loss function, dataset, and mathematical equation in methodology. Our model has been compared with other existing models, and the results evaluated in the Experiments Section. There is also a short conclusion at the end of every chapter. The thesis is summarised and concluded in Chapter 7.

2 BACKGROUND

The development of Artificial Intelligence (AI) has transformed the field of medical imaging, leading to a new era in diagnostic accuracy and patient care. The purpose of this chapter is to provide an overview of the historical progression for medical imaging. Machine learning algorithms, image processing methods, and artificial intelligence play an important role in the advancement of diagnostic procedures. While traditional imaging techniques are effective, they were often limited in terms of resolution, speed, and the need for manual interpretation, and machine learning algorithms and specifically AI are increasingly overcoming those challenges. The integration of advanced machine learning algorithms and deep learning models into medical imaging has not only improved the diagnostic process but has also opened up opportunities to diagnose diseases in early stages and personalized medicine. The ability of AI to analyze complex patterns in imaging data surpasses the accuracy of humans in certain aspects, reducing diagnostic errors and expanding the range of diseases that can be detected. This chapter provides a comprehensive background on AI in medical imaging, including improving image quality and identifying abnormalities automatically. As many sophisticated algorithms rely on classic neural network structures, we will also provide a simple overview of these foundational models.

2.1 IMAGE PROCESSING

A digital image is captured by small sensors that are capable of detecting and measuring light intensity. As a result of this measurement, three channels of red, blue, and green are converted to intensity in one pixel. Images consist of thousands of pixels adjacent to one another that represent the observed object [21].

Digital images can contain a great deal of information about the structure and color of the scene they capture. To make sense of the information recorded in digital images, we need to analyse them, which is referred to as image processing. Image processing can be used to extract features related to specific objects, tissues, or diseases. Whenever a feature can be found that relates to a disease, it can assist clinicians in accurate diagnosis. Further, image processing is also capable of finding some relationships between patterns in images that can be used to predict a disease [22]. In brief, image processing involves methods, algorithms, and mathematical models that are used to understand, summarise, and visualise the information that is extracted from digital images. Additionally, it can be used to compress images, reduce noise and artefacts, and increase image resolution.

Formally, a digital image can be described as matrix of pixels, each a function $f(x, y)$, where x and y are spatial coordinates and $f(x, y)$ contains spatial information about the region. This chapter summarises the necessary information about the concept of image processing as well as a couple of methods related to image classification and feature recognition [23].

2.1.1 *Image Segmentation*

Image segmentation refers to the process of separating desired regions or objects from an image. "Object Segmentation" and "Region Segmentation" are two main categories to consider:

Object Segmentation: This involves classification of each pixel in an image in order to identify which pixels correspond to which objects. As part of object segmentation, all pixels are classified to determine whether they are background pixels or pixels related to the object (Figure 2.1).

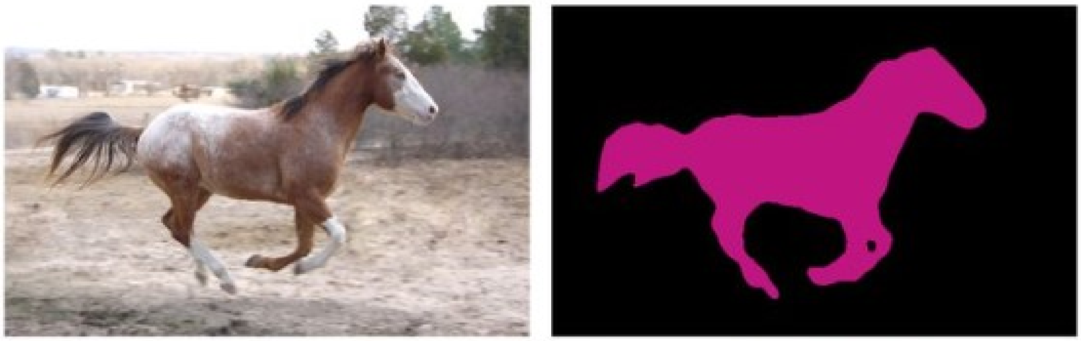


Figure 2.1: As an object segmentation task, the horse segmented from the image [24].

The classification of pixels may be based on the intensity of each pixel or its relationship to its neighbouring pixels. By using spatial information, such as intensity or entropy, we are able to extract information about the texture of the object to segment. Sometimes, the shape of the object is used in combination with texture for the purpose of segmentation.

It is common to use classical image processing methods for bone segmentation, such as thresholding, edge detection, watersheds [25], level sets / graphs / cuts [26, 27], deformable models [28–30], self-organising maps [31, 32] and others [33, 34]. Some machine learning methods have also become popular for bone segmentation in recent years, such as Random Forest (RF) or convolutional neural network architectures such as U-net [35, 36].

It is also possible to detect objects using contours. The Canny edge detection, Sobel or any other contour detection algorithm [37] could detect the contour and border. The closed or nearly closed contour is considered as the object. Moreover, thresholds can also be used for binarising pixels of an image to determine whether they represent objects or backgrounds. As an example, if the intensity of a pixel in an image is higher than a specific value, it would be considered to correspond to the object.

Region Segmentation: Region segmentation is the process of segmenting an area larger than an object as a region of interest (ROI) (Figure 2.2). Rather than focusing on shape, region segmentation primarily focuses on the texture. In AI medical assistant systems, ROI has great importance, as it allows clinicians to double-check the ROI in which there is a higher probability of abnormalities than in the rest of the image.

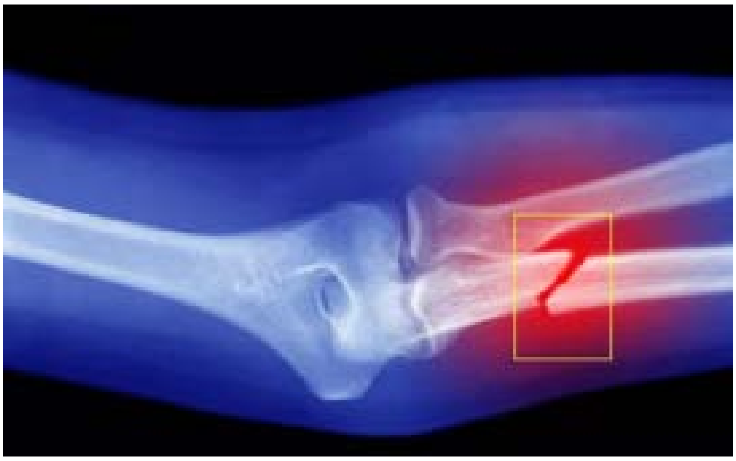


Figure 2.2: The region including bone fracture is segmented as ROI.

2.1.2 *Image Registration*

X-rays, CTs, positron emission tomography (PET), MRI, or other imaging modalities may be used depending on the nature of tissue and the clinical application. An image registration process involves aligning multiple images using spatial information in order to produce an integrated image. The use of this method allows us to avoid issues such as image rotation, scaling, or transformation, that are common when overlaying images [38]. Additionally, it can be used to produce larger panoramic images or images with a higher resolution [39]. For medical imaging, image registration is often used to align images from different cameras in order to provide a more accurate visual representation of an organ. An image registration technique can also be used to align or fuse images between two different imaging modalities, such as MRI and X-ray, to assist clinicians in making an accurate diagnosis [40]. The MRI and CT images can be registered together in order to have more diagnostic information and consequently better segmentation (Figure 2.3).

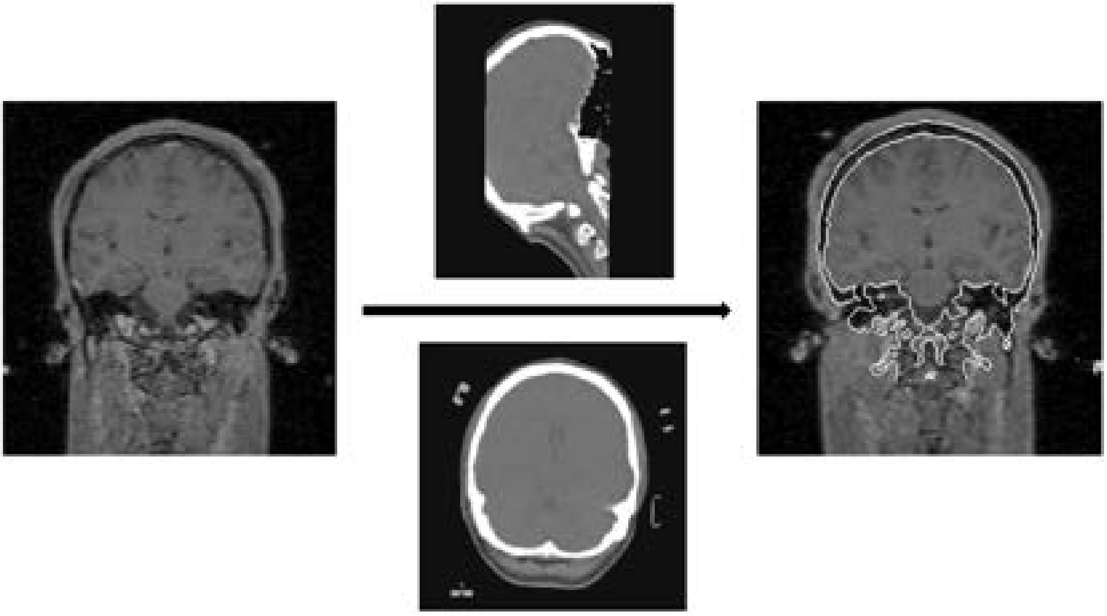


Figure 2.3: Image left represents the original MR image of the brain, while image right represents the registered MR image that skull bone has been segmented using CT images. [41].

2.2 MACHINE LEARNING

2.2.1 *Classical Machine Learning Methods*

As a result of advances in computer technology, implementing machine learning algorithms with higher computational costs have become more feasible. The algorithm is referred to as "supervised learning" or "unsupervised learning" depending on whether the training dataset has labels for the data. When supervised learning takes place, the algorithm learns from the input set of training data and its ground-truth labels. A classification algorithm or a regression algorithm produces an output by minimising the distance between the output and a labeled target. In computing, this distance is referred to as a loss function.

As opposed to supervised learning, unsupervised learning does not rely on any label. The objective of unsupervised learning (clustering) is to divide the data into distinct clusters by minimising intra-cluster members and maximising inter-cluster ones. The purpose of this algorithm is to explore some patterns for clustering unlabelled data. Semi-supervised learning also uses both labelled and unlabelled datasets.

Currently, deep learning algorithms are the most effective methods for performing computer vision tasks when sufficient training data are available. The purpose of this section is to provide an overview of some classical classifiers and new deep learning feature recognition models. Our neural network architecture and classification methods in this thesis are based on the support vector machine algorithm and the classical artificial neural network. It is therefore important to have a better understanding of these algorithms.

2.2.1.1 *Support Vector Machines*

Support vector machines (SVM) [42] is a supervised learning algorithm used to solve regression and classification problems. This algorithm separates points from different classes by utilising one or more hyperplanes drawn in n-dimensional space and optimally set to have the maximum distances with the existing cluster members (Figure 2.4).

In the case of linearly separable data, one linear hyperplane is sufficient for separating two classes of data. Mathematically, given training data $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_n, y_n)\}$ which $y_i \in \{-1, 1\}$, SVM is learning a linear classifying function $f(x) = w^T x + b$ that $f(x_i) > 1$ if $y_i = 1$ and $f(x_i) < -1$ if $y_i = -1$ where b is the bias (distance from hyperplane to the origin). As a result, the SVM equation is as follows:

$$\min_{w} \|w\|^2, \quad \text{s.t.} \;\; y_i(w^T x_i + b) \geqslant 1 \quad \text{for } i = 1, \cdots, n. \tag{2.1}$$

The hyperplane w is then estimated using a risk minimisation criterion. Whenever the data set cannot be linearly separated, the extension of SVM is applied to consider the trade-off between miss-classified points and the margin between classes.

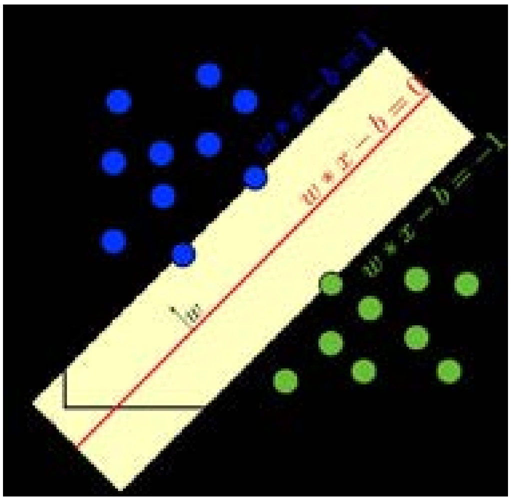


Figure 2.4: The diagram to show the margin between two classes.

2.2.1.2 *Artificial Neural Network*

An artificial neural network (ANN) is a model of neural computation that was first proposed in 1943 by [43]. ANN is inspired by the biological network in the brain. An artificial neural network is based on simple computational blocks called perceptrons or neurons, and the connections between the perceptrons are similar to those between neurons in the brain. There are three layers in this model: an input layer of neurons, a hidden layer and an output layer. There are numerical values assigned to each connection, which are called weights and biases. Input and weight products will be added to biases and fed into the activation function. Here is an expression for the equation:

$$\mathrm{Output}_{i^{th}} = \emptyset\left(\sum_{j=1}^{N} X_j W_{ij} + b_j\right) \tag{2.2}$$

where $\emptyset$ denotes activation function, N denotes the number of neurons, W_{ij} is the weight between i^{th} neuron in the previous layer and j^{th} neuron in the current layer, X is the input and b is the bias.

The backpropagation (BP) algorithm is used to train the ANN [44]. The BP algorithm using the gradient chain rule to update weights based on the loss function. The BP algorithm is composed of four steps:

1. Feeding the input to the network to compute the output
2. Calculating the error based on the loss function.
3. Backpropagate the error to hidden layers using the chain rule.
4. Updating weights and biases.

It is possible for an ANN to contain more than one hidden layer, known as a multilayer perceptron (MLP). An MLP can provide greater performance than an ANN with only one hidden layer [45]. A diagram of the MLP architecture is shown in Figure 2.5.

2.2.2 *Deep Learning Algorithms*

The term "deep learning (DL)" refers to a subset of machine learning algorithms based on ANNs [46]. There are more layers in deep neural networks in order to identify complicated relationships between input and output. Some of the most prominent methods of deep learning include the Variational Autoencoder (VAE), the Convolutional Neural Network (CNN), the Recurrent Neural Network (RNN), and the Generative Adversarial Network (GAN) [47].

Currently, Deep Neural Networks (DNNs) are widely used the image segmentation. Figure 2.6 depicts a neural network with a multi-layer representation in which each layer corresponds to a "distributed representation".

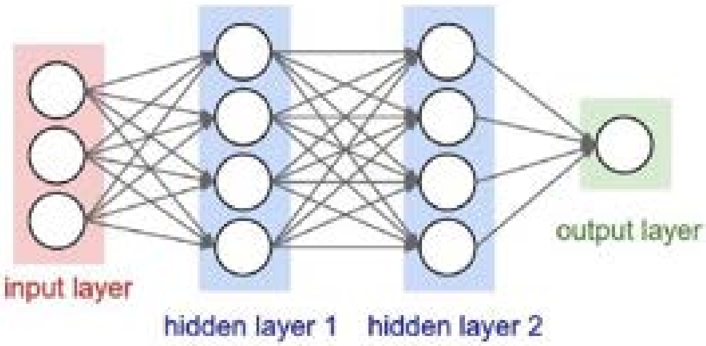


Figure 2.5: MLP architecture.

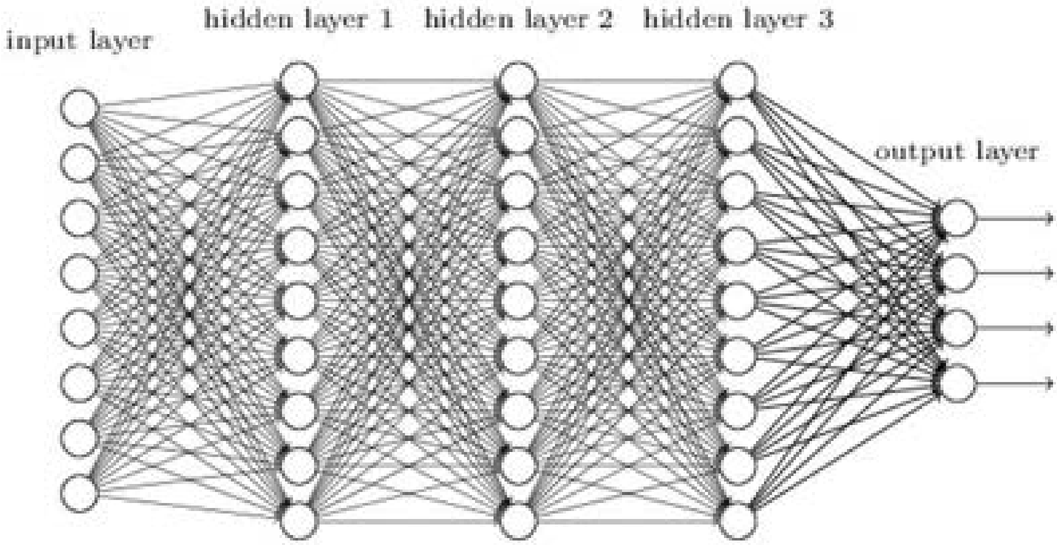


Figure 2.6: The diagram of neural network with 3 hidden layers.

2.2.2.1 *Convolutional Neural Network*

A CNN is a type of deep neural network that has become increasingly popular for solving computer vision tasks. For CNN the input is provided to neurons in order to perform a non-linear connection. It is designed to learn spatial information using multiple blocks, including convolutional layers, fully connected layers, and pooling layers, and it is trained by backpropagation [48].

An important distinction between an ANN and a CNN is the presence of convolutional layers that are arranged in three dimensions, as opposed to ANNs. It is important to note that neurons within any layer are only connected to a small region of input.

Convolution layer: In order to extract features from the grided input, a convolutional layer is the main component. By calculating the scalar product between inputs and

weights, convolutional layers can determine the neurons that are connected to a specific region. A convolutional layer in the CNN contains convolution operation and elementwise activation function to find non-linear connection. Each convolutional layer contains learnable variables known as weights and biases. Convolution is the process of applying a matrix of numbers, called kernels, to inputs, called tensors. The element-wise product between kernel and tensor can be used to extract features that contain spatial information from images. Figure 2.7 illustrates this procedure [49].
Afterwards, the output of the convolution is fed into some activation function, such as a hyperbolic or sigmoid function.

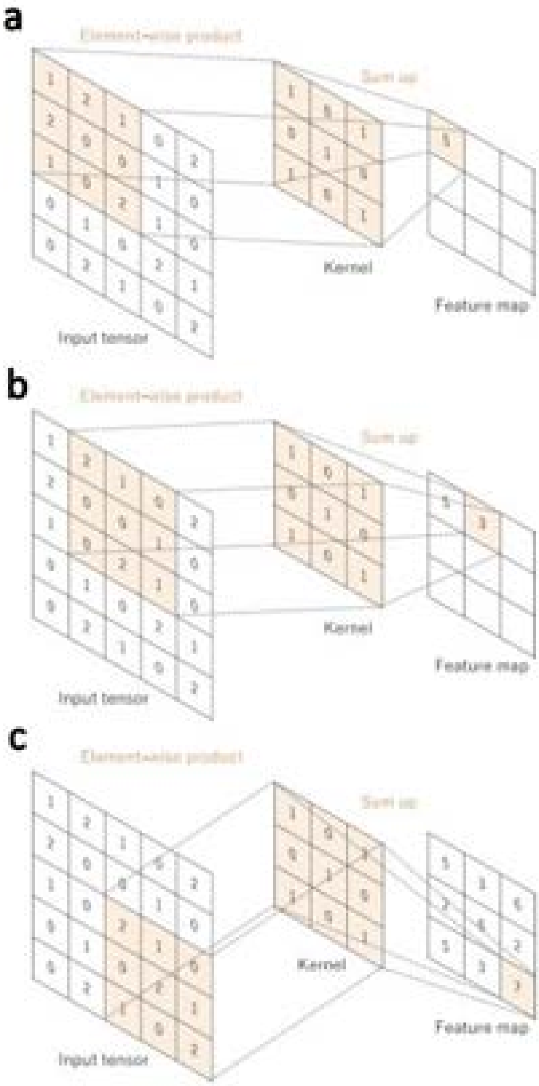


Figure 2.7: Convolution of an input image of size 5 × 5 with a kernel of size 3 × 3 [50].

Pooling layer: In CNN architecture, the pooling layer reduces the size of the feature map by downsampling, which ultimately reduces the number of parameters and complexity of the representations. The pooling layer does not contain any learnable parameters and is only used to reduce the number of learnable parameters. However, the kernel size and stride of the pooling layer are similar to those of the convolution operator. Stride indicates the number of pixels we are moving during each step of the convolution process.

The two most common types of pooling layers are maximum pooling and average pooling. In the max pooling operator, the maximum value of the extracted patch representation is selected and the remainder is discarded. Average pooling is the process of averaging all the values in the patch extracted from the data set.

Fully Connected layer: Fully connected (FC) layer which is known as the dense layer, has a similar architecture to ANN. In the Fully Connected layer, all neurons in one layer are connected to neurons in adjacent layers. Through the FC layer, the spatial information extracted from the last convolution layer will be connected to all outputs via learnable weights in order to transform them into a one-dimensional vector. As with convolutional layers, each fully connected layer is followed by a nonlinear activation function such as ReLU.

A number of designs have been proposed based on CNNs that may show great promise in the field of complex problems. Among the first designs that achieved great success in the recognition of handwritten numbers, LeNet [51] was one of the most successful, followed by AlexNet [52] which demonstrated great potential for computer vision tasks. The following paragraph describes the architecture of some of these algorithms, including LeNet-5, AlexNet, VGG16, and ResNet [53].

There are five layers in LeNet-5, which is why the name LeNet-5 was chosen. It includes three convolutional layers with average pooling, two fully connected layers, and the SoftMax classifier to classify images into different classes, which it was initially designed to classify into ten categories. The convolutional layers consist of six filters with kernel sizes of 5×5 and a pooling layer with size of 2×2 and a stride of 2. As shown in Figure 2.8, the schematic for Lenet-5 can be viewed.

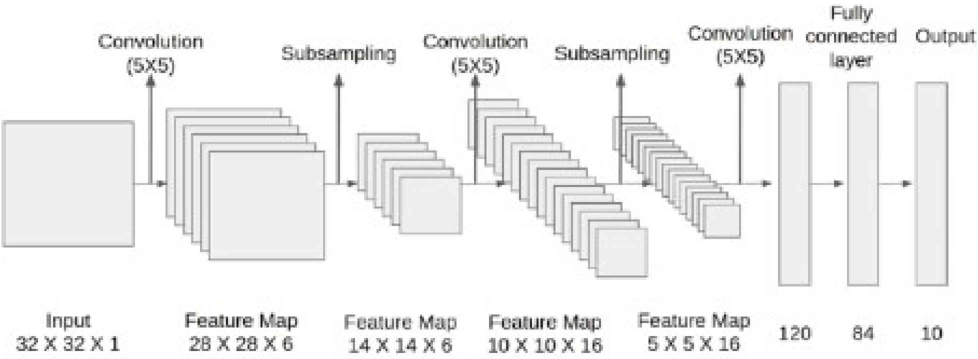


Figure 2.8: Schematic of the architecture of LeNet-5 [51].

In terms of architecture, AlexNet is similar to LeNet, but it is more complex and has larger number of filters. The first convolution layer in AlexNet has 96 filters with a size of 11 × 11 and stride of 4, followed by an ReLU activation function. The next layer is the max pooling layer, which has a kernel size of 3 × 3 and a stride of 2. Following this layer, there are layers with 256 filters with a kernel size of 5 × 5, 384 filters with a kernel size of 3 × 3, and 256 filters with a kernel size of 3 × 3. Each of these layers is followed by a ReLU activation layer and a 3 × 3 max pooling layer. The final layer is composed of two fully connected layers followed by a soft-max. As shown in Figure 2.9, a schematic of AlexNet is presented.

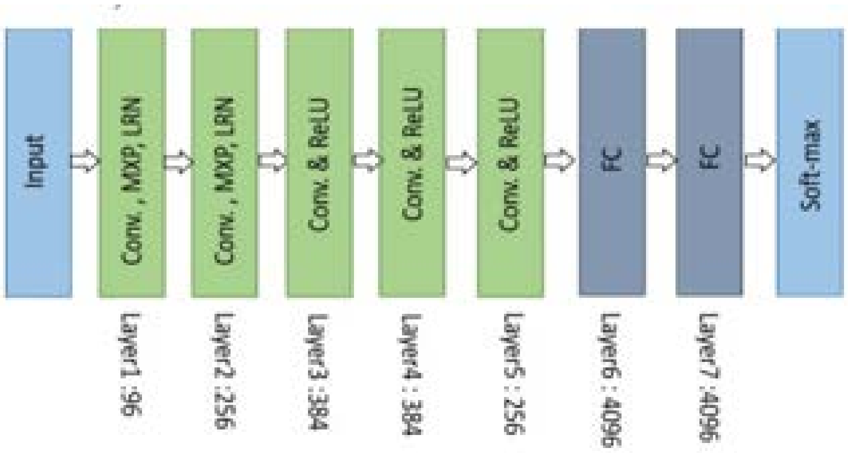


Figure 2.9: Schematic of AlexNet.

According to VGGNet [54], the depth of DNNs has a significant impact on performance for computer vision tasks. As an example, VGG16 has 16 convolutional layers [54]. In VGGNet, each layer consists of two convolution layers followed by an ReLU activation function and a maximum pooling layer. Towards the end of this model, some fully connected layers are also used in conjunction with ReLU activation functions. Based on the VGG architecture, VGG-11, VGG-16, and VGG-19 have been proposed that each have 11, 16 and 19 layers, respectively.

In recent years, residual networks have become increasingly popular. The ResNet architecture utilises residual blocks, which are capable of solving gradient vanishing problems in deep neural networks [53]. In neural networks, gradient vanishing occurs when the gradient of the loss function approaches zero as more layers are added. Thus making it more difficult for the deeper network to be trained [55]. The residual block shown in Figure 2.10 allows us to train deeper models by solving the gradient degradation problem. Today, it is possible to train DNNs with 34, 50, 101, or even 120 layers [48]. Basic ResNet has two pathways: one is the normal convolutional pathway, while the other is the skip layer.

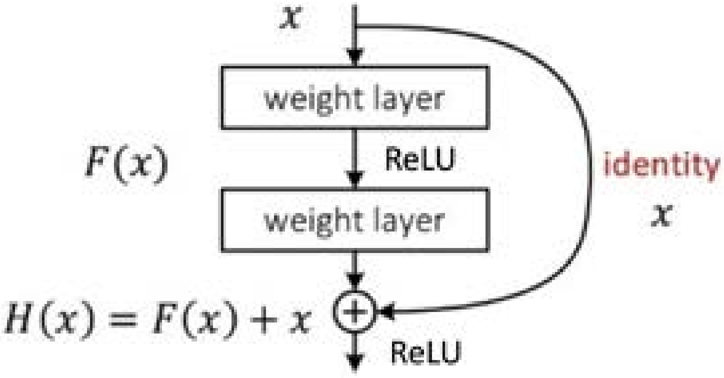


Figure 2.10: Schematic of residual block design [53].

There have been some variant designs of residual block proposed [56–58]. Figure 2.11 illustrates the architecture of one of these residual blocks.

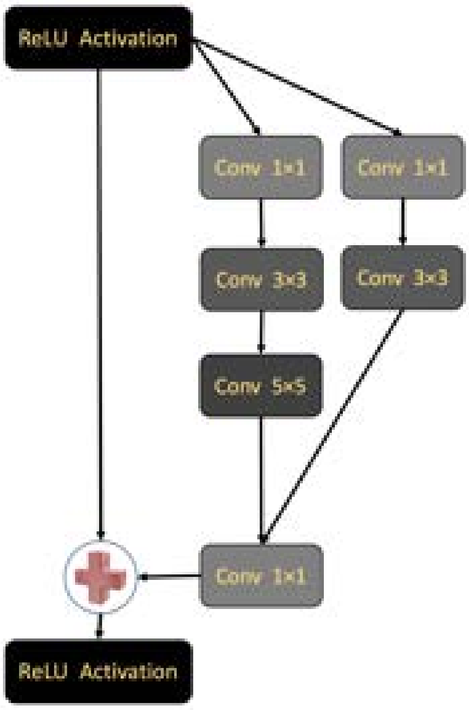


Figure 2.11: One of the variant designs for residual blocks.

This residual block can be represented as follows:

$$x_1 = \mathcal{F}(x_{i-1}^{3\times 3} \odot x_{i-1}^{5\times 5}) + x_{i-1} \tag{2.3}$$

where $\mathcal{F}(\cdot)$ is ReLU activation function and $\odot$ is element-wise multiplication between the output of 5×5 and 3×3.

2.2.2.2 *Recurrent Neural Networks*

All of our thoughts are based on the knowledge we have gained previously, so our brain does not start from scratch when learning something to make a decision. In other words, when you are thinking, you consider what you saw yesterday or last month, and yesterday is more influential than last month because it is recent. CNN networks do not have such an ability to keep knowledge and it seems that all of your memories are lost in the process of happening new incidents [59]. In order to address the aforementioned problem, an RNN has been proposed, which uses sequential information to train. In contrast to traditional DNNs, RNNs are capable of remembering previous information as they process the current data. In recent years, RNNs have demonstrated great potential in the areas of signal processing and natural language processing.

As shown in Figure 2.12, a loop has been designed in the RNN architecture in order to maintain the hidden state of the network. The purpose of this loop is to facilitate the transfer of information from previous steps in sequential data to the current step.

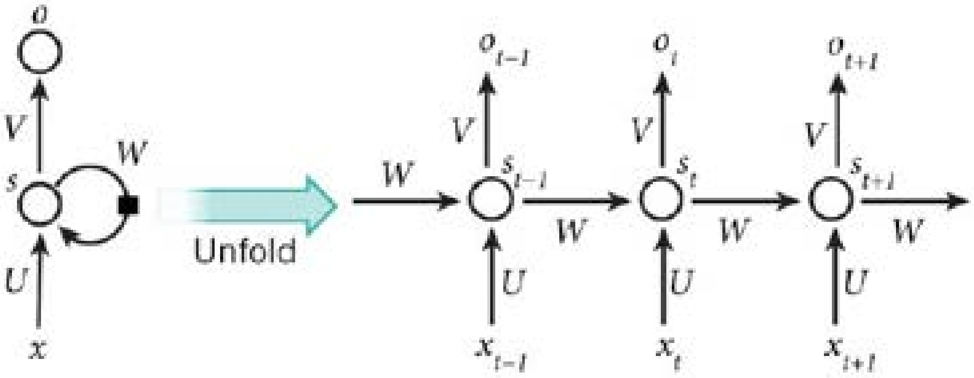


Figure 2.12: The architecture of unfolded loop in RNNs [59].

According to the RNN's loop diagram, input X_t will be fed into the network at time t, and O_t will be the output at that time. According to the unfolded diagram, the hidden state of the neuron from the previous time step S_{t-1} will be used for the state of the current time step S_t which can propagate through sequential data. The following is a mathematical representation of this loop:

$$S_t = \phi\left(UX_t + WS_{t-1}\right) \tag{2.4}$$

where S_t is a hidden state of the neuron and X_t is input at the current time step, U is weight matrix and W is the hidden to hidden weight matrix.

RNN suffers from the problem of exploding or vanishing gradients. RNN is optimised based on backpropagation through time, which requires a large number of multiplications between layers and time steps [60]. A large amount of additional multiplications made it more likely that vanishing or exploding gradients would occur. To resolve the problem of vanishing gradients, long short-term memory (LSTM) has been used in later designs. As a result of LSTM, information is able to persist and neurons are able to remember information for a long period of time [61].

2.2.2.3 *Generative Adversarial Network*

In 2014, the GAN was proposed by Goodfellow [62, 63]. This method is based on the concept of Nash equilibrium and game theory for modelling data utilising conditional probability. Probability density functions were used in previous models but were not successful, but recently, deep learning GANs have proven to be extremely effective at generating synthesised datasets. In other words, each image of an object has its own probability distribution, and GAN should be trained in such a way that, when random samples are provided to the model, it can generate the desired numbers of images with similar probability distribution.

As shown in Figure 2.13, this model comprises two competing networks: A generator (G), which uses noise vectors to generate fake data, and a discriminator (D), which attempts to distinguish between real and fake data. As a result of the training process, the generator is trained to generate realistic fake data to fool the discriminator and the discriminator learns how to classify fake data and real data with greater accuracy. The steps of training G and D are repeated in order to improve both networks.

A GAN is an unsupervised deep learning approach based on likelihood probabilities. The generator is trained to find the true distribution of the data and fool the discriminator to maximise its error. On the other hand, the discriminator seeks to determine whether the distribution is based on real or false data.

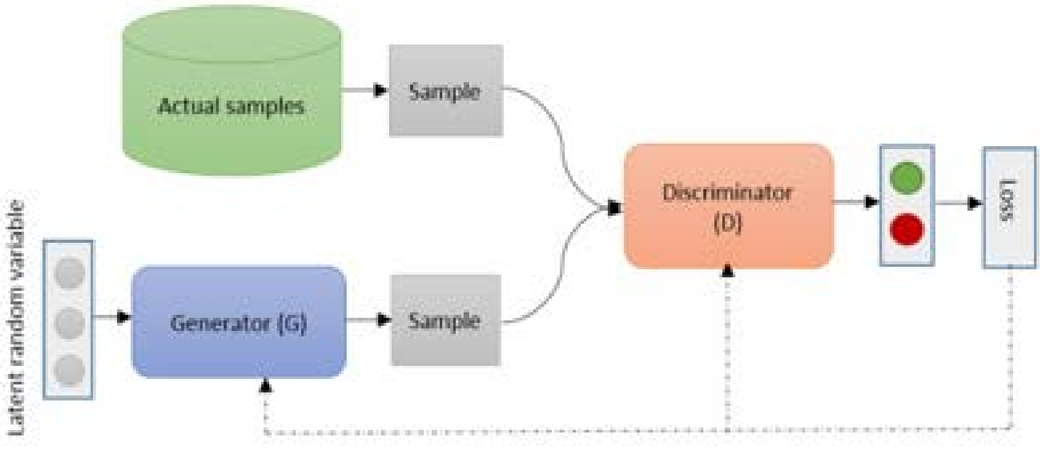


Figure 2.13: Diagram of the GAN architecture.

It is important to note that during the process of training, the generator does not see real data, but it is trained based on the output of the discriminator. The GAN also has some limitations due to the following issues:

- There is no loss function that can provide a practical solution to determine how much data is realistic.
- It is possible for the network to collapse into a specific mode, causing the generator to generate only specific types of data. For example, a network is designed to generate images of dogs, but it is now only generating images of Golden Retrievers.
- Due to the fact that classifying the data into two groups of real and fake is much easier than generating data using a noise vector, the discriminator learns much faster than the generator.

There have been several recent attempts to solve the aforementioned problems by using a new loss function called kl-loss or by changing the design of the network, but the problems are still likely to occur in GANs networks [64].

Training will continue until the output of the generator appears realistic enough for our purposes. The generator and discriminator play a min-max game during the training process which can be expressed as follows:

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x \sim p_{data}}(x)\left[\log\left(D(x)\right)\right] + \mathbb{E}_{z \sim p_z}(z)\left[\log\left(1 - D\left(G(z)\right)\right)\right] \tag{2.5}$$

where D(x) is the output of the discriminator and G(z) is the reconstructed data, using the z latent space.

2.2.3 *Limitations of ML Systems*

ML systems have advanced significantly in recent years, but they still have some limitations. In hospitals, object detection and segmentation are used for urgent and easily missed cases, for example, diagnosing brain artery issues using radiological images [65]. In some of these circumstances, patients have a limited period of time before they face with permanent brain damage. It is necessary for systems to be fast enough with a minimum level of computational complexity to operate real time. As a result, there is always a trade-off between using deeper models (models with more layers) to achieve more accurate results or using shallower networks (models with fewer layers) to have faster diagnosis. A method of reducing the complexity of NN is by reducing its number of outputs and inputs. Using four corners to determine the ROI has reduced the number of outputs in our network, compared to systems using all the pixels. NN-based classifiers may be accelerated by optimising their number of parameters for each particular applications and methods such as drop-off where the unused links are not re-trained [66]. It is also difficult to build a supervised deep-learning system for a new medical imaging task without having access to a sufficient amount of labelled data. To solve the problem of having a small dataset, data augmentation and transfer learning techniques have been found to be effective [67]. The GAN has also been used recently to synthesise data in order to produce a larger dataset [68]. There are two types of NN: traditional methods that have been used for image classification and deep learning models that have been used for feature learning. Deep models have been responsible for most of NN's success in recent years. Although the speed of DL systems and the availability of datasets are important concerns, the greatest issue with DL systems is overfitting and underfitting [69]. An DL system is trained on a set of training data and then applied to predict the outcome of a new task. The system may not perform well on either the training dataset or the test dataset if there is a lack of data or insufficient tuning of the parameters. An underfitted network is unable to perform well for both the training or test sets. In order to prevent underfitting, the network should be trained for more iterations or a larger network should be used. Meanwhile, the goal should be to maximise the accuracy of the network for new data sets, rather than simply work well on the training datasets. In fact, if the model is trained too much to fit the training data, it may end up fitting to the features of the training data rather than to the general rule for real-world dataset [70]. In other words, Overfitting occurs when a model does not generalise well from the observed data to the unseen data [66]. As a result of overfitting, the model performs well on the training set, but underperforms on the testing set. A model that is over-fitted has difficulty finding features in the testing set that is different from the features in the training set.

The reasons for overfitting can be complex but it can be categorised into four groups:

1. Noise learning on the training set: Overfitting can occur when the training set is too small or is very noisy. In this scenario, the network learns noise features rather than features that can be used to predict the unseen data [71].

2. Model complexity: The trade-off in complexity is a compromise between variance and bias. Having too many learnable variables can make a model accurate, but it loses consistency. When there is a low level of consistency, there will be an overfitting to data samples from the train set, but the network will not be accurate enough on the test set. Due to this situation, the models accuracy can be significantly different when applied to different datasets [71].

3. Evaluation method: Despite the fact that some models are capable of producing acceptable results using some evaluation methods, they are overfitted to the training set in the real world and are not able to predict correctly for the unseen data. For example, some super-resolution models may increase the image PSNR, but for unseen data, this may result in loss of more important details. [72].

4. Restricted training set. The training set is large enough, but the variety is insufficient. As an example, the DL method has been trained to classify dog and cat breeds, but in the training set, instead of using different breeds of dog and cat, there are only German Shepherd dogs and Persian cats. As a result, when network classifies the unseen data from another breed of dog, such as a Husky dog, it leads to inaccurate results.

Overfitting can be prevented by implementing the following solutions [73]:

Stop the training early or change the learning rate: This idea behind this strategy goes back to 1970 [74]. Overfitting occurs when the accuracy of algorithms stops improving after some epochs, or even gets worse. To prevent it, training should be stopped after a limited number of epochs. Furthermore, by lowering the learning rate, it is possible to achieve more accurate solutions and prevent overfitting.

Network complexity reduction: It is based on the pruning strategy, which is proposed to reduce the size of the network in relational learning, particularly in DNN. A pruning theory reduces network complexity by removing irrelevant, or less meaningful data, and ultimately improves classification accuracy by preventing overfitting [75].

Increasing the size of the training data: the performance of ML systems is not just influenced by the architecture, but also by the size and nature of the training sets. An expansion of the training set may be achieved by adding some noise to the dataset, data augmentation, or by using GANs to produce a new dataset. Overfitting can be prevented with an expanded dataset [68].

Regularisation: Overfitting a model means it takes into account all the features, even if some of them are not useful or correspond to noise. It is possible to select useful features manually or use dropout for regularisation in order to prevent the model from learning useless features. The dropout strategy involves freezing some weights during the training process, and then training the remaining weights. For each iteration, different weights between layers would be frozen to prevent overfitting [76].

2.2.4 *ML Applications in Medicine*

DL systems are available in many different fields, including computer vision (CV), natural language processing, and signal processing. Since CV is more closely related to our study, we primarily focus on computer vision applications. In the field of CV, deep learning has has been very successful. CV deals with image and video processing and performs tasks such as segmentation, and object classification to be used in variety of applications from diagnostic to surgical robotics.

Recent advances in ML for image classification and object detection can greatly benefit medical imaging and surgical robotics [77].

DL can assist physicians by offering second opinions and flagging concerns about areas in images. This is due to the fact that DL systems are capable of performing some tasks as well as humans, if not better.

It has been shown that the majority of deep learning systems perform well when combined with transfer learning, i.e. [78] they are first trained on a large general dataset like ImageNet [79] that is unrelated to the task of interest, and then retrained using more related data to perform the task at hand.

In the healthcare industry, numerous studies have demonstrated the potential of the DL method for complex diagnostic issues. Examples of potential applications in the healthcare industry include classification of skin cancer [80], detection of melanoma by CAD system in dermatology [81], and diagnosis of early-stage diabetic retinopathy in ophthalmology [82]. Further, deep-learning models have also demonstrated physician-level accuracy in diagnosing a wide range of other diseases, such as identifying moles from melanomas [83] diabetic retinopathy, cardiovascular risk, and referrals from fundus images [84], optical coherence tomography (OCT) [85] of the eye, breast lesion detection from mammogram [86], and spinal analysis from MRI [87].

3 AUTOMATED BONE SEGMENTATION USING HYBRID MODEL OF DRLSE AND TRANSFER LEARNING

3.1 INTRODUCTION

X-rays are one of the most commonly used medical imaging techniques for diagnosing bone fractures. An X-ray machine is cost-effective and an easy-to-use device. As per statistics, X-rays are performed on a very large number of patients each year, and are an extremely valuable diagnostic tool for bone fractures [88].

Despite its value in diagnosing bone and joint abnormalities, radiographs are less useful in detecting some types of fractures and pathological conditions of bones and joints early in their development. The limitations in X-ray images are primarily due to the large focal beam, absorption and reflection of overlapping bones, the patient's pose, underexposure and overexposure of X-ray beams. Therefore, these limitations lead to human error in the diagnosis of bone fractures [89]. For example, in scaphoid fracture detection, some bone segments may overlap causing error in X-ray - based diagnosis [90].

Computer-aided diagnosis (CAD) can significantly assist doctors in reducing their errors, as well as reducing the amount of work they have to do as clinicians. In recent years, the development of artificial intelligence (AI) has led to an expectation that CAD may reduce the number of mistakes in fracture diagnosis. As the first step towards fully automated CAD, we need a fully automated bone segmentation system that allows us to segment all fractured and healthy bones as well as regions surrounding fractures in order to generate proper regions of interest (ROIs). It is important that the CAD system be robust against noise, deformation and translation since the X-ray images are taken in a noisy environment and the resolution is variable depending on the apparatus, operator and fracture type. In this regard, segmentation can be a difficult task, as it involves both detecting targets and precise segmentation.

In general, medical image segmentation methods fall into the following categories: classical image segmentation methods (threshold and edges-based), pattern recognition-based, wavelets- derived models [91–94], and DNN-based techniques such as CNN [95] and UNET [96]. Classical methods cannot be fully automated and require pre-processing such as thresholding and normalisation. Additionally, classical methods have a lower level of accuracy compared with DL-based models.

There are three major issues with all DL-based segmentation methods. Firstly, most of them require meticulous pixel-by-pixel labelling. Second, they do not have the capability of segmenting the area around fractures outside the bones, which could provide clinicians with valuable information. Lastly, there is a lack of accuracy in the case of broken dislocated bones. In this study, we have used transfer learning to train faster regional convolutional neural networks (Faster R-CNN) alongside the use of distance regularised level set evolution (DRLSE) methods to segment the bones. As a result of our method, we do not have to label pixels individually and only need to draw a region of interest for labelling. We can also segment dislocated broken bones and the surrounding area.

Previous research has evaluated a number of image segmentation methods, including Watershed [97], thresholding [98], k-means clustering [99] and others. A number of advanced methods, including Markov random models [100], sparse representations [101], and active contours [102], have been employed. However, these methods require some operator to identify ROIs or background and foreground pixels in order to be initiated. As a result of the development of new deep learning methods, new generation image segmentation methods have been enhanced and are now capable of being automated. According to [103], 100 recent image segmentation methods have been revised, and despite the higher accuracy of DL methods, most of them require considerable labouring effort to complete pixel-wise labelling.

For those working with saliency maps, since they are only segmenting the objects, they are not able to segment the region around the fractures [104]. Therefore, it could result in the loss of a great deal of useful information from that region. In advanced methods of using DL for image segmentation, lots of methods have been used to show the effectiveness of region based CNN (R-CNN) like early methods [105], [106], DeepMask [107] or Dai et al. [108] presented a multiple-stage cascade which can segment from bounding-box proposals.

The proposed hybrid method in this chapter can not only automate the fracture diagnosis but also omit redundant data from regions which correspond to shadows of skin in X-rays, so it can be useful for further research into fully automated fracture diagnosis systems.

3.2 FASTER R-CNN

The classical object detection technique consists of three steps. As a first step, some regions are selected as candidates, no matter if these regions contain the desired object or not. In most cases, more than one thousand regions are selected as candidates by selective search or edge boxes. Then, in the second step, feature vectors are extracted from each region for the purpose of detecting objects. In the last step, the network classifies the regions as background or containing objects. Figure 3.1 illustrates the workflow for object detection.

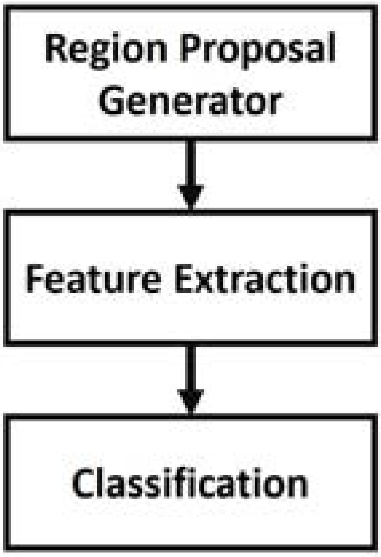


Figure 3.1: Classical object detection steps.

R-CNN was proposed in 2014 [109], in this technique, the only difference with classical object detection methods is that the feature extraction would be performed using CNNs. The rest of the architecture is similar to classical object detection methods. The architecture of R-CNN is presented in Figure 3.2.

The main disadvantage of R-CNN is that this method is very slow since it uses ConvNet for each proposed region without sharing any spatial information.

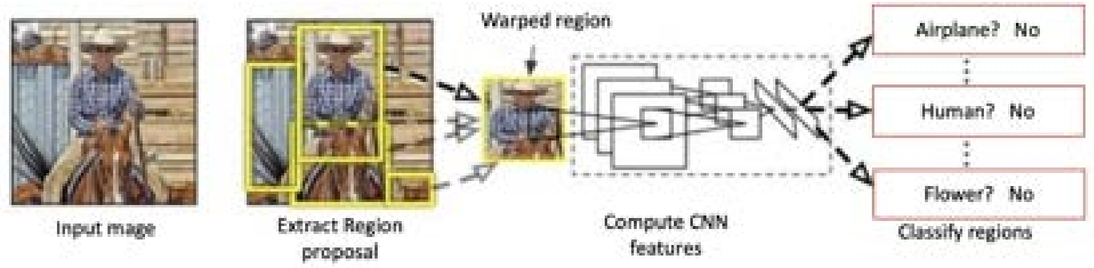


Figure 3.2: An overview of the R-CNN workflow [109].

Later, the Facebook AI researcher proposed the Fast R-CNN method to solve the speed problem in R-CNN [110]. Compared with previous models, the Fast R-CNN has many advantages, including speed, higher detection rate, and single stage training. As part of the Fast R-CNN, the whole image and the proposed region are fed into the network. It then processes the image, using convolution and max pooling layers to produce a feature map [98]. As a final step, we feed the feature map extracted from the proposed region into a fully connected layer for classification into background and desired object classes. The novelty in the design can be summarised as follows [110]:

1. An ROI pooling layer is used to extract fixed length vectors from the proposed regions.
2. As opposed to R-CNN, which is a multi-stage design that includes region proposal generation, feature extraction, and classification layer, Fast R-CNN only consists of one stage.
3. Instead of calculating convolution parameters for each proposed region separately, the fast R-CNN shares convolution parameters across all the proposed regions.

The workflow for Fast R-CNN is shown in Figure 3.3. Using the fast R-CNN design, the entire image is passed through the CNN and the features of the last convolution layer are extracted. At the same time, object proposal windows are obtained through a region proposal algorithm like selective search [111]. Object proposals are rectangular regions on the image in which there is a high probability that there is an object present. A portion of the CNN feature map that belongs to the proposed window is then fed into the ROI Pooling layer. A ROI pooling layer basically divides selected proposal windows into sub-windows.

The ROI Pooling layer's output features are then fed into the FC layers, softmax, and bounding box (BB) regressions. We have used a FC layer with 64 nodes in our models. Using the softmax classification branch, the probability values of ROIs belonging to N categories and one background category are generated. As we only detect one phalange in our model, there are two categories: phalange and background. By using the BB regression branch output, the region proposal algorithm can generate more precise bounding boxes. As a result, we are able to determine the four corners of the rectangular ROI.

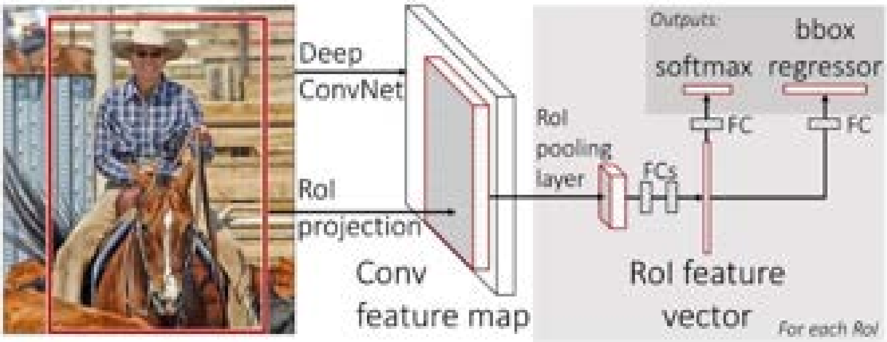


Figure 3.3: An overview of the Fast R-CNN workflow [110].

Fast R-CNN has many advantages over R-CNN, but it also has a number of drawbacks. Firstly, Fast R-CNN is based on the selective search algorithm, which is still a time-consuming process.

In Faster R-CNN [112] the region proposal network (RPN) is proposed, which can share the whole image convolutional feature with the detection network. As a result, no region proposal task is required. Difference between Faster R-CNN and Fast R-CNN design is that Faster R-CNN incorporates RPN into its design, while Fast R-CNN does not. RPNs are fully convolutional networks that calculate object boundaries or objectness scores across images.

In this architecture, there are two novel features:

1. RPN has been used to generate proposal regions of varying sizes. In other words, RPN helps network where to look.

2. In order to reduce the number of computations, RPN and Fast R-CNN share computations.

The RPN module added to Fast R-CNN to design Faster R-CNN is responsible for proposing regions. We have used pre-trained ResNet-50 for our CNN part. The proposed regions are fed into the ROI pooling layer, which then classifies them into classes based on whether they contain desired objects or not. Figure 3.4 illustrates the RPN diagram.

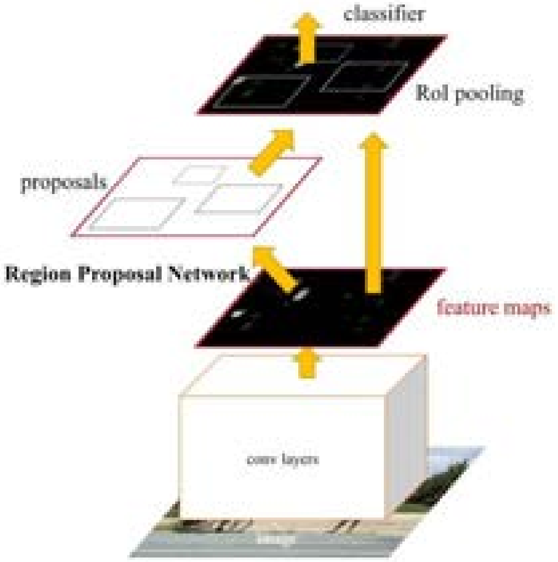


Figure 3.4: An overview of the RPN workflow.

At the end of the network, for each proposed region, the extracted feature vector is fed into two layers of fully connected (FC) network. The first FC layer defines a vector containing the bounding box for the region. The second FC layer provides us with the objectness score for each proposed region. The output for this second FC layer is a vector of two elements. A region proposal is classified as background if the first element is 1 and the second element is 0. If the first element is 0 and the second one is 1, it means that the desired object is located in that region.

The objectness score for RPN training is calculated based on intersection-over-union (IoU). In terms of IoU, it is defined as the ratio between the area of overlapping between the proposed anchor box and the ground truth box over the area of union between the proposed anchor box and the ground truth box. An illustration of an IoU calculation can be found in Figure 3.5.

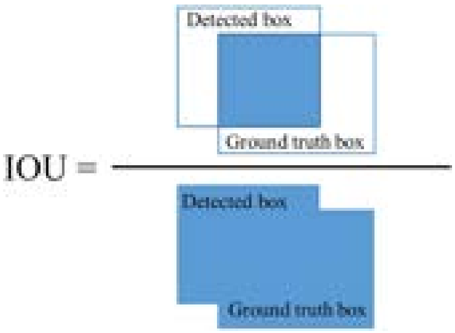


Figure 3.5: IOU equation.

A positive objectness score indicates that a proposed region contains an object, while a negative objectness score indicates a background region. A rectangular ROI with a probability of presenting the object is referred to as an anchor box in the following paragraph. There are four conditions that determine whether an anchor box has a positive or negative objectness score:

1. An anchor box with an IoU greater than 0.7 in conjunction with a ground truth box is considered to have a positive objectness score.

2. In the case where there is no anchor box with an IoU value of greater than 0.7, then the positive objectness score is assigned to the anchor box with the highest IoU value.

3. The anchor boxes would be assigned a negative objective score value if the IoUs for all anchor boxes are less than 0.3.

4. In the event that the anchor box does not have either a positive or a negative value, it would not be used for training.

The two modules of RPN and Fast R-CNN are trained independently in Fast R-CNN, but in Faster R-CNN both modules are trained once by a method known as feature sharing. Both RPN and Fast R-CNN use the same convolutional layers in order to achieve feature sharing. A faster R-CNN network can be trained in two ways [112]:

1. **Alternating training:** In this method the RPN is trained first to tune the weights for its independent convolutional layers and shared layers with fast R-CNN. For independent layers, random values are used as initial values, and shared convolutional layers are initialised using weights from pre-trained models on ImageNet. When the RPN training has been completed and it has been able to propose regions, the tuned weights of the shared layers are retained, and independent layers in Fast R-CNNs are initialised using random variables. In the second phase, the tuned weights of the shared convolutional layers are used to initialise the shared layer to train the RPN again and this process is repeated [112].

2. **Approximate joint training:** This method considers both RPN and Fast R-CNN as a single network. This method has a lower accuracy than the previous method without updating the shared layers after the region proposal, but it can reduce the training time by up to 50%.

It is possible to train Faster R-CNN with other methods, such as non-approximate joint training, but they are all based on the above two methods with only few modifications.

It can be summarised that Faster R-CNN architecture is used to detect objects, but there are some pros and cons associated with it. While recent efforts have been made to create faster and more accurate algorithms such as YOLO [113], [88], Faster R-CNN remains a reliable and accurate model, although it is quite slow.

3.3 TRANSFER LEARNING

ML techniques have demonstrated impressive success in classification, clustering, and regression [114]. In most ML algorithms, training, testing, and validation are assumed to have the same feature space and probability distribution, but this assumption is not accurate in real-life applications. In order to adjust the model to the new feature space, the entire model must be rebuilt and retrained. It is also important to note that training sets can be very expensive and difficult to obtain. Considering the above issue, it may be feasible to transfer the knowledge from the domain of the old task to the domain of the new task to reduce the amount of time, money, and effort required for the model to be adjusted. Transfer learning refers to the process of transferring knowledge from a domain task to a target task [115].

There are a number of examples of how transfer learning can be beneficial. A specific example is web classification, which aims to classify web documents into categories [116]. As an example, manually labelled documents have been used to train a web classifier for an online shop. Implementing the same method for each new website requires cumbersome labelling, but transfer learning can assist in adjusting the current model to accommodate a new website with a different feature set. Transfer learning involves transferring knowledge from one domain to another rather than training a model for each task. Figure 3.6 illustrates the difference between classical ML models and transfer learning.

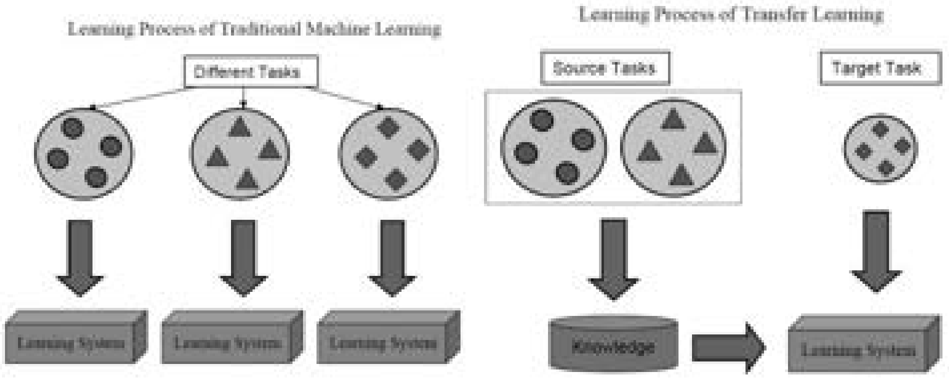


Figure 3.6: Difference between classical ML methods and transfer learning method [115].

There are several questions that need to be answered in order for transfer learning to be successful. These questions include what knowledge should be transferred or how to transfer it. This method of transfer learning is based on the assumption that the knowledge can be transferred between source and target domains. However, transferring irrelevant knowledge from the source domain to the target domain not only does not assist the training process, but can even exacerbate it [115]. On the basis of the answers to the questions listed above, we categorise it into two subgroups, inductive transfer learning and transductive transfer learning [115]:

Inductive transfer learning: There is a difference between a source task and a target task in this category, but their domains are the same. In particular, when the training set in the domain source is large enough, this method would be applied. The model would be trained using labelled datasets from the source domain, then tuned for the target task using a limited number of labelled training sets in the target domain.

Transductive transfer learning. There is a similarity between the source task and the target task in this category, although the domains are completely different. The labelled data in the target domain is either unavailable or of a very small size in this category.

There are some notations and definitions used in transfer learning that are helpful in understanding this concept mathematically. A domain D consists of two components: a feature map X and marginal probability distribution $P(X)$ that
$X = \{x_1, x_2, x_3, \cdots, x_{i-1}, x_i\}$ which $x_{i^{th}}$ is the i^{th} feature of the vector, corresponding to a task. As an example, in classification, each metric for evaluation can be viewed as a feature. As a result of all features being added to X, a feature map can be created that can be analyzed for classification purposes.

X is the space of all possible features of a particular learning sample. For a given domain D, a task T is defined by two parts, a label space Y, and a predictive function $f(\cdot)$, which is learned from the feature vector and label pairs $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in Y$. Y is the set of labels, y_i takes on a value of correspondence class, and $f(x)$ is the learner that predicts the label value for the X. From the definitions above, a domain $D = \{X, P(X)\}$ and a task $T = \{Y, f(\cdot)\}$. D_S is defined as the source domain data where $D_S = \{(x_{S_1}, y_{S_1})..., (x_{S_n}, y_{S_n})\}$, where $x_{S_i} \in X_S$ is the i^{th} data instance of D_S and $y_{S_i} \in Y_S$ is the corresponding class label for x_{S_i}. In the same way, D_T is defined as the target domain data where $D_T = \{(x_{T_1}, y_{T_1}), \cdots, (x_{T_n}, y_{T_n})\}$, where $x_{T_i} \in X_T$ is the i^{th} data instance of D_T and $y_{T_i} \in Y_T$, is the corresponding class label for x_{T_i}. Further, the source task is notated as T_S, the target task as T_T, the source predictive function as $f_S(\cdot)$, and the target predictive function as $f_T(\cdot)$.

Given a source domain D_S with a corresponding source task T_S and a target domain D_T with a corresponding task T_T, transfer learning is the process of improving the target predictive function $f_T(\cdot)$ by using the related information from D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$ [117].

3.4 DISTANCE REGULARISED LEVEL SET EVOLUTION

The level set method for segmenting shapes was proposed by Osher and Setian [116]. The idea was to represent a contour as zero level set from higher dimension function called level set function (LSF) and formulate the contour to capture the shape.

First active contour models [93] are formulated as a dynamic parametric curve with a spatial parameter s and a temporal variable t. The curve equation can be described as follows:

$$\frac{\partial C(s,t)}{\partial t} = FN \tag{3.1}$$

In this equation, F represents the function controlling the contour's movement, and N represents the curve's inward normal vector of the curve C that N can be expressed as $N = -\nabla\phi/|\nabla\phi|$, where ∇ is gradient operator.

The curve defined in terms of a parameterised contour can be transformed into a level set formulation by embedding the dynamic contour $C(s,t)$ into a time-dependent $\phi(x,y,t)$. The curve equation can be expressed as the following partial differential equation:

$$\frac{\partial \phi}{\partial t} = F|\nabla\phi| \tag{3.2}$$

Based on the original paper, we call the level set evolution in our formulation a distance regularised level set evolution (DRLSE) [118].

Even though the output of a level set method is the zero level set of the LSF, it is important to keep the LSF in a good condition so that the level set evolution is stable.

In order to achieve this, the LSF must be smooth, not too steep or too flat during proximity to its zero level set. This condition can be reached by a signed distance functions $\nabla\phi = 1$. The signed distance function is considered as a surface in the 2-D case, therefore, in order to have $\nabla\phi = 1$, the tangent plane of signed distance property surface must have an angle of 45 with the x-y plane and the z-axis.

Let $\phi : \Omega \to \mathbb{R}$ be a LSF defined on a domain Ω. We define an energy function $E(\phi)$ of DRLSE by:

$$E(\emptyset) = \int_\Omega p(|\nabla\emptyset|)dx + \int_\Omega g\delta_\varepsilon(\emptyset)(|\nabla\emptyset|)dx + \int_\Omega gH\varepsilon(-\emptyset)dx \tag{3.3}$$

where $\emptyset$ is a level set function, p is a energy potential function and ∂ and H are the Dirac delta function and the Heaviside function defined respectively by:

$$\delta_\varepsilon(x) = \begin{cases} \frac{1}{2\varepsilon}\left[1+\cos\left(\frac{\pi x}{\varepsilon}\right)\right], & x \leqslant \varepsilon \\ 0, & x > \varepsilon \end{cases} \tag{3.4}$$

,

$$H_\varepsilon(x) = \begin{cases} \frac{1}{2}\left(1+\frac{x}{\varepsilon}+\sin\left(\frac{\pi x}{\varepsilon}\right)\right), & |x| \leqslant \varepsilon \\ 1, & x > \varepsilon \\ 0, & x < -\varepsilon \end{cases} \tag{3.5}$$

and

$$g = \frac{1}{1+|\nabla G_\sigma * I|^2} \tag{3.6}$$

where G_σ is a Gaussian kernel with a standard deviation σ. In this case, the potential function $p(s)$ would have a minimum point at $s = 1$. The potential function p for distance regularisation is defined as:

$$p = p_1(s) = \frac{1}{2}(s-1)^2 \tag{3.7}$$

3.5 METHODOLOGY

One of the most challenging aspects of working with X-ray images is the fact that the quality of these images depends on both the instrument and the operator. Furthermore, X-ray images may be taken from different angles or from different parts of the organ. Therefore, all models that use X-ray images should be robust. Robustness may be measured in terms of lighting changes, deformations, rotations, translations, etc. Deformation and translation are two of the most important factors Figure 3.7. These terms are respectively defined as follows:

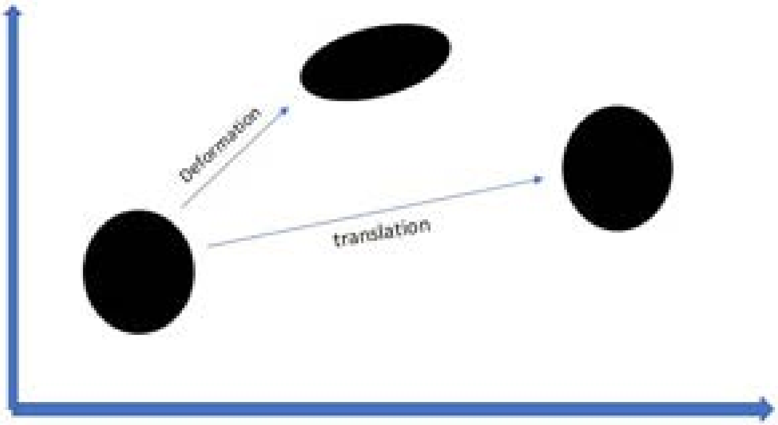


Figure 3.7: Schematic of shape translation and shape transformation

- Translation: $x_c(t) = x(t - c)$
- Deformation: $x_\tau(t) = x(t - \tau(t))$

Therefore, the main conditions for robustness are:

- Translation-invariance

$$\Longrightarrow \forall\, c \in \mathbb{R} \rightarrow \emptyset(x_c) = \emptyset(c) \tag{3.8}$$

- Deformation-invariance

$$\Longrightarrow \forall\, \tau \in \mathbb{R} \rightarrow |\emptyset(x_\tau) - \emptyset(x)| \leqslant K \sup |\nabla_\tau(t)||x|| | \tag{3.9}$$

SIFT

According to Lowe (1999, 2004), SIFT is a descriptor for image-based matching and recognition. In computer vision, this method, together with its related image descriptors, are used for many image segmentation purposes. SIFT is robust to translations, rotations, and scaling transformations in the image domain. However, due to the wide range of shapes and sizes of bones, as well as the similarity of bones, SIFT points are unable to locate a specific phalange. For example, the SIFT points from the second phalange of the ring finger in an X-ray image may be more similar to the SIFTs from the second phalange of the middle finger in another X-ray image from another patient rather than those from the ring finger.

It was initially attempted to use the scale-invariant feature transform (SIFT) points as a method of detecting ROIs or finding some points within the bone, but the results were not satisfactory (Figure 3.8).

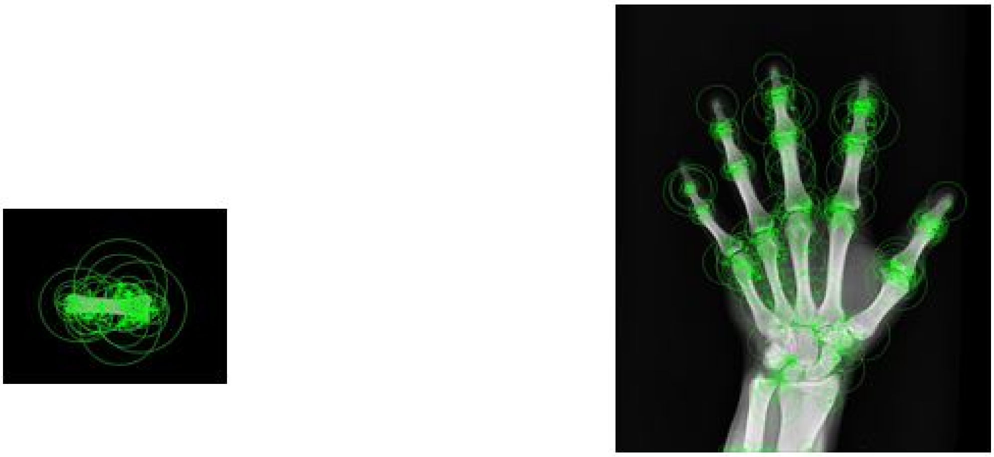


Figure 3.8: To detect ROI, 100 SIFT points from the left image were compared with 100 SIFT points in the left image.

We have picked the first 100 SIFT points of the segmented bone from the entire hand image and tried to see if we can detect that bone in X-Ray hand image of another person. But in the new hand it can only find one SIFT point with low confidence (Figure 3.9).

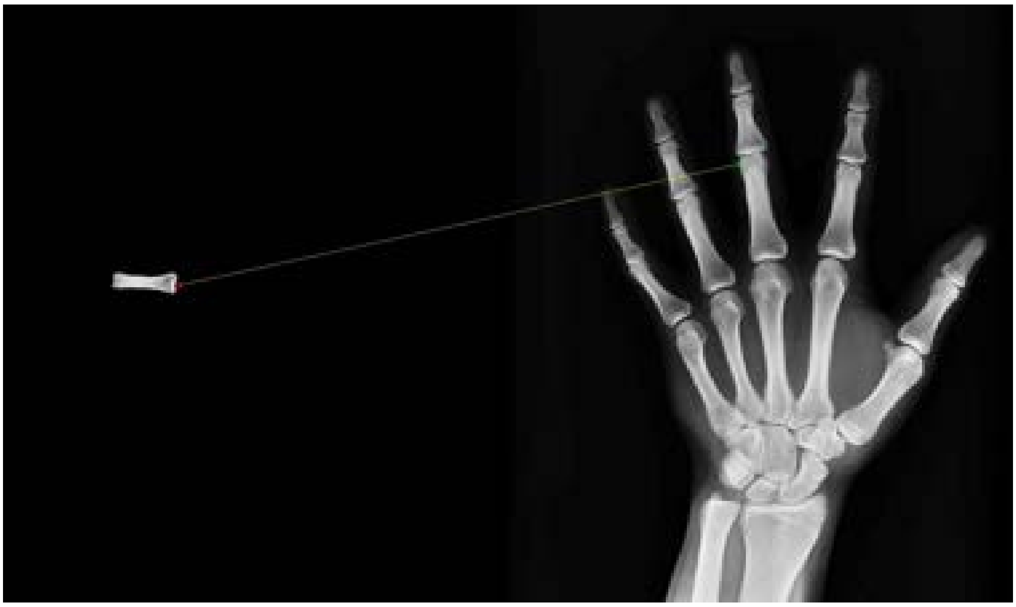


Figure 3.9: Using SIFT points of one intermediate phalange to detect the intermediate phalange of another unseen hand X-ray image.

DESIGN

In order to have a robust model for dealing with changes in X-ray images, deep neural networks can be used.

Currently, both model driven ML, after applying image processing and feature detection, and data-driven ML, by means of DNNs are popular. In our proposed bone segmentation method, we perform bone segmentation in two steps. In the first step, we change the architecture of SoftMax layers in ResNet50 [53] based on the Faster R-CNN design. To implement the mentioned change, we added a dense layer with nodes equal to our output number and softmax as an activation function. Using transfer learning, we freeze the weights in all the layers except those in the last three layers of the pre-trained network, and then we retrain it based on our data. The ROI has then cropped automatically. The DRLSE method was initiated by using four points at the center of ROI to accurately segment the bone. As shown in Figure 3.10, the entire proposed method is schematically represented. We have developed a novel hybrid model that allows us to create an automated model with high accuracy without the need for pixel-by-pixel labelling or a great deal of training.

Since the texture inside bones is homogenous and free of skin artefacts and shadows, ROI detection is able to initiate DRLSE from within the bone, leading to more accurate segmentation.

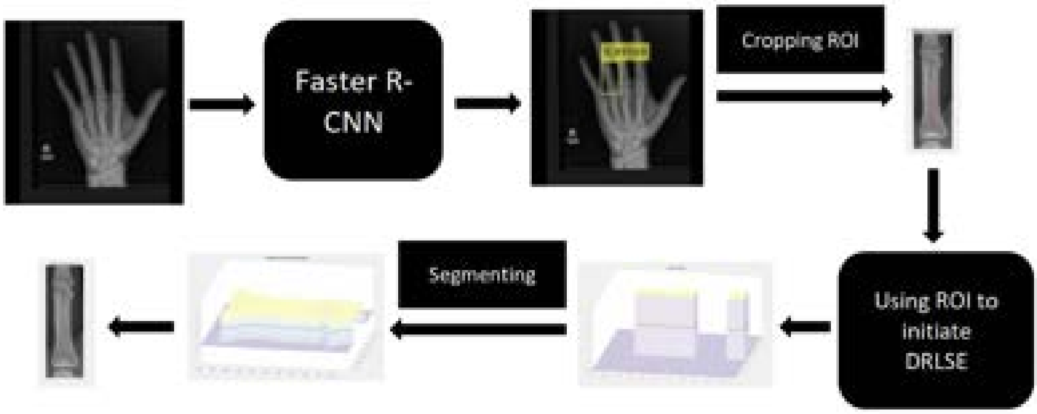


Figure 3.10: Workflow of our bone segmentation method.

FASTER R-CNN

A Faster R-CNN has demonstrated excellent results for the segmentation of images as one of the most recent generations of region-based object detection methods [119]. Despite the fact that this method is slower than innovative methods such as Yolo [88], it is still reliable and accurate. As our system does not run in real-time and accuracy is an important factor for us, we have implemented Faster R-CNN. Faster R-CNN consists of two parts. In the first part, there is the feature extraction network, which is typically a CNN. In the second part, two subnetworks follow the first stage. First, the RPN is used to define object proposals. Secondly, the Faster R-CNN is used to improve and filter the proposed regions and to accurately predict the class of regions (Figure 3.11).

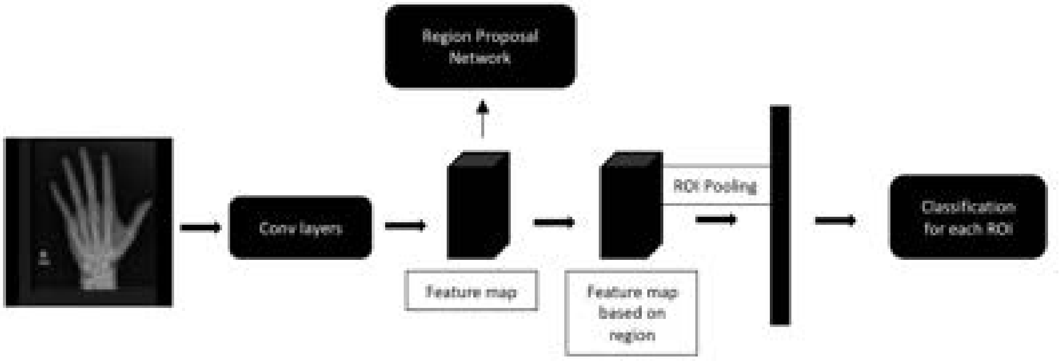


Figure 3.11: Overview of the Faster R-CNN network.

For the purpose of this research, we have used a pre-trained ResNet50 model as a feature extractor. ResNet50 shows great potential in the detection of objects and has been trained extensively for a variety of object detection tasks. As a result, ResNet-50 is a very suitable option to be used as a feature extractor. While ResNet models use skip connections for two layers, our ResNet-50 model skips three layers in order to train such deep models as ResNet-50 and to prevent gradient vanishing. In Figure 3.12, you can see a schematic representation of ResNet-50.

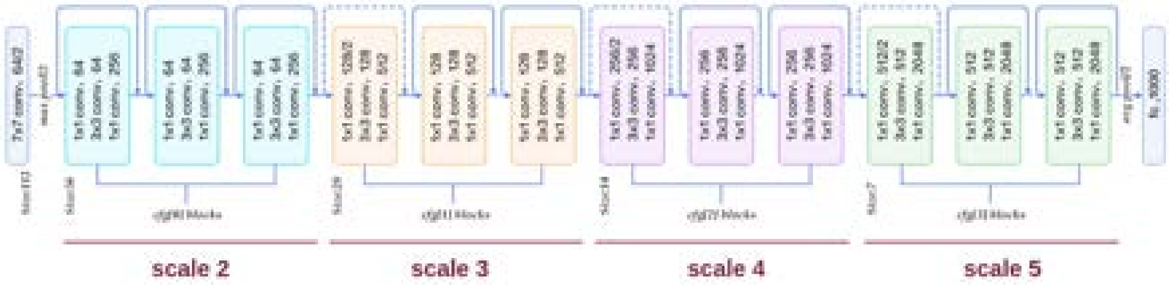


Figure 3.12: The schematic of the our ResNet-50 architecture.

We modified the architecture of the classifier at the end of our models in order to fit the pre-trained model to our dataset. the ResNet50 is used as a feature extractor and the last three classification layers are replaced with new classification layers that includes the members of the object and background classes. (Figure 3.13)

As a short summary, we have frozen all weights in the feature extractor in order to implement transfer learning, and we have retrained the classification layers in order to implement transfer learning. Following the modification of the classification layers of the object classes, the network was trained over 15 epochs and 5700 iterations. Training has been conducted in four steps. As a first step, the RPN is trained. Using the proposed

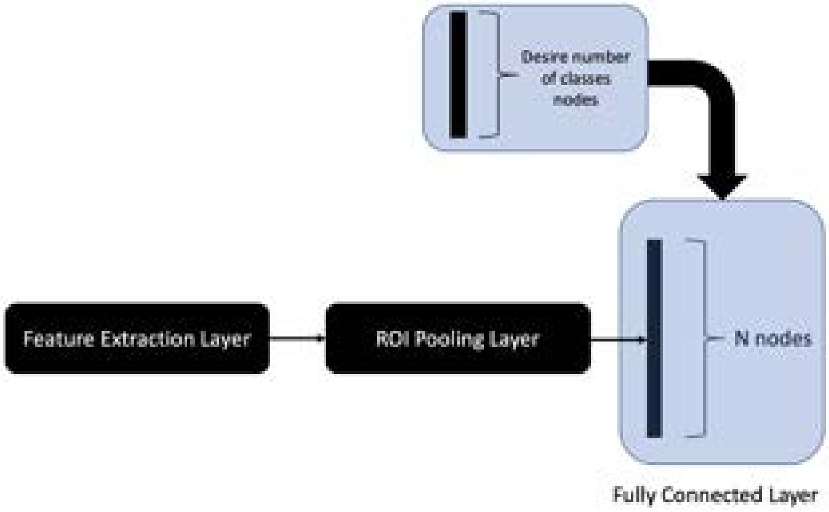


Figure 3.13: The overview of modification on the model to adjust to our dataset.

region from the RPN from step 1, the Faster R-CNN network is trained in step 2. In step 3, the RPN is re-trained using shared weights with Faster R-CNN. The purpose of this step is to reduce the computational cost in the Faster R-CNN method. Step 4 involves re-training the Faster R-CNN using the updated RPN in order to extract the ROIs.

Since we have trained our NN on only one phalange in this study, we have two classes of objects. By using transfer learning, we have used pre-trained ResNet50 that has been used for other segmentation problems and re-trained it for our dataset in order to reduce the computational costs of Faster R-CNN.

DRLSE

Many methods, such as active contours and snakes, shrink the boundaries from the outside to segment the object. These classical methods cannot segment bones accurately on X-ray images due to shadows cast by the skin or overlapping bones (Figure 3.14).

Using the DRLSE method, which defines a zero-level set function, is more accurate than other active contour methods because instead of segmenting the objects by shrinking from the outside, it can segment objects by growing from the inside.

A customised DRLSE method has been designed to improve the DRLSE method by reducing the impact of the distance between the contour edge and the initial points, enabling the segmentation of small pieces of bones falling apart or fractured bones when the pieces are dislocated. Further, instead of setting some blind points to begin the contour, we have used the cropped ROI output of neural network-based object

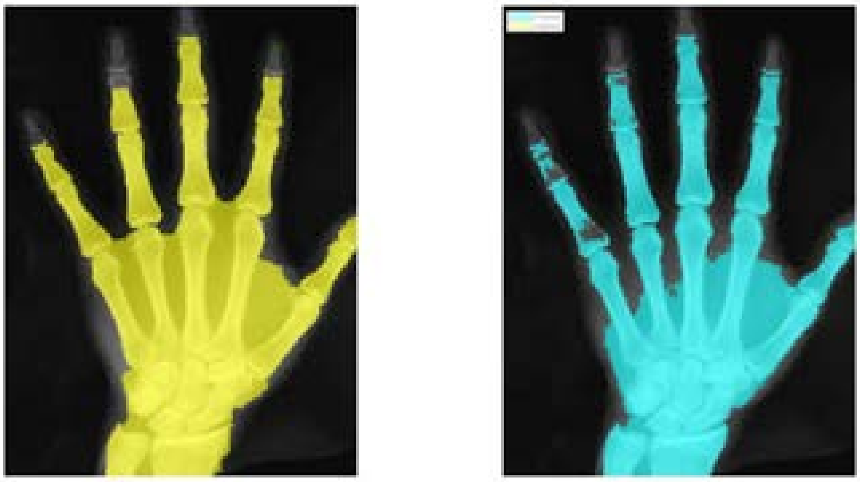


Figure 3.14: Using thresholding and active contour for bone segmentation.

detector to ensure that all the points are set inside the bone as the contour start points. It is possible to define customised DRLSE as follows:

$$E(\emptyset) = \mu R_p(\emptyset) + \lambda L_g(\emptyset) + \alpha A_g(\emptyset) \tag{3.10}$$

where μ, λ, α are control constants, $\emptyset$ is a level set function, $R_p(\emptyset)$ is internal energy term, $L_g(\emptyset)$ is length regularisation term and $A_g(\emptyset)$ is area regularisation term defined as:

$$R_p(\emptyset) = \int_\Omega p(|\nabla\emptyset|)dx \tag{3.11}$$

,

$$L_g(\emptyset) = \int_\Omega g\delta(\emptyset)\nabla\emptyset dx \tag{3.12}$$

and

$$A_g(\emptyset) = \int_\Omega gH(-\emptyset)dx \tag{3.13}$$

According to our model, the internal energy has a decreased impact on the segment region around fractures or small miss-placed pieces.

3.6 experiments

We collected 475 images from the MURA dataset (Stanford University ML Group) with heights and widths between 500 and 1000 pixels. In order to have a varied range of quality and shapes of bones in hand X-ray images, we have resized all the images using a variety of resizing methods, such as adding margins or interpolation techniques. In all the images, the rectangular ROI around the proximal phalange of the ring finger has been manually labelled. We divided the dataset into 375 images for training and

95 images for testing. To avoid data augmentation that may have inverse effects, we cropped the images in our dataset and used the image sub-blocks independently for training. Meanwhile, we applied transfer learning so that we could benefit from the network trained by millions of public images with different resolutions before applying to our own data.

The ROIs for the total number of 475 images in the MURA dataset have been labelled with rectangular boxes. After that, we divided the images into two groups, 380 for training and 95 for testing. Four corners of the ROI box have been used to re-train the pre-trained Faster R-CNN. The training samples in this research have been labelled for proximal phalange of the ring finger, and other bones (such as scaphoid) as may be needed for our future research. The accuracy of our proposed system is approximately 95%. TABLE 3.1 shows the comparison between the results of Faster R-CNN, X-Net and our proposed network as a hybrid model of Faster R-CNN & DRLSE. It has also been evaluated by precision-recall (Figure 3.15).

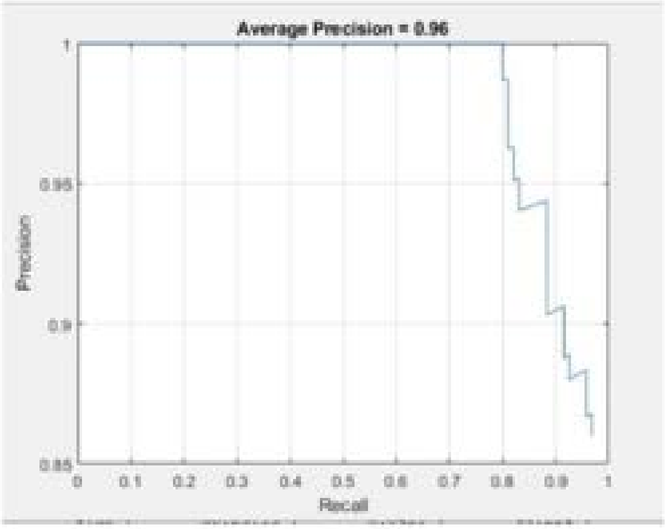


Figure 3.15: Precsion recall curve on the testing set.

In a few cases, the results are not accurate as few ROIs are obtained instead of one (Figure 3.16), or no boxes appear. From 95 test images, in 15 cases, the ROI was duplicated, and for three of them, no ROI was detected.

In order to resolve this issue, we have first set a threshold for the confidence score of each ROI. If the confidence score for an ROI was less than 0.8, that ROI would be disqualified. For the rest of the ROIs we have used customised IOU between ROIs. We have defined this IOU as the ratio of the overlapped area of the ROI with all other ROIs over the area of the union of all ROIs. The IOU is normally checked against ground truth, but ground truth is not available in real-world, which is why we have used the customised IOU. Our results have been improved by 6% as a result of this change. The final result has been cropped for use in the next step. (Figure 3.17).

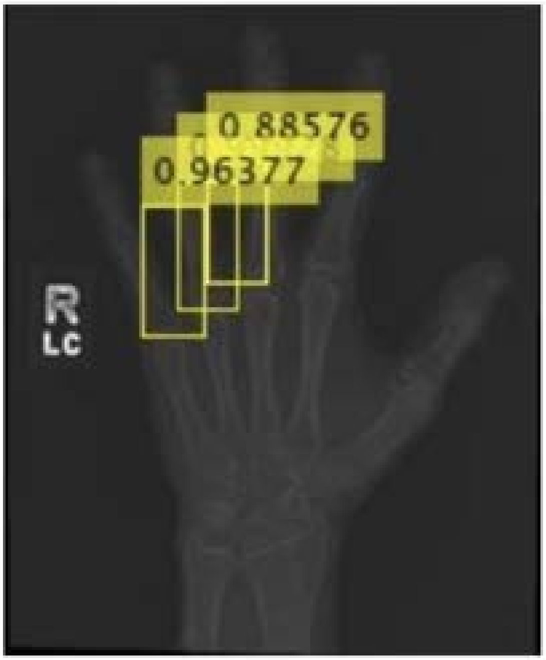


Figure 3.16: An example of duplicated segmented ROI.

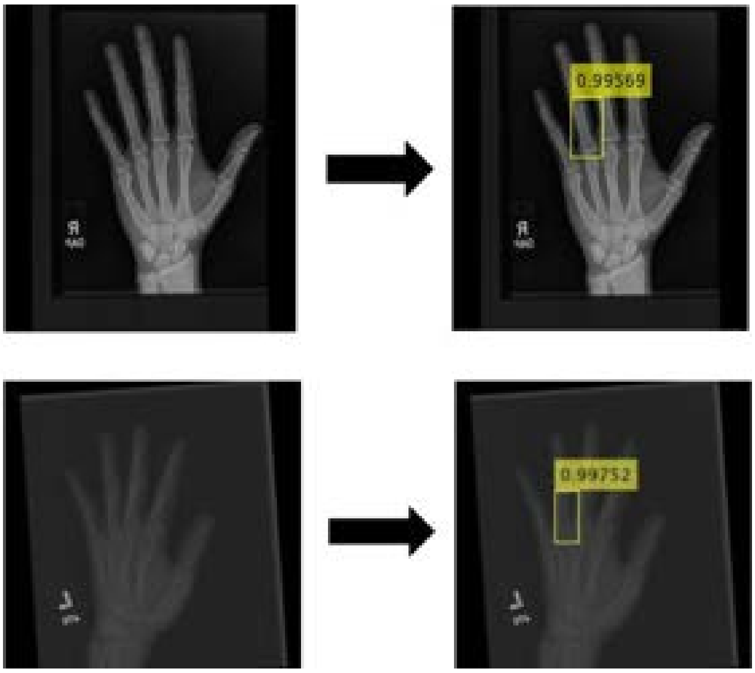


Figure 3.17: Samples of detected ROIs.

The ROI is then cropped in order to initiate the DRLSE contour. Thanks to homogeneity of the bone texture, we initiate the level set function inside the bone (Figure 3.18).

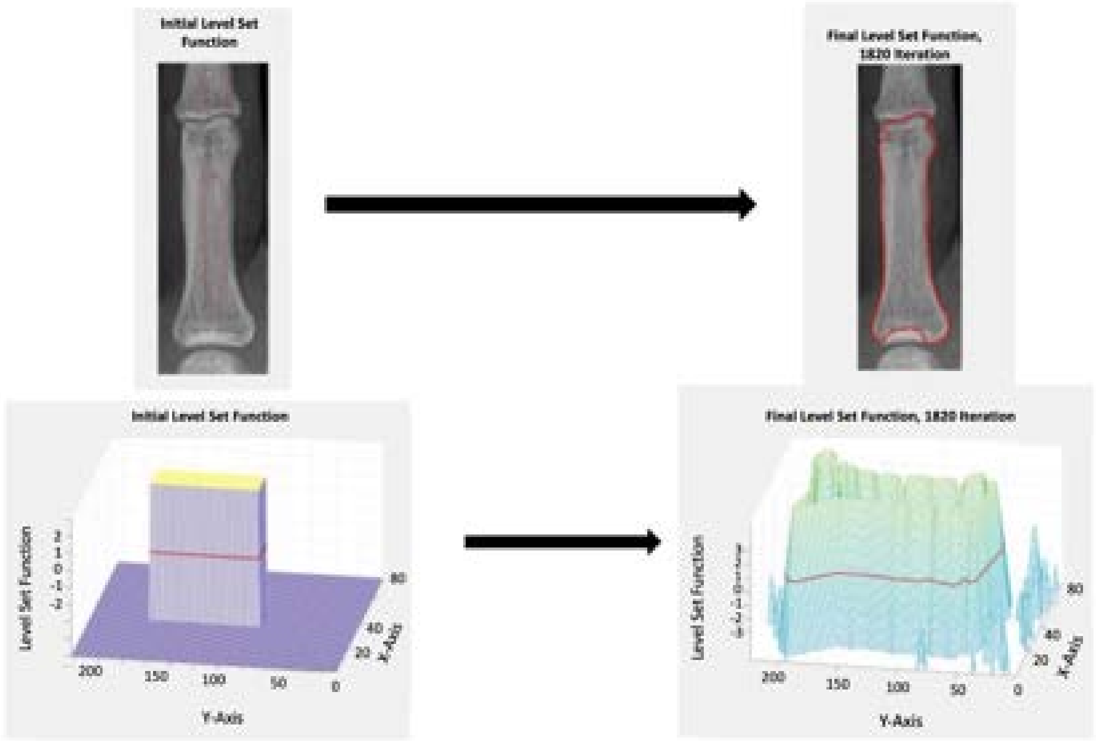

Initial Level Set Function
Final Level Set Function, 1820 Iteration
Initial Level Set Function
Level Set Function
2
1
0
-1
-2
200
150
100
50
0
Y-Axis
80
40
20
X-Axis
Final Level Set Function, 1820 Iteration
Level Set Function
200
150
100
50
0
Y-Axis
80
40
20
X-Axis

Figure 3.18: This diagram illustrates the initial contour on the left and the end result after stabilising LSF on the right.

In recent years much effort has been made for accurate bone segmentation. One of the recent researches is X-Net [120]. We have compared our method with recent attempts on NN to draw ROI, and also we have compared our method with other image processing techniques.

As it showed in Table 3.1 we have compared our method with Faster R-CNN and X-Net in terms of confidence score, average precision and accuracy. As in the first step of our method, we have used Faster R-CNN, the confidence and average precision are similar for both.

Table 3.1: Confidence, Precision and Accuracy results for the whole hand X-ray image.

Method	Confidence	Precision	Accuracy
Faster R-CNN	0.98	0.96	84%
X-Net	0.97	0.93	88%
FRCNN-DRLSE	0.98	0.96	95%

Similar to X-Net, which is applicable to different types of bones, our method can be generalised to different types of bones without losing accuracy. The second stage of our method where we have used our customised DRLSE method is compared with one of the current methods called Watershed. As watershed technique is segmenting all the bones in the image without targeting a specific phalange, we have cropped the phalange accurately to decrease the number of false positives for the detected pixels using watershed. We referred to this as adjusted watershed. The comparison is shown in Table 3.2. These parameters are derived in terms of specificity, sensitivity and accuracy by using true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.14}$$

$$Specificity = \frac{TN}{FP + TN} \tag{3.15}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.16}$$

The DRLSE method can also segment the dislocated broken bone. It has the ability to segment the region around fracture, but other methods are not effective for such an application (Figure 3.19).

Table 3.2: Performance of systems in terms of Specificity, Sensitivity and Accuracy of cropped images.

Method	Specificity	Sensitivity	Accuracy
Watershed	0.962	0.92	0.973
Adjusted Watershed	0.993	0.92	0.978
Customised DRLSE	0.987	0.96	0.984

A model's complexity is determined by the number of training iterations, the number of nodes and number of layers used. Since we have used Transfer Learning, we use fewer training iterations with similar numbers of nodes and layers as for Faster R-CNN. Moreover, although our model is deeper than X-net, it has less complexity than X-net because we have only four nodes in the output, whereas X-net has a number of nodes equal to the image size. In order to compare the complexity of the bone segmentation models, we have analysed their average speed in segmenting the bone on ten images. Times are shown in Table 3.3 By adding the time for Faster R-CNN and DRLSE, we can determine the length of time needed for our model to segment the bone. The X-net is faster than our model when using the GPU, but when using the CPU, because the speed of DRLSE is not that different whether running on CPU or GPU, our model can be faster.

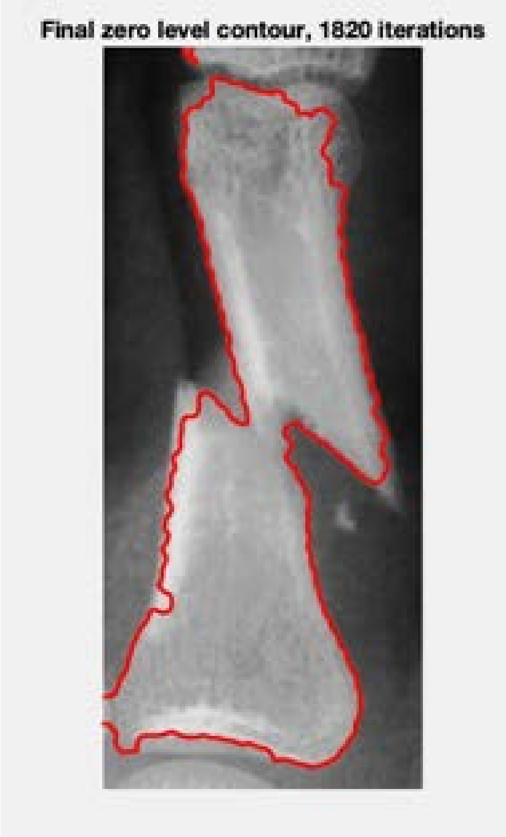


Figure 3.19: The result of our network on dislocated bones.

Table 3.3: The average amount of time it takes to segment bones for each model.

Model \ time (s/per image)	GPU	CPU
Faster R-CNN	1.37	12.74
X-NET	2.23	18.94
Customised DRLSE	1.81	3.45

3.7 CONCLUSION

This research proposes a hybrid model of Faster R-CNN and DRLSE for bone segmentation from X-ray images. In comparison with classical methods like watershed or active contours, it is more accurate. Moreover, unlike the saliency map it can segment the region around the fracture, which has significant diagnostic information. Last but not least, our method does not need any cumbersome pixel-wise labelling in the training stage, and the training can be done using the labelled ROI.

In terms of speed, the X-net has a slight advantage over our model when running on GPU, since the DL-based methods are faster on GPUs, but our model is faster on

CPUs, since the DRLSE method works almost at the same speed on CPUs and GPUs. The speed of these models can vary based on the size of the bone being segmented and the CPU or GPU being used to perform the task.

4 CNNS AND GANS FOR MEDICAL IMAGE SUPER-RESOLUTION

4.1 INTRODUCTION

Reconstruction of a high resolution (HR) and high-quality image from a low-resolution (LR) and low-quality image is known as image super-resolution (SR). A wide range of applications of image SR can be found in fields such as astronomy, computer vision, military, and medicine. The field of SR had received significant attention before deep learning (DL) became popular. Traditionally, a large number of LR and HR images are used to build the learning model. Knowledge gained from previously generated learning models is used to reconstruct the high-frequency details of the LR images in case of not having access to the HR images. There are various types of image interpolation algorithms that can work as SR technique for many image types to a certain extent, such as nearest neighbor, bilinear, and bicubic interpolation methods [121]. But, the results obtained by these algorithms or those combining source separation and learning [122, 123] are not suitable for medical images. This is because they blur the image and also result in a loss of detail in the reconstrained image as these interpolation methods smooth out the images.

Single Image Super Resolution (SISR) was first tackled using prediction-based methods. The linear, bicubic, or Lanczos [124] filtering approaches can be very fast, but they simplify the SISR problem and produce overly smooth solutions. The rapid growth of DL has led to a breakthrough in image SR. The first practical SR approach is based on convolutional neural networks (SRCNN) [125]. The algorithm uses three convolutional layers for feature extraction, nonlinear mapping, and image reconstruction. As compared to traditional methods, SRCNN not only improves the peak signal-to-noise ratio (PSNR), but also gives a great subjective improvement. With the success of SRCNN, more SR algorithms based on DNNs have been derived. In this regard, information distillation networks [126], deep back projection networks [127], zero-shot SR [128], and scale-wise convolutional networks [129] are a few examples.

Approximately 30% of emergency department visits are due to hand trauma [130]. In cases where fractures are not initially evident, the percentage is unlikely to be reduced by missed diagnoses. Patients are distressed by these missed diagnoses, and health centres are burdened with high costs. A new technological tool is thus urgently needed to alleviate these errors and assist emergency services, especially when such services are always under time and resource constraints. X-rays are the first imaging modality used to diagnose fractures in patients. However, not all the fractures are visible on plain

X-Rays. Microfractures or cracks in the scaphoid are such fractures that are difficult to detect on X-rays, even when experienced orthopaedists are involved [131]. When a bone fracture is not diagnosed correctly, it can lead to medical complications such as nonunion, avascular necrosis (AVN), and arthritis [132]. There is a possibility that X-Ray images do not reveal fractures of the hand, wrist, and carpal bones, as they have low contrast and in some cases, like the scaphoid bone, they can "hide" behind other carpal bones. In X-ray radiography, a point source of X-rays is used in combination with an appropriate detector to collect 'snapshot' projections or shadowgraphs of human tissue and bone. As X-rays pass through the body their intensity decreases due to absorption, which is a function of the type of tissue as well as its thickness and density. For example, a soft tissue such as a muscle or an organ results in far less attenuation in comparison with bone. A fundamental feature of X-ray projections (even a theoretical one taken with a true point source) is the apparent blurring of detailed structure and edges. This is called absorption or shape unsharpness. The effect is due to the gradual change in attenuation of the X-ray beam near the periphery of a projected feature i.e., where the beam path through the edge of a physical feature becomes increasingly thin. The terminology of unsharpness applied to this effect is something of a misnomer, as it is not a deficiency in technique, and is a natural consequence of using transmitted radiation to produce a shadow graph image. In addition, X-Ray images are noisy and with low contrast, making it difficult for specialists to detect fractures. This type of fracture is often discovered through 3D images of other modalities, such as MRI, which provides higher spatial resolution and more accurate information. A medical diagnosis can be made using both X-ray and MRI images. In clinical practice, such as for detecting bone fractures in Accident and Emergency (A&E) units, if the fracture is not visible in the X-Ray image, often taking an MRI becomes inevitable, which involves cost and delays in the treatment [133]. Any effective enhancement of cheap X-ray data can therefore avoid the high cost of MRI and is likely to provide the same diagnostic information.

This section aims to improve the resolution of the original X-ray images using DNNs based image reconstruction so that bone microfractures can be detected more easily. To this end, we used the SR method along with the bone segmentation method introduced in the previous section. There are multiple DNNs that can be used for image SR, and among them, the one that fits X-ray images best should be selected. A DNN will then be further trained and optimised to meet our requirements, and during this training process, there are two main steps: 1) Feature extraction: using a small number of LR-HR X-ray image pairs, finding the mapping relationships between them, and applying this relationship to another new image to obtain the HR version. 2) Determine the optimal cost/loss function for X-ray enhancement. PSNR and Structural Similarity Index Measure (SSIM) are usually used to measure DNN performance. Even if an image has the highest PSNR/SSIM, this does not guarantee that it has the best visual effects. It is important to choose a cost function that suits our project objectives. Next, is applying generative adversarial network (GAN) and CNN to optimise the DNN to minimise the cost function.

One of the important issues in traditional SR techniques is that these methods work efficiently when we have few LR images and we want to generate one HR image. But, in the case of having only one LR image, constructing an HR image is not accurate enough. In this part of the research CNN and GAN are used for single-frame super-resolution techniques that can be applied when we only have one LR image as an input.

4.2 SRCNN

In the SRCNN method, a direct mapping is learned between LR and HR images. CNNs take LR images as inputs and output the HR ones. We further show that the traditional sparse-coding-based SR methods can also be viewed as a deep convolutional network. But unlike traditional methods that handle each component separately, this approach optimises all layers simultaneously.

A single LR image is first upscaled using bicubic interpolation, which is the only pre-processing performed. The interpolated image is denoted by Y. The goal is to recover from Y an image F(Y) that is as similar as possible to the ground truth high-resolution image X. Although Y has the same size as X, we call Y LR image, X ground truth and F(Y) HR reconstructed image. The mapping F is composed of three operations of patch extraction and representation, non-linear mapping and reconstruction. [121].

4.2.1 *Patch Extraction and Representation*

This operation extracts patches from the LR image Y and represents them as high dimensional vectors. Each of these vectors is composed of a set of feature maps, whose dimensionality determines how many features are used.

Image restoration strategies [134] typically involve extracting patches and representing them using systems pre-trained by PCA features. The technique is similar to convolving an image with a series of filters. The formulation includes optimisation of these parameters. Formally, our first layer is expressed as an operation F_1:

$$F_1(Y) = \max(0, W_1 \times Y + B_1), \tag{4.1}$$

W1 and B_1 represent the weights and biases, and the image Y has been convolved with this layer. $Max(0, x)$ represents the ReLU [135]. This layer has n_1 number of filters with the size of $c \times f_1 \times f_1$, where c represents the number of channels for the input image, for example, RGB images have three channels, and f_1 refers to the size of the filter.

4.2.2 *Non-Linear Mapping*

In this operation, each high-dimensional vector is nonlinearly mapped onto another high-dimensional vector. The mapped vectors conceptually represent conversion from LR to HR patches.

For each patch, the first layer extracts an n1-dimensional feature map. This second operation is mapping each n1-dimensional vector into an n_2-dimensional vector. This definition is true when 1×1 filter has been used, but it can be generalised to bigger filter sizes like 3×3 and 5×5.

$$F_2(Y) = \max(0, W_2 \times F_1(Y) + B_2), \tag{4.2}$$

Like the previous layer W_2 and B_2 represent weight and biases. This layer has n_2 number of filters with size $n_1 \times f_2 \times f_2$, which f_2 is the size of filter and n_1 is the numbers of filter from previous layer.

4.2.3 *Reconstruction*

In this process, the HR patch-wise representations are aggregated to create the final HR image. In this case, the generated image should be similar to the ground truth image X.

To generate HR image using the output of previous layer we use.

$$F(Y) = W_3 \times F_2(Y) + B_3 \tag{4.3}$$

which W_3 and B_3 are weight and bias. The generated high resolution image is $F(Y)$.

All three operations above lead to the same form of convolutional layer, despite their different intuitions. All three operations are combined to form a CNN. An overview of the network is shown in Figure 4.1. The filtering weights and biases are to be optimised in this model. These concepts are directly used in the design of a SRCNN [136].

SRCNN has shown significant potential in single frame SR technique. In our study, SRCNN can help us to generate HR image using only one LR image. In order to train SRCNN, the loss function will be defined as average of mean squared error (MSE) between generated image and HR image in the training set. The equation is defined as following:

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \|F(Y_i, \Theta) - X_i\|^2, \tag{4.4}$$

4.3 SRGAN

GAN has become popular due to its applications to various image or data conversion systems. Therefore, it is expected to perform well in highly desired super-resolution

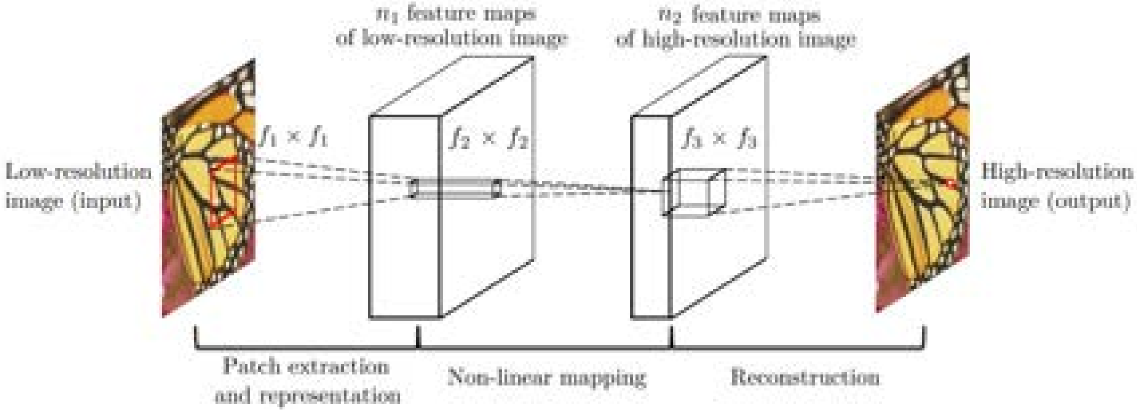

n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
Low-resolution image (input)
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

Figure 4.1: Shematic of SRCNN workflow.

of medical images [137–139]. The ability of GANs to generate data without explicitly modelling the probability density function has gained much attention in computer vision. GAN works based on competition and equilibrium. As seen in Figure 4.2, GANs work by competing between the functions of two distinct adaptive units - generator (G) and discriminator (D). The Generator generates the desired content by minimisation of a loss between the target data and the reconstructed one, while the discriminator maximises discrimination between content and possible undesired information. Each of the two blocks is a DNN, usually a CNN. Using the discriminator's adversarial loss, unlabelled samples can be included in training leading to more consistency.and higher order consistency can be imposed. Many applications have proven to benefit from this, including domain adaptation, data augmentation, image-to-image translation, and SR. Medical imaging and image processing researchers have been attracted to these properties. The training of a GAN is challenging since we need to train both neural networks (each with its own cost) and balance the operation mechanism between the two. By using this mechanism, it is possible to obtain a very good "G" that can be used for producing an HR image.

While convolutional neural networks have improved the accuracy and speed of single-image super-resolution, a key problem remains to recover details in the large-scale image [140]. Most recent work has focused on minimising the mean squared reconstruction error, but like it is mentioned before, MSE is not enough to evaluate the missed details in HR images. The reconstructed image can have sufficient noise to peak ratio but missed details can be seen in HR. SRGAN is a GAN for single image super-resolution (SISR). It is capable of generating details in HR images even for large upscaling values. In order to prevent losing details, state-of-the-art loss function namely content loss has been proposed. The content loss helps the network recover missed details in HR image generated from other SISR techniques. The SRGAN uses a basic GAN network [68] to reconstruct the HR images using LR images as input.

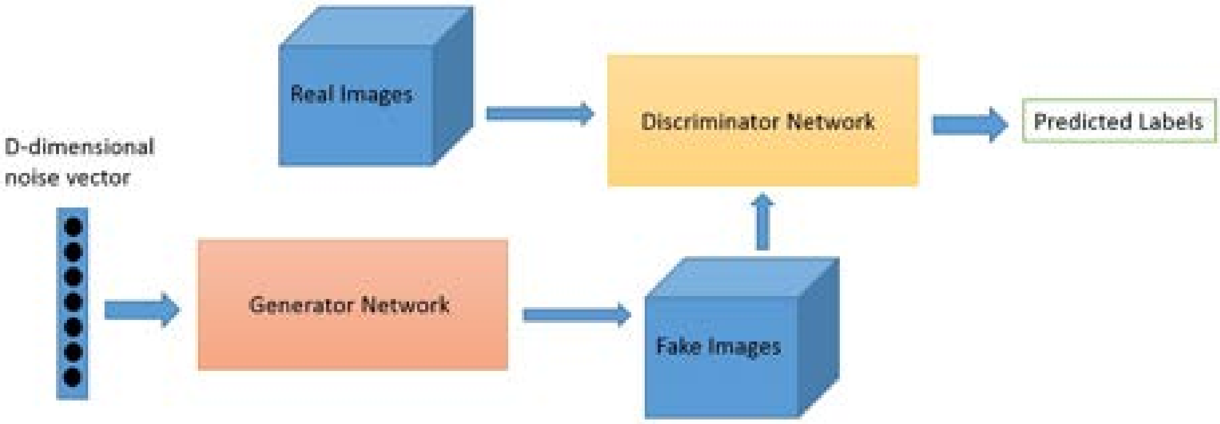


Figure 4.2: Workflow schematic of GAN.

Moreover, similarity loss has been used alongside the content loss to have more realistic HR images. In other words, the similarity loss evaluates how closely the intensity of each pixel in the reconstructed image matches the intensity of the HR image, and the content loss evaluates how much detail has been recovered in the reconstructed image [18]. The reason for using two types of loss function is that the blurred reconstructed image may have a low similarity loss but a high content loss. Photo-realistic textures can be recovered from heavily downscaled images by SRGAN [141]. By using SRGAN, significant gains in perceptual quality can be observed based on an extensive mean-opinion-score (MOS) test. SRGAN's MOS is closer to that of ground truth images than other methods.

From a noise vector, GANs can generate natural-looking images with lots of spatial detail. The potential and procedure of GAN are helping to reconstruct realistic details in HR images from LR images.

Based on a LR input image I^{LR}, an HR reconstructed image I^{SR} is estimated. The I^{LR} is an LR version of the I^{HR} . For simulation purposes, in order to obtain the I^{LR} to train the network, a Gaussian filter can be applied to the HR image I^{HR} followed by downsampling the image by a sampling factor r. The I^{LR} serves as input to the network to generate the I^{SR}, and the I^{HR} serves as the ground truth.

In SRGAN, the G will be used to reconstruct the I^{SR} using I^{LR} as an input and the D will attempt to distinguish between I^{SR} and I^{HR}. It is important to note that the generator does not have access to the I^{HR} images and will only be trained using the output of the discriminator. The two networks will continue to fool one another until the point where D cannot distinguish between the output of the G network, ISR and the I^{HR}.

For I^{LR} a tensor with size $W \times H \times C$ will be defined, which C is the number of channels of image which is 1 for greyscale images and 3 for RGB images, and $W \times H$ is the image size. Respectively, tensor with the size of $rW \times rH \times C$ is defined for I^{HR}, I^{SR}.

When we transform a picture into a tensor, each of the pixels in the picture becomes a tensor in the position they hold in the picture. Our models can be trained using these tensors once they are transformed.

G is trained as a feed-forward CNN parametrised by θG, where $\theta G = \{W_1 : L; b_1 : L\}$ represents the weights and biases of the G network.

In order to optimise the G network, following equation is solved:

$$\widehat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l^{SR} \left(G_{\theta_G}(I_n^{LR}), I_n^{HR} \right) \tag{4.5}$$

where N is the number of images in our dataset.

Like the original GAN architecture [68], the D is optimised by solving following equation:

$$\begin{aligned} \min_{\theta_G} \max_{\theta_D} \; & E_{I^{HR} \sim p_{train}(I^{HR})} \left[\log D_{\theta_D}(I^{HR}) \right] \\ & + E_{I^{HR} \sim p_G(I^{LR})} \left[\log \left(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})) \right) \right] \end{aligned} \tag{4.6}$$

The overview architecture of SRGAN used in the original SRGAN paper is shown in Figure 4.3.

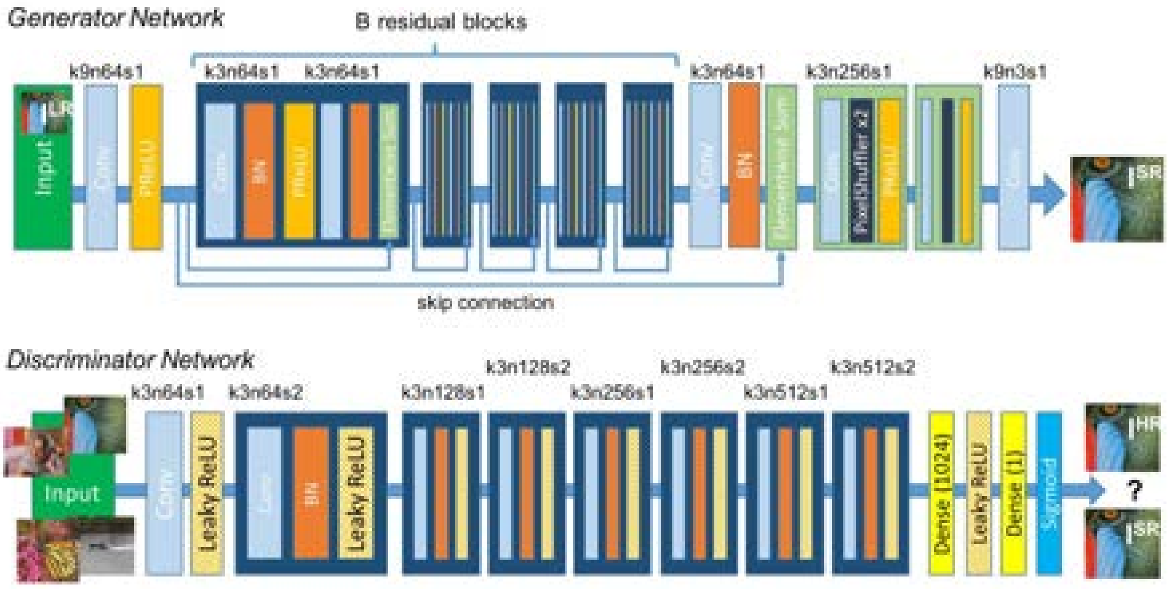


Figure 4.3: Overview of SRGAN architecture [142].

PERCEPTUAL LOSS

Usually, the loss function is based on MSE or pixel-wise difference between the constructed and ground truth images. But, in this model, the perceptual loss function is defined as the summation of adversarial loss and content loss. As the adversarial loss is considerably bigger than the content loss in most of the cases, in order to better train the network better, the weighted adversarial would be used. The perceptual loss function is defined as:

$$l^{SR} = \underbrace{\underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}} \tag{4.7}$$

The common way to define content loss is by MSE, which is the pixel wise difference between the constructed image and ground truth formulated as follows:

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \left(I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y}\right)^2 \tag{4.8}$$

As mentioned before, having less MSE does not guarantee that the network can reconstruct the images with sufficient details. Thus, another loss function has been defined for SRGAN which is called VGG loss:

$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}\left(G_{\theta_G}(I^{LR})\right)_{x,y}\right)^2 \tag{4.9}$$

where $\phi_{i,j}$ is the feature map between j^{th} and i^{th} layer. The generator is represented by G, the LR image by I^{LR}, and the reconstructed image by $G(I^{LR})$. The size of each feature map is determined by W and H. Although the VGG loss can help the network be trained to reconstruct more details for HR images, these details should be realistic as well. To achieve this, another term called adversarial loss is used in SRGAN:

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}\left(G_{\theta_G}(I^{LR})\right) \tag{4.10}$$

This loss function is based on the output of D to check if I^{SR} is a realistic HR image.

4.4 METHODOLOGY

In this section different architectures of SRCNN and SRGAN have been evaluated for X-ray images. We also analysed the effect of these methods on the regions around fractures. To train these models, we have down-sampled the available HR images with the scale of four by using Gaussian filters to make some LR images. In the next step the LR images have been used as input and HR images are used as ground truth to train our models.

4.4.1 *SRCNN*

As our dataset consists of HR images, Gaussian filters have been used in down-sampling the images by factor of four to generate LR images. We used LR images as input for our network. In order to have the same size HR and LR images, bicubic interpolation is used to pre-process the LR images. Even for reconstructed images, the output of our network was down-sampled to have the same size for I^{HR}, I^{LR} and I^{SR} . To create the I^{SR} enhanced image, the pre-processed I^{LR} image is applied as input of the SRCNN, while the reconstructed image should be as similar as possible to the original I^{HR} image. Accordingly, the SRCNN structure resembles the diagram in Figure 4.4. Depending on the image size and the expected quality of the HR image, the number of layers in each stage can vary. Using more training data and iterating the learning process improves SRCNN's results quantitatively.

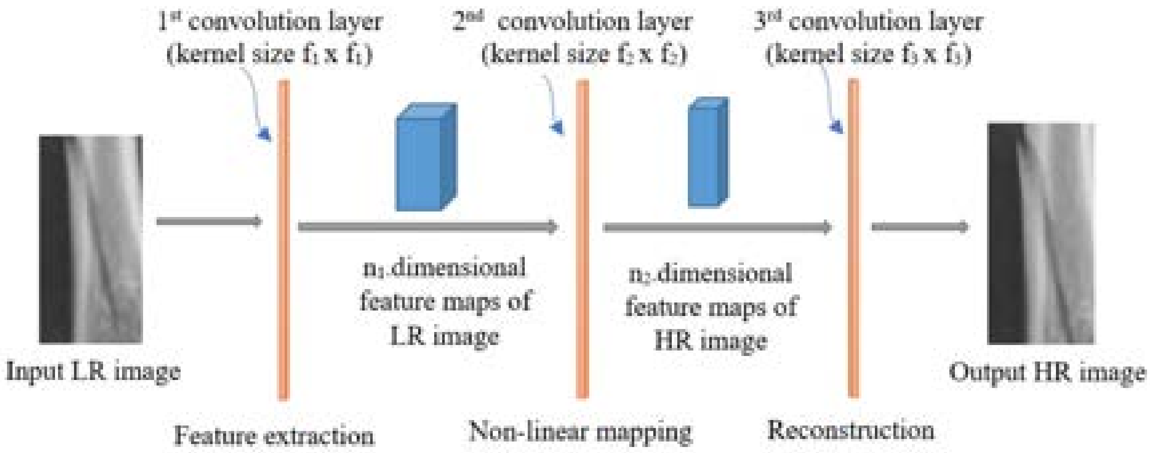


Figure 4.4: The workflow of our SRCNN system.

The first layer is used for patch extraction. This layer extracts multiple image patches from LR images, converts each patch to multiple-dimensional feature vectors by convolution, and forms feature maps. The kernel size is $9 \times 9 \times C$ for RGB images, and 64 filters are used. The number of channels (C) is three for RGB images and for greyscale images, it is one. At the end an ReLU activation function is used.

The second layer is non-linear mapping. The LR features are mapped to HR features in this layer, which is similar to finding a sparse dictionary. One-to-one convolutional layer is used for this layer, so the kernel size is $1 \times 1 \times 64$ and 32 filters are used. For this layer we also used ReLU for the activation function.

The last one is the image reconstruction layer. This layer uses features corresponding to HR image to reconstruct the HR image. This process is similar to dictionary learning [143]. This layer does not have any activation layer, so the convolutional operation is linear. For this layer, the kernel size is $5 \times 5 \times 32$ and as it needs to reconstruct one HR image, one filter is used.

To have a better visual understanding of how these layers are working, the corresponding output feature map of these layers is shown in Figure 4.5.

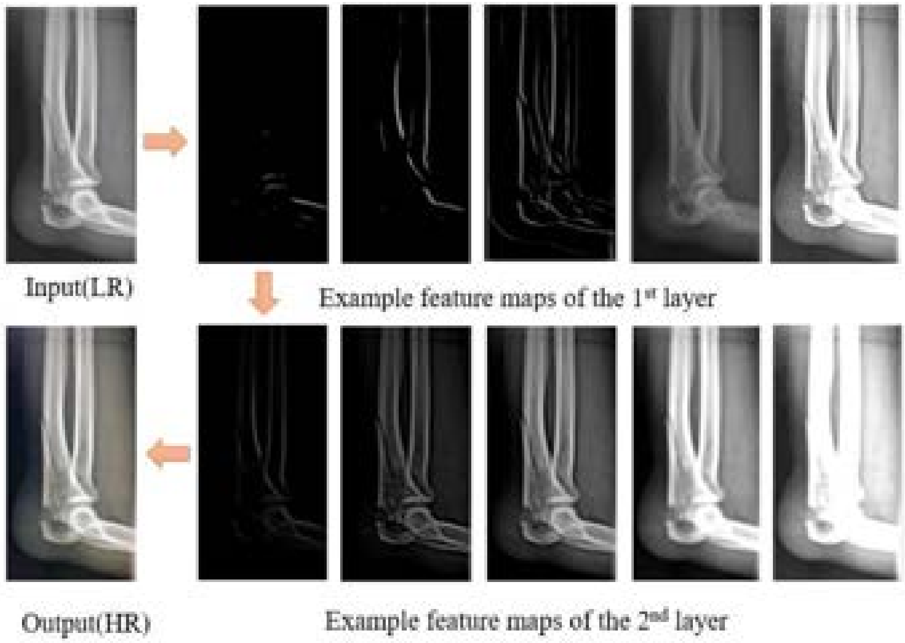

Input(LR)
Example feature maps of the 1st layer
Output(HR)
Example feature maps of the 2nd layer

Figure 4.5: The output of each layer in SRCNN.

Based on Figure 4.5 and its sub-figures, it can be seen that the first convolutional layer mainly extracts the structure details like edges. In addition to edge detection, there are other filters, that have other functions, such as gaussian filter that can eliminate noise. The second layer extracts information about the intensity of each region of the image.

To train our model, we have used the loss function based on the MSE error. By minimising the following loss function, we have optimised weights and bias values of each convolutional layers. We have defined the loss function as below:

$$\mathcal{L}(P) = \frac{1}{n}\sum_{i=1}^{n} \|F(Y_i;p) - X_i\|^2 \tag{4.11}$$

where P is set of parameters containing weights and biases. Y is LR mage and $F(Y,P)$ is the reconstructed HR image, and n is the size of the batch, that we set to 128.

In order to determine the most appropriate architecture for our SRCNN model, we have tested a variety of architectures. As deeper models provide a greater abstraction of features, we first tried deeper models. For the purpose of making it deeper, the first layer involves mostly edge detection and the last layer involves image reconstruction from the HR feature map, so there is no benefit in adding those layers. Nevertheless, in order to achieve a better mapping between the LR feature map and the HR feature map, we added one-on-one convolutional layers similar to the current second layer to change the architecture from 9-1-5 to 9-1-1-5. The kernel size for the new layer is 1×1 with 16 filters. The other layers have been left unchanged.

Furthermore, a wider network can allow us to increase the dimension of our feature map, which could be beneficial for our network. With a wider network and a feature map of higher dimensions, it is possible to extract more left-over features from LR images. As a result of these extra features, HR images can be reconstructed with more detail. On the other hand, it can make our model slower and more likely to become under-trained. As the last layer is used primarily for the reconstruction of images from feature maps, there is no need to make it wider, so we have increased the number of filters in the first and second layers. In the first layer, we have doubled the number of filters from 64 in the previous model to 128, and in the second layer, we have doubled the number of filters from 32 to 64.

Last but not least, the loss function of our model was not decreasing, but the loss over training set was decreasing as a consequence of over-fitting. We have increased numbers of images in our training set.

Our dataset originally contained 60 x-ray images, mostly hand images. After we expanded our dataset, we used 90 different images. processing large images needs a huge computational cost, so we cropped different regions of each image to have 16,000 blocks of images. We performed 10,000,000 iterations for training. For testing, 15 images were used, some containing fractures, to evaluate our model for bones with (Figure 4.7) and without fractures (Figure 4.6) and determine whether it can assist in more accurate diagnosis.

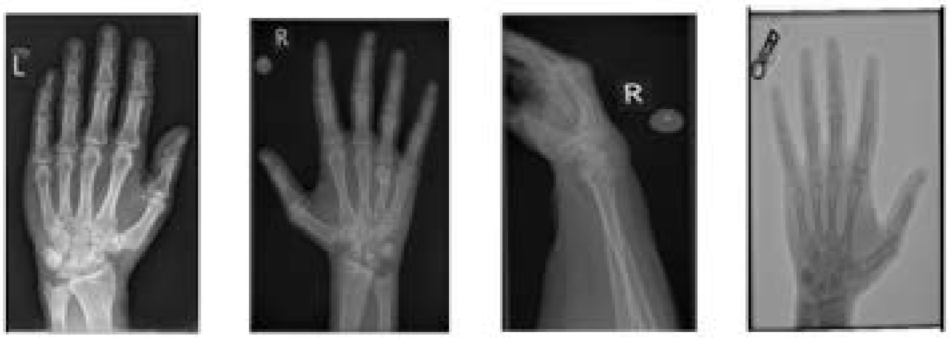


Figure 4.6: X-ray image samples from the dataset without bone fracture.

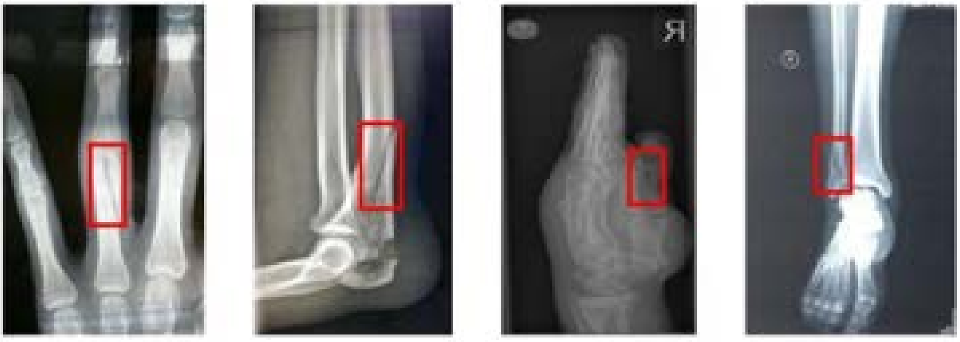


Figure 4.7: Samples from the X-ray image dataset with fracture.

4.4.2 *SRGAN*

In some GAN-based SR applications (SRGAN), similar CNN structures for both generator and discriminator have been used e.g. the one proposed by Ledig et al. [144]. Based on their proposed structures, the generator is a ResNet [53], consisting of several residual blocks, each with two 3×3 convolution layers. The batch normalisation layer (BN) and the parametric ReLU layer (PReLU) are applied after each convolution layer. By skipping the connection operations across the backbone, the gradient vanishing in the back-propagation process is mitigated, making deep networks easier to train. Convolution layers are applied to the final stages of the generator in order to increase the feature dimensions. A discriminator, on the other hand, has eight convolution layers. The number of features increases as the network structure deepens, but the size decreases. Leaky ReLU is chosen as the activation function for the CNN, and sigmoid activation functions for the two fully connected output layers.

In our proposed method, we used a similar SRGAN architecture with a few modifications. For the discriminator, we used a residual network with eight convolutional blocks followed by three fully connected layers at the end. The first block is a convolutional operation with 3×3 kernel size using 64 filters followed by a leaky ReLU activation function. For the rest of the convolutional block, three layers are used. The first layer is convolutional operation with 3×3 kernel size using 64,128,128,256,256,512,512 filters. For the second layer, batch normalisation is used, and the leaky ReLU used as activation function. In the fully connected layers in the last block, a dense layer with 1024 nodes, followed by leaky ReLU activation function and another dense layer with one node used.

In the generator, ten blocks are used. The first block contains a convolution layer using a kernel of 9×9 and 64 filters. From second to sixth layer the same architecture is used. Start with convolutional operation using 3×3 kernel size and 64 filters, following by batch normalisation layer, PReLU activation function, another convolutional layer similar to the first layer, and another batch normalisation layer are used. The seventh block consists of one convolutional layer with 3×3 kernel size using 64 filters. This layer is linear and does not have any activation function. Eighth and ninth block contain one convolutional layer that again has kernel size of 3×3 and it has 64 filters, following by a shuffling layer and PReLU avtivation function. In the last layer one convolutional layer using 9×9 kernel size with 3 filters are used. The number of filters in the last layer is related to the numbers of channels we want to have in the output i.e. HR images. For instance, one channel should be used for greyscale images and 3 for RGB images.

The network is trained using loss function that has defined by comparing between reconstructed image by SR and HR ground truth images.

In this study, we used a set of 90 images that we cropped into 16,000 blocks of images. According to Figure 4.8, our model reached a point where further training was likely to lead to overfitting, so we trained it for approximately 9,000 iterations.

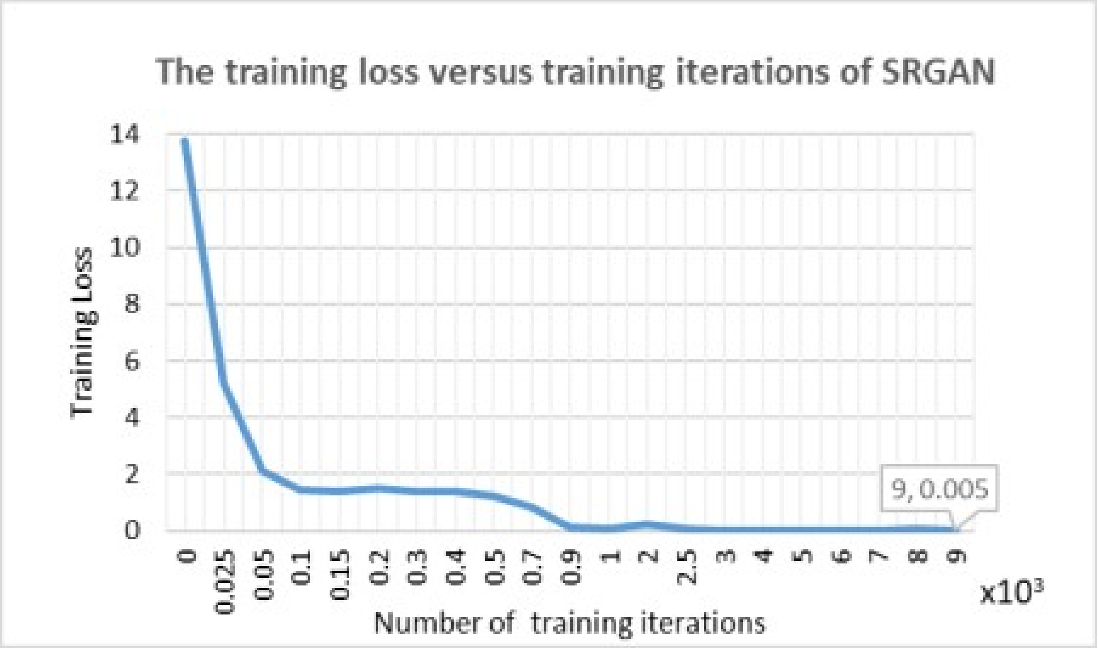


Figure 4.8: Training loss over number of iterations.

4.5 experiments

We process our dataset using SRCNN, bicubic interpolation, and SRGAN. This dataset consists of 60 X-ray images of size 900 × 680 pixels. In order to train our network, these bone images were cropped into 16,000 blocks, that 15 images contained fractures as well. This helped assess whether our model can assist with fracture diagnosis. In our evaluation first we compared different SRCNN architectures to determine were compared the best one, and second, SRCNN, SRGAN, and bicubic interpolation were compared.

Enhancing the visual quality of a digital image can be subjective. The quality of an image can differ from person to person depending on which detail is reconstructed. Therefore, both objective and subjective measures are required to compare the effects of image enhancement algorithms on image quality. We used evaluation indexes such as PSNR and MOS to compare different models. PSNR can be a useful metric for evaluating the image contrast, however, missing details in the reconstructed image can result in a misdiagnosis, which is not measurable by PSNR. Therefore, MOS has also been used to assess the usefulness of reconstructed images in improving diagnosis accuracy.

PEAK-SIGNAL-TO-NOISE-RATIO

In signal processing, PSNR refers to the ratio between the maximum possible value (power) of a signal and the distortion caused by noise. It is common for PSNR to be expressed using logarithmic decibel scales because many signals have large dynamic ranges. PSNR is an objective standard for evaluating the quality of image reconstruction as well. A PSNR value can be used to measure the quality of image processing or SR technique by measuring the difference between the original image and the HR image or any other output image after some kind of processing. PSNR is defined by the following equation:

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{\mathrm{MAX}_f}{\sqrt{\mathrm{MSE}}}\right) \tag{4.12}$$

where MAX_f is the maximum possible value we can have in original data, and the MSE is:

$$\mathrm{MSE} = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|f(i,j) - g(i,j)\|^2 \tag{4.13}$$

where $f(\cdot)$ is original data or in our research it is HR ground truth image and $g(\cdot)$ represent the reconstructed image. MAX is also 255 for images, as we are using 8 bits for their quantisation.

MEAN OPINION SCORE

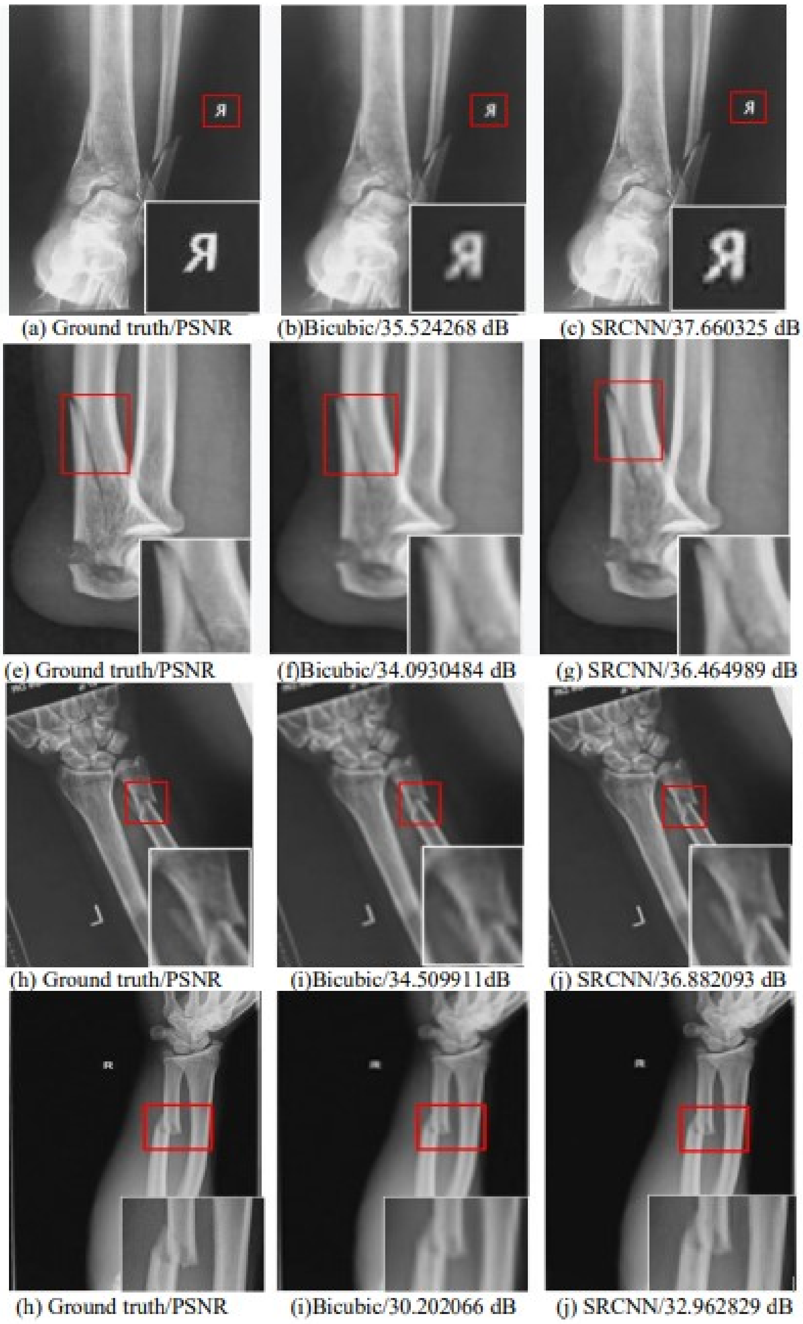
PSNR is an objective evaluation, whereas a mean opinion score (MOS) is a subjective evaluation. An observer evaluates the quality of an image using MOS, which is based on the following equation:

$$\mathrm{MOS} = \frac{\sum_{i=1}^{k} n_i c_i}{\sum_{i=1}^{k} n_i} \tag{4.14}$$

The number of people scoring each output is indicated by n_i, and each score is denoted by c_i. In our study, 34 participants provided MOS, including clinicians from the Hand Unit at Derby Hospital and researchers from the Electrical and Electronic Engineering Department at Imperial College London.

First, a SRCNN architecture with three layers using 9-1-5 filter sizes was used to compare the quality of the reconstructed image with bicubic interpolation. Figure 4.9 illustrates the results.

As previously mentioned, SRCNN has a significant advantage over blind techniques such as bicubic interpolation. Next step, we compared different architectures of SRCNN in order to determine the most effective design. Following are the changes we have made in our comparison:



(a) Ground truth/PSNR (b)Bicubic/35.524268 dB (c) SRCNN/37.660325 dB

(e) Ground truth/PSNR (f)Bicubic/34.0930484 dB (g) SRCNN/36.464989 dB

(h) Ground truth/PSNR (i)Bicubic/34.509911dB (j) SRCNN/36.882093 dB

(h) Ground truth/PSNR (i)Bicubic/30.202066 dB (j) SRCNN/32.962829 dB

Figure 4.9: The comparison between SRCNN model and blind model like bicubic interpolation.

- Change the design from 9-1-5 to 9-1-1-5 by increasing the depth of the network.
- To change the design from 9-1-5 to 9-3-5, we need to use kernels with larger sizes to increase the width of the model.
- Increase the dimensionality of the feature map by using 128 and 64 filters in the first and second layers $(n_1 = 128, n_2 = 64)$ instead of 64 and 32 filters in the

previous model ($n_1 = 64, n_2 = 32$). As a final step, we use a bigger dataset in order to avoid overfitting.

Table 4.1 presents all the comparisons based on PSNR.

Table 4.1: Here is a comparison of different architectures of SRCNN in terms of PSNR and training time.

Index / Network	Average PSNR (dB)	Average Training time/100 iters	Improvement in PSNR (dB)
Initial: 9-1-5	36.19621	1.30612 s	–
Increase depth: 9-1-1-5	36.19652	1.86879 s	0.003
Increase kernel size: 9-3-5	36.33652	2.36472 s	0.14
Increase width: 9-1-5 ($n_1 = 128, n_2 = 64$)	36.25742	2.31047 s	0.06
Increase dataset: 9-1-5 (90 dataset)	36.23652	1.31741 s	0.04

The result could be improved by increasing the depth and width of the network, but the time consumption is twice the time used by the initial model. However, using a larger dataset not only improved the results, but also did not significantly affect the computation time. To compare the SRCNN model with SRGAN in the following section, we used the initial model of SRCNN.

For our subjective assessment a group of 34 people participated in this study, including clinicians from Derby Hospital's Hand Unit and researchers from Imperial College London's Electrical and Electronic Engineering Department. Figure 4.10 shows an overview of the MOS and PSNR results as well as a visual comparison. Table 4.2 provides a comparison and the numerical results. PSNR may increase with more iterations, but MOS may not necessarily follow the same trend, since MOS reflects the sensitivity of the human visual system to changes in images.

Table 4.2: PSNR and MOS based comparison between different methods (SRCNN, SRGAN, Bicubic).

Method	Bicubic	SRCNN	SRGNN
PSNR (dB)	35.01	37.27	37.05
MOS	1.63	2.225	3.27

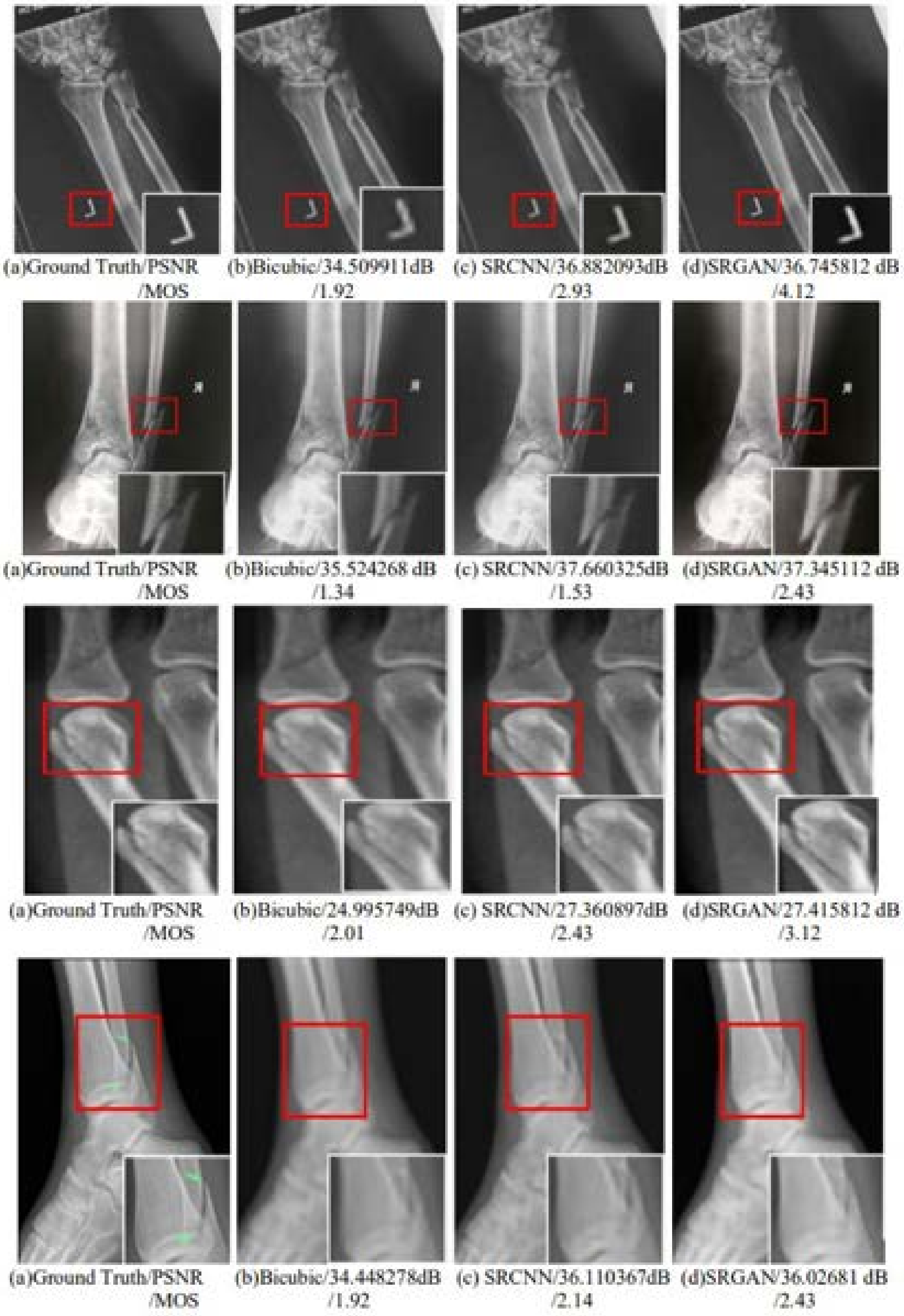

(a)Ground Truth/PSNR /MOS
(b)Bicubic/34.509911dB /1.92
(c) SRCNN/36.882093dB /2.93
(d)SRGAN/36.745812 dB /4.12
(a)Ground Truth/PSNR /MOS
(b)Bicubic/35.524268 dB /1.34
(c) SRCNN/37.660325dB /1.53
(d)SRGAN/37.345112 dB /2.43
(a)Ground Truth/PSNR /MOS
(b)Bicubic/24.995749dB /2.01
(c) SRCNN/27.360897dB /2.43
(d)SRGAN/27.415812 dB /3.12
(a)Ground Truth/PSNR /MOS
(b)Bicubic/34.448278dB /1.92
(c) SRCNN/36.110367dB /2.14
(d)SRGAN/36.02681 dB /2.43

Figure 4.10: Visual comparison between different applied SR techniques.

The PSNR and MOS of SRCNN and SRGAN methods are higher than those of blind methods such as bicubic interpolation. According to the above results, SRGAN cannot improve PSNR when compared to SRCNN. Through the concept of content loss, SRGAN can produce images with greater detail, therefore having higher MOS values, resulting in a more useful clinical assistant system.

As discussed in the previous chapter, we trained and tested our bone segmentation model primarily for HR images. Since x-ray images can vary in noise or resolution based on the device or operator that was used, we conducted another test to see if SRGAN and SRCNN can achieve higher accuracy for SR of LR images.

We have used 95 images as a test set in order to compare these methods. The images were HR, so we used a Gaussian filter to transform them to LR, then we reconstructed them using bicubic interpolation, SRCNN, and SRGAN. LR, SRCNN, SRGAN, and HR image sets were used individually as inputs of the model we developed in Chapter 3 to make the comparison.

THE EFFECT OF SR IN DRLSE-BASED IMAGE SEGMENTATION

Using the object detectors developed in Chapter 3, we found the ROIs in images to start the DRLSE method, but in some cases, ROIs couldn't be identified or were duplicated. We utilised state-of-the-art IoUs for duplicated problems, but hoped to test whether these super-resolution methods could help with missing ROIs.

In order to evaluate the effects of SR techniques on the image segmentation model, we have used 104 images with LR, HR, and reconstructed HR using SRCNN and SRGAN. Table 4.3 summarises the comparison between the number of images with missing or duplicate ROIs produced by the model in Chapter 3 for different datasets including HR, LR, and Reconstructed HR images.

Table 4.3: Summary of the number of images with missing or duplicated ROIs based on different image datasets used as input.

SR methods ROI error	LR	SRCNN	SRGAN	HR (Ground truth)
Duplicated ROI	27	19	17	15
Missed ROI	21	11	6	3
Single ROI	56	74	81	86

Our image segmentation model detects ROI more accurately after using SRGAN and SRCNN algorithms to increase the resolution of the LR image. Therefore, it is reasonable to assume that the SR algorithm can increase the accuracy of ROI detection. This is also one of the evidences that SRGAN can improve the visual effect of X-ray images since deep models are designed to extract features similar to the human visual system.

Figure 4.11 illustrates some samples from each dataset that have been used for the analysis.

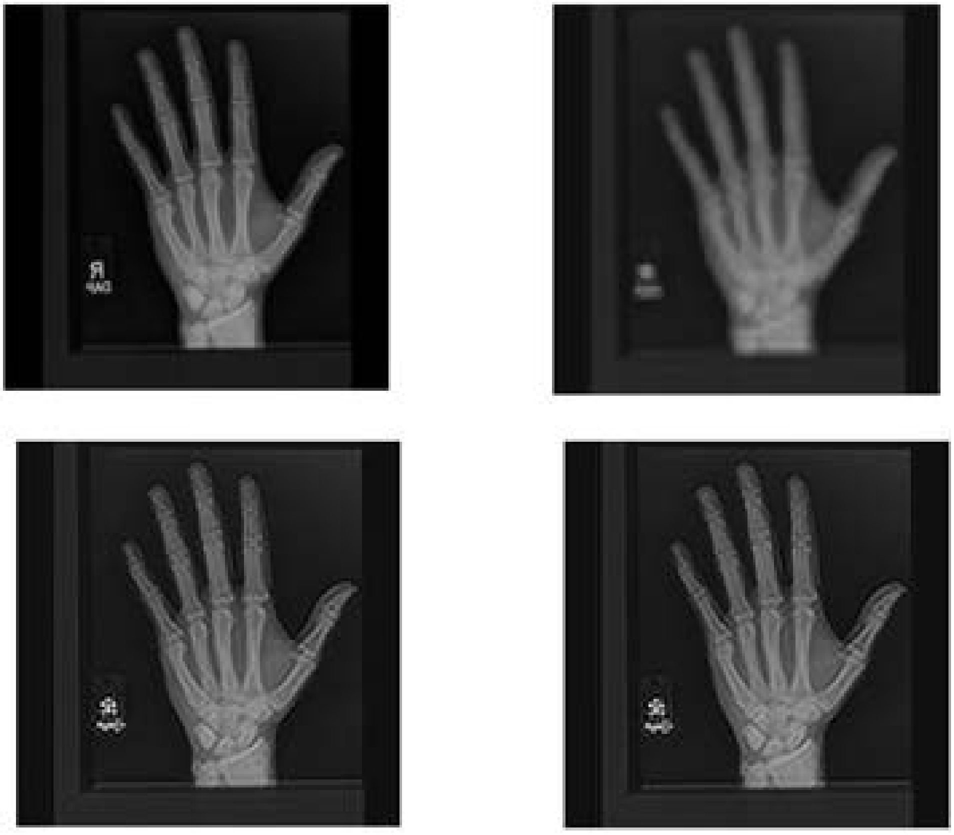


Figure 4.11: The upper left is the HR image, the upper right is the LR image produced by downsampling, the left down produced by SRGAN and the right down by SRCNN.

Additionally, we had a confidence score defined in our model in Chapter 3 for each ROI predicted in the model (Figure 4.12). We have used the average of confidence scores over the test set as well to evaluate different SR techniques. Results are summarised in Table 4.4.

Table 4.4: Comparing the average confidence score for different SR techniques used to preprocess the input.

Model	LR	SRCNN	SRGAN	HR
Avg confidence score	0.72	0.81	0.85	0.93

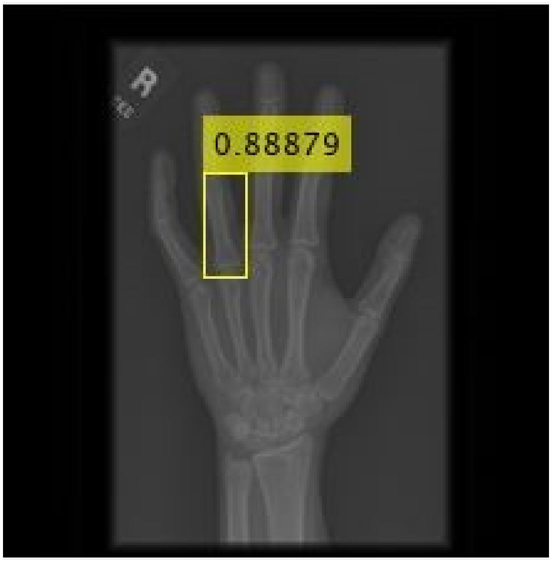


Figure 4.12: A sample of the confidence score in the detected ROI.

As shown in Table 4.4, SRGAN has a higher average confidence score for ROIs. By implementing the super-resolution technique to pre-process the input of our deep model, we can provide a good metric to evaluate our super-resolution technique for fracture diagnosis method, since deep models, like clinicians, are looking for high frequency details.

4.6 CONCLUSION

The purpose of this chapter is to study the application of two DNNs for SR of X-ray images, namely SRCNN and SRGAN. It has been shown that DNNs based on image SR technology could effectively improve the resolution of X-ray images, making it easier to detect micro fractures. The two networks were trained with X-ray images and some adjustments were made to them. In comparison to other DNNs, SRCNN is the first to use CNN structure for SR. In comparison to bicubic interpolation, SRCNN achieves better reconstruction effects when trained with X-ray images. Having higher PSNR does not necessarily mean that the output looks more informative for clinicians or that it leads to a better micro-fracture diagnosis, so we also evaluated our models using MOS. Furthermore, when the network parameters are changed, it is apparent that SRCNN does not perform better when the depth of the network is increased, which contradicts the general conclusion. It is however possible to improve SRCNN performance by increasing the network width, convolution kernel size per layer, and training dataset. Nevertheless, the cost is that it takes longer to train because more parameters must be learned. GAN is introduced into the image SR technology for SRGAN. The performance of SRGAN also improves with the number of training iterations. The PSNR is not significantly different from that of SRCNN, but the improved loss function results in a higher subjective quality (higher MOS) in reconstructed images. Since SRGAN has a deeper and more complex network structure, it takes more time to train than SRCNN. So, there is a trade-off between performance and training time. In spite of its lightweight structure, the deep CNN achieves state-of-the-art restoration quality and fast speed for practical online use.

In the previous chapter, SR techniques were used to pre-process images for the proposed model. The effect of SR techniques on DNN can be a good evaluation metric since DNN work similarly to the human brain, and the result showed that SRGAN results in greater accuracy improvements. Furthermore, SR techniques can help the model be more flexible to the resolution of the input by pre-processing the dataset.

5 USING SALIENCY MAP OF ENTROPY AND INTENSITY TO DETECT BONE FRACTURES

5.1 INTRODUCTION

Many important organs in the human body are protected by bones, including brain, heart, and other constantly functioning organs. There are 206 bones in the human body, which come in different shapes, sizes, and structures [145]. The bones by a strike bones can fracture by applying pressure, kicking, falling down or severe impacts during physical games. Medical imaging tools such as X-rays, Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and ultrasound can be used to detect different types of fractures. In order to diagnose fractures, doctors primarily use X-rays and CT scans, since they provide the fastest and most accurate way to examine damaged bones and joints. To diagnose fractures and where they are located, doctors usually use X-ray images [146]. MRI and X-ray images are our particular interest in this thesis.

Radiologists and clinicians manually examine X-ray images to determine the occurrence and nature of fractures. It is common for radiologists to find fractures in only a small portion of the images they examine. According to the X-rays collected from a local hospital, only about 12% of the femurs are fractured [147]. The result of reviewing many images depicting healthy femurs was that a fatigued radiologist overlooked a fractured femur among the many healthy ones.

A computer vision system can assist doctors in screening X-ray images and flagging suspicious cases that require further investigation. By automating the screening of X-ray images, doctors are able to focus their attention on suspicious cases and make better diagnoses at a faster pace. As a result, it is extremely useful for clinicians to have a computer vision system, and it is now feasible since all radiological images are being performed digitally. X-ray images are now routinely captured using digital X-ray machines.

One of the few studies that have been conducted on fracture diagnosis using X-rays is [148]. In their method, femur fractures can only be detected if the angle between the neck axis and shaft axis changes significantly due to severe fractures. It is also possible to detect fractures using non-visual methods. Ryder et al. examined acoustic pulses [149] in order to determine whether a fracture had occurred. Kaufman et al. employed a neural network model to analyze mechanical vibrations in order to detect fractures [150]. As 30% of women and 13% of men are suffering from osteoporosis [147], most of the research effort has been dedicated to the detection of osteoporosis using first order texture analysis [151, 152] or second order texture analysis [153, 154], but no research has been conducted in order to detect fractures. Among the studies, Caligiuri et al. [155] showed great potential for distinguishing between fractured bones and healthy bones by using fractal dimension.

5.2 TEXTURE ANALYSIS

In image processing, texture analysis (TA) is used to analyze details of structures within images. In an image, there are multiple patterns and subpatterns, for example brightness, colour, size, and density as patterns, and entropy, frequency, linearity, and phase as subpatterns. The texture of an image would be composed of all these patterns and subpatterns [156].

The objective of texture analysis is to extract features from an image so that they may be used as the texture descriptors. Patterns or sub-patterns may be used to create these features. Following is an explanation of how different textural features can be categorised into three groups: first order, second order, and higher order [157]. Diagnostic imaging is one of the most useful tools in medicine. Since it is non-invasive, it is very easy to perform. However, it is difficult to extract information from these images. It is important to note that TA is the most common method of extracting meaningful information [158].

The birth of TA in medical imaging goes back to the 1970s, when the first digitised medical images were used. It is important to note that the human eye can only appreciate images with a low level of complexity. Thus, TA using statistical methods, such as entropy or fractal dimension, presents a method for assessing the complexity of medical images. It is possible that the quantity and quality of the extracted information are imrpoved by TA [159]. TA has been widely used for medical image segmentation mostly related to tissue characterisation[160]. Three methods can be used to derive the textural parameters:

- Transform-based
- Statistical-based
- Model-based

One of the most common methods is the statistical method, in which the features are extracted based on spatial distributions of pixel intensities as well as local features within pixels. This method is divided into three groups: first order, second order, and higher order. The first order (one pixel) uses intensity or grey level frequency to identify global features within a region of interest.

As an example, it can be used to determine textural features by analyzing the mean, skewness and standard deviation of the histogram of the pixel's intensity. In the second order (two pixels), local features are found using spatial grey level dependence (SGLD) or co-occurrence. Based on entropy, energy, and homogeneity within a region of interest, the algorithm attempts to extract similarity or dissimilarity. For higher order (more than two pixels) features, are extracted using neighbourhood grey-tone difference matrices (NGTDM). In NGTDM, the relationship between voxels is extracted within an image plane or between neighbouring planes so that it can be compared with the human visual system. As an example, it can analyse the relationship between the dynamic range of intensity levels and the textural contrast to extract granularity information [161]. The following simulation could provide a better indication of the differences between these orders (Figure 5.1):

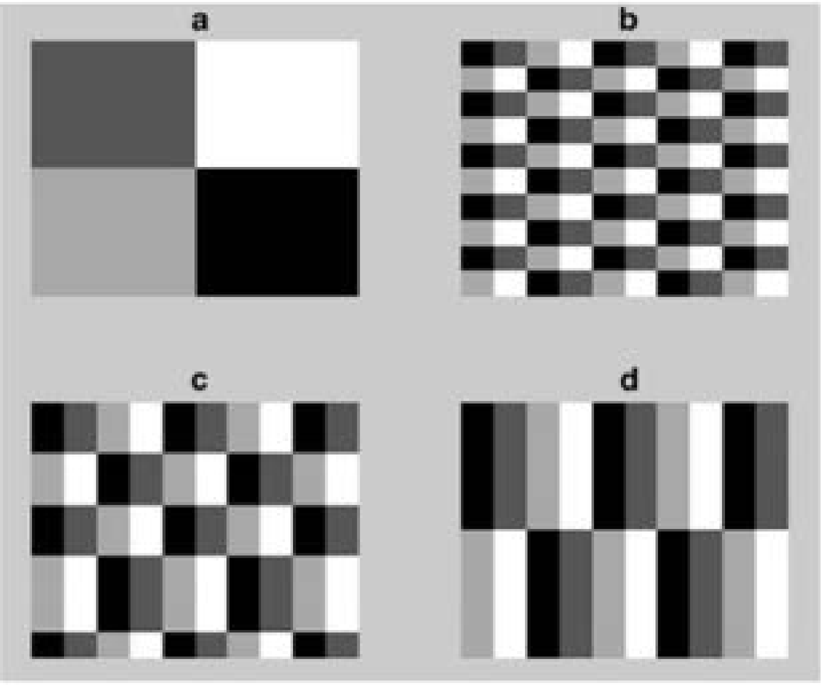


Figure 5.1: All simulations above have the same feature space if first order TA is used. As the second order can distinguish between (a) and the rest of them, but the higher order should be used to distinguish between (b), (c) and (d) [162].

One of the most useful sub-patterns in TA is the entropy of X-ray images. The entropy is used to estimate the randomness of intensity values [163]. Since fractures exhibit a greater degree of texture diversity than other regions of the bone [164]. As a result, fractures can be distinguished from other regions using intensity variation and entropy.

5.3 ENTROPY

The information entropy was first defined by Claude Shannon in 1948. In image processing, entropy can take a variety of forms, for example, it can be used to measure the quality of an image by estimating the level of artefacts [165], or it can be used as a parameter in prediction systems by estimating the uncertainty level [166]. Following is an equation that can be used to calculate entropy H(x):

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5.1}$$

The p_i value is the occurence probability of a pixel value. In order to calculate p_i value, we can use:

$$p_i = \frac{\text{Number of occurences of the intensity level } i}{\text{Number of intensity levels}} \tag{5.2}$$

That in our case the number of intensity levels is 256 as we are using unint 8 images.

5.4 MACHINE LEARNING

ML includes clustering and classification. Both classification and clustering are used for categorising objects into one or more classes based on their characteristics. In classification, predefined labels are assigned to each input instance based on its properties, whereas in clustering, those labels are not present. Due to the lack of labels in our dataset, we primarily used clustering methods for detecting bone fractures. Choosing the appropriate technique to solve a problem depends largely on the data that is available to us. The data in almost all fields can assist immensely in future decision making if suitable techniques are applied to extract hidden but potentially useful information contained within the data [167]. A data analysis system uses machine-learning algorithms to determine the relationships between each individual records. In classification analysis, supervised learning algorithms are used for classification. The term clustering refers to unsupervised algorithms that divide dataset into related groups called clusters. In many fields, such as banking [168] or health care [169], clustering the information is a key technique for data mining. It is more difficult to cluster data when there is no knowledge of the distribution of the data points. As a rule, clustering involves dividing a dataset of m records of data distributed in n-dimensional space into k separated clusters, so that points within a cluster are more similar to one another than to points in other clusters. There are two challenges associated with clustering:

1. To assess the similarity between different elements or individual records, we need to define a similarity measurement.

2. Developing an unsupervised method for identifying clusters of similar elements.

In classical clustering techniques, Euclidean distance is used to determine the similarity of two data points [170]. When the variables in a dataset are numerical, it works well. However, Euclidean distance does not measure the similarity between data points when variables are categorical. In recent years, data mining has been used for collections of categorical data, including data from banks, health sectors or etc. Health care industry data include many categorical datasets including sex, smoking or not smoking and many other numerical and categorical datasets that are impossible to cluster using classical methods. It is difficult to cluster mixed datasets into meaningful groups unless an efficient clustering algorithm is used alongside a good distance measure that can accurately measure the data similarity. Iterative K-mean clustering algorithms group the data into k clusters by minimising a cost function to cluster the data [171]. For k-mean clustering, the following equation will be used:

$$\text{Cost Function} = \sum_{i=1}^{k} \|O_j - C_i\|^2 \tag{5.3}$$

O_j represents the j^{th} data object and C_i represents the centre of the i^{th} cluster. The above equation uses euclidean distance to measure the similarity. Reliably of bone fracture identification in radiographs is difficult because the fractures are uniquely heterogeneous, they can occur in any bone and their appearance depends on the regional anatomy and radiographic projection. Clinicians without a specialisation in musculoskeletal imaging may not be able to localise the fracture accurately [145]. Following the success of DL in the computer vision field, the medical imaging field began to use this methodology for solving its own problems including image classification [172], segmentation [173].

Despite the fact that deep learning techniques have shown significant potential in the field of medical imaging [174–176], they cannot be applied to the images of different parts of the body with the same parameters and hyperparameters. Since DL algorithms are highly abstract, mostly they do not require a change in methodology when moving between fields. Nevertheles, DL systems for fracture detection are limited in use to single bones or phalanges [177, 178]. In order to generalise the system to diagnose bone fracture in the whole body it requires re-training. Although DL algorithms can take advantage of previous success even when solving a different problem using transfer learning [179], re-training is still necessary.

Therefore, we need to label new datasets for new tasks so they can be re-trained. Further, all methods of deep learning require a great deal of tedious labelling work and computational resources. The purpose of this chapter is to avoid the aforementioned problem by using image processing techniques and limiting the use of DNNs in order to reduce the need for labelling and training.

5.5 METHODOLOGY

We attempted to detect bone fractures using a saliency map. The term saliency map has been developed here which is 2D figure of entropy and intensity for each region of the bone. We used windows of size 21 × 21 and calculated entropy and intensity for each region. It is believed that, due to the structure of fracture, the entropy in the area of fracture and around the edge of the bone is distinct from other regions. Alternatively, due to the fact that the fracture is inside the bones, the intensity of those regions varies differently from the edge of the bone (Figure 5.2). In addition, we used k-mean clustering for points, which is the reason why the figure has four colors. Because the values have not been normalised and the range of intensity is greater than entropy, clustering only appears on the intensity axis.

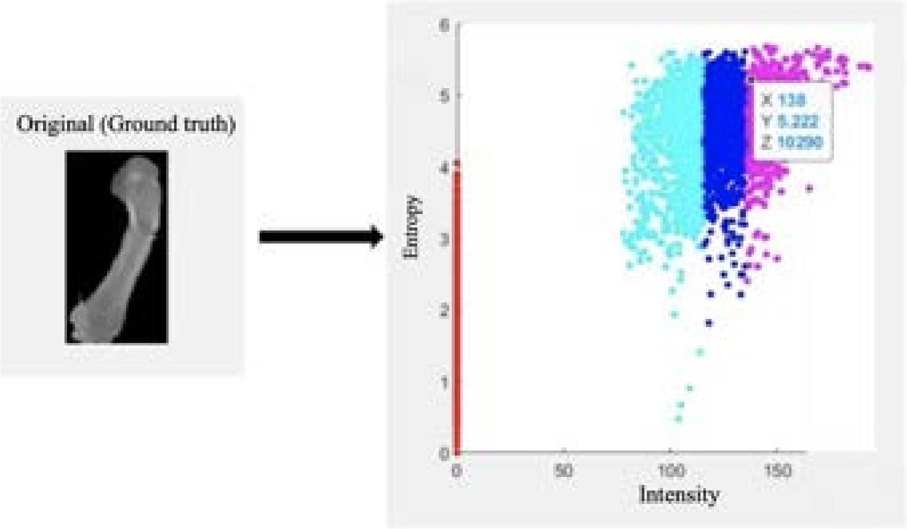


Figure 5.2: An example of mapping one bone to saliency map. Different colors were caused by clustering the points.

For the purpose of mapping the points from the saliency map back to the image, a third dimension containing the voxel number has been defined as the Z-axis. As a result of this, every point in the image can be mapped into the original image as shown in Figure 5.3.

Our second stage involved the use of K-means clustering to determine which points correspond to fractures as a single class of points. Due to the fact that the intensity range was between 0 to 255, while the entropy level was between 0 and 6, we have normalised both of them.

One of the most challenging aspects of using X-ray images is the large amount of salt and pepper noise that can cause a significant change in our saliency map. In order to eliminate the effects of those noises, we have applied a Laplacian of Gaussian (LoG)

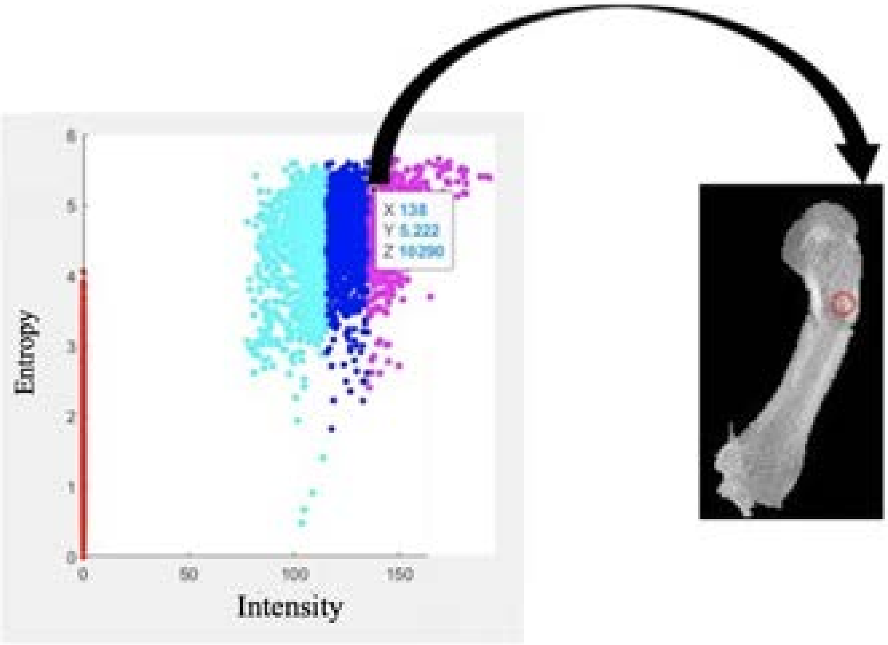


Figure 5.3: An example of mapping back from saliency map to original image.

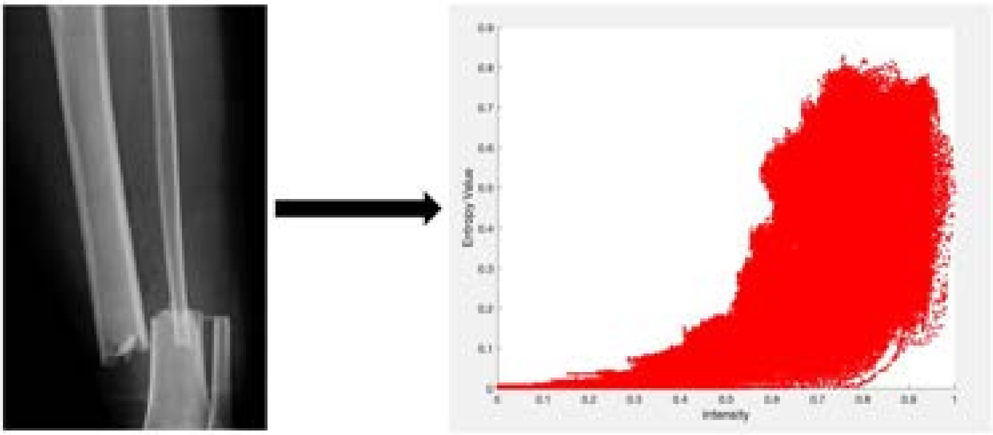


Figure 5.4: An example saliency map for normalised value of intensity and entropy for X-ray image of broken bone.

filter to remove noise from the data. The following kernel has been used to convolve with our images:

$$\mathrm{LoG}(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5.4}$$

where σ is the variance.

Due to this, the filtered image is much smoother than the original image, which may make it easier to locate the fracture region. Figure 5.5 illustrates the difference in visual.

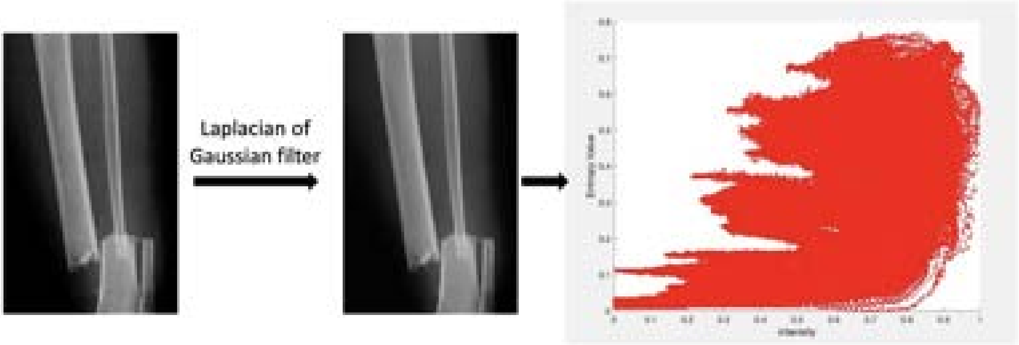


Figure 5.5: Appying the LoG filter and sample of saliency map of the image after the filter.

The next step is the use of k-mean clustering to identify one class that corresponds to fractures. We have used silhouette evaluation to determine the appropriate number of classes.

The silhouette identifies the optimal number of data clusters in order to have more discriminated clusters and a greater margin between them. A silhouette value is a measure of how similar a point is to other points in the same cluster, compared to points in other clusters. An appropriate clustering solution is one with a high silhouette value for most points. The clustering solution might have too many or too few clusters if many points have low or negative silhouette values. The average of the silhouette values for all the images in our dataset is considered to be the silhouette value for clustering the data into that specific number of groups.

As we mentioned earlier, the silhouette value can range from -1 to 1. Having a value of 1 indicates that the clusters are well separated from one another and clearly distinct. A value of 0 indicates that the clusters are indistinguishable, or that there is no significant separation between clusters. In the case of a -1, it means that the data is clustered incorrectly.

$$\text{Silhouette Score} = (b - a)/\max(a, b) \tag{5.5}$$

where a represents the average distance between every two points within a cluster and b represents the average distance between clusters.

To determine the most appropriate number of classes to use k-mean clustering, we have used silhouette evaluation. The average result of the sillouhette evaluation of the ten images has been used to determine the number of clusters which can vary between 2 to 6.

In spite of the fact that entropy and intensity together have never been used before to derive the saliency map and use it to detect fractures, we evaluated the effect of SR techniques, such as SRCNN, on the saliency map with fractures in order to improve the results for the first time. We hypothesised that, as SR techniques can make fractures in X-rays clearer by sharpening the image, the SR technique could affect our saliency map.

It has been discussed in the previous section that SR techniques based on CNN have better performance when it comes to improving PSNR than those based on GANs, so SRCNN has been selected here. It has been determined that SRCNN has had a significant impact on the saliency map that we have developed. In Figure 5.6, we have shown a schematic representation of what our workflow looks like.

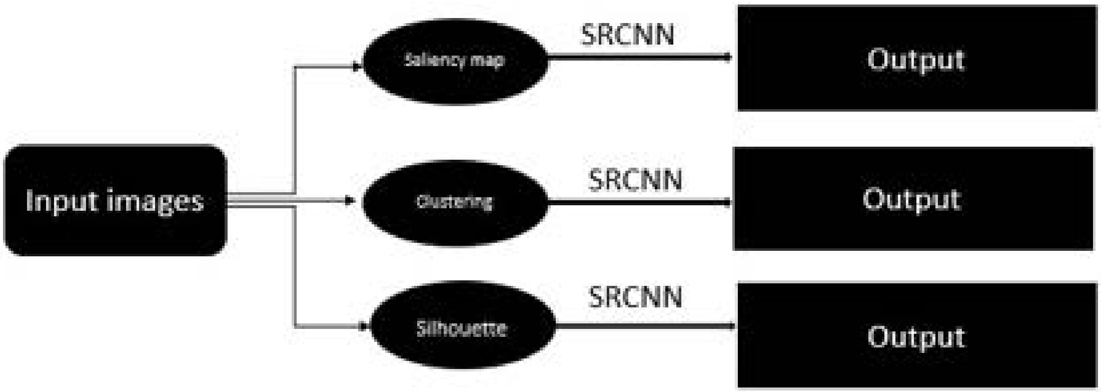


Figure 5.6: Workflow of our research.

For our dataset, we have purchased 15 images from the "https://www.shutterstock.com" website, all of which contain fractured femure X-ray images with varying types of fractures, and the size of the images is approximately 1800 × 2400 pixels.

5.6 EXPERIMENTS

Even though LoG filters are beneficial for reducing the texture noise, they can sometimes mask fractures, as illustrated in Figure 5.7. This is why we did not use LoG prior to drawing the saliency map in order to avoid omitting the fracture.

Silhouette has been utilised in order to determine if a particular number of clusters can be used to detect the bone fracture. We found that the silhouette evaluation result was higher when we clustered our images into four clusters, with each cluster representing four categories: Out of bone, edge of bone, healthy bone texture, and fractured bone. The result of the silhouette evaluation is shown in Figure 5.8.

Our saliency map which is a entropy-intensity feature space has been divided into four groups using k-mean clustering based on saliency, as in Figure 5.9:

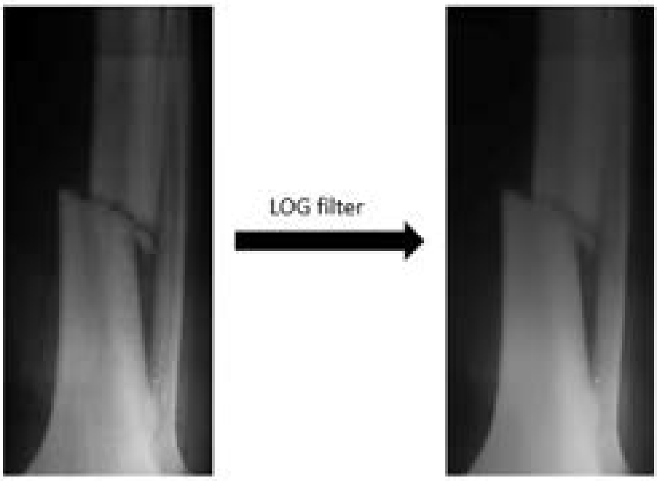


Figure 5.7: Sample of the effect of LoG on the X-ray image with fracture.

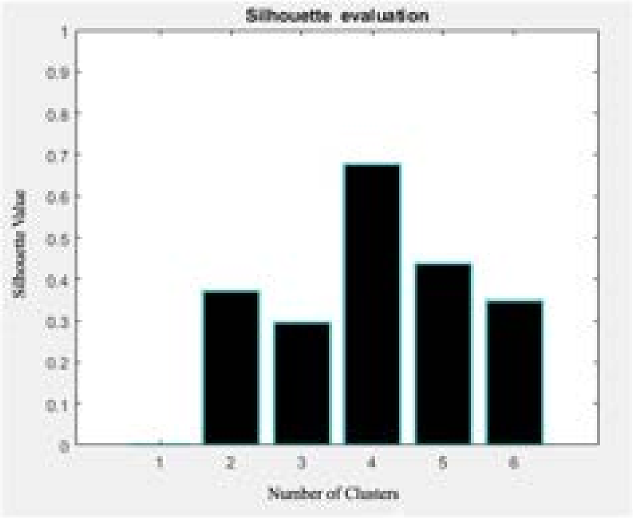


Figure 5.8: Silhouette evaluation result for clustering to 2 to 6 groups.

To determine which cluster corresponds to fracture, we have cropped the fracture region and plotted the saliency map separately. As a result of observing the saliency map of the fracture region, we were able to determine which part of the image we could map back to the original image. As it is shown in Figure 5.10, there are blue points on the saliency map of the original image indicating the saliency map of the fracture region.

To improve the result, we have used SRCNN and we evaluated the effect of super-resolution technique on our saliency map. The use of SR techniques has made the fracture more easily separable because the edge has been sharpened.

As shown in Figure 5.11, the average effect of SRCNN on sillouhete evaluation for 10 images is as follows:

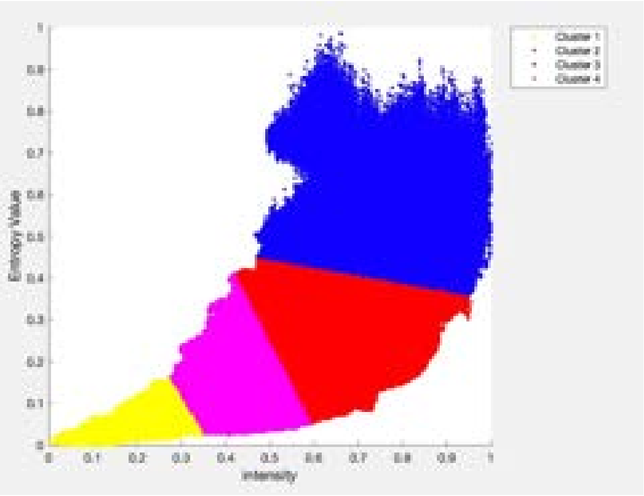


Figure 5.9: K-mean clustering is used to divide points in saliency map into 4 groups.

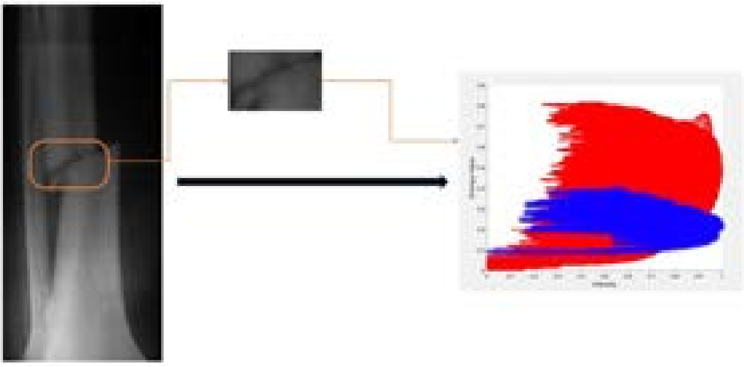


Figure 5.10: The saliency map of the fracture region and the original image.

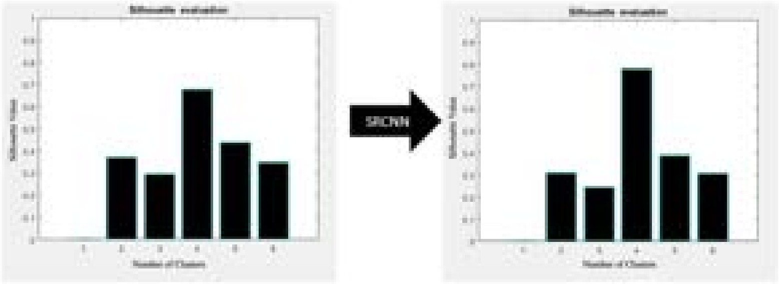


Figure 5.11: The effect of SRCNN on silhouette evaluation.

After the implementation of SRCNN, a higher value in sillouhete evaluation indicates that the clusters are divided with a larger marginal gaps between the clusters. As shown in Figure 5.12, SRCNN has the following effect on the saliency map of fracture regions:

As part of the evaluation process, we have compared accuracy, precision, sensitivity, and specificity, that have been explained in Chapter3. Table 5.1 compares the saliency

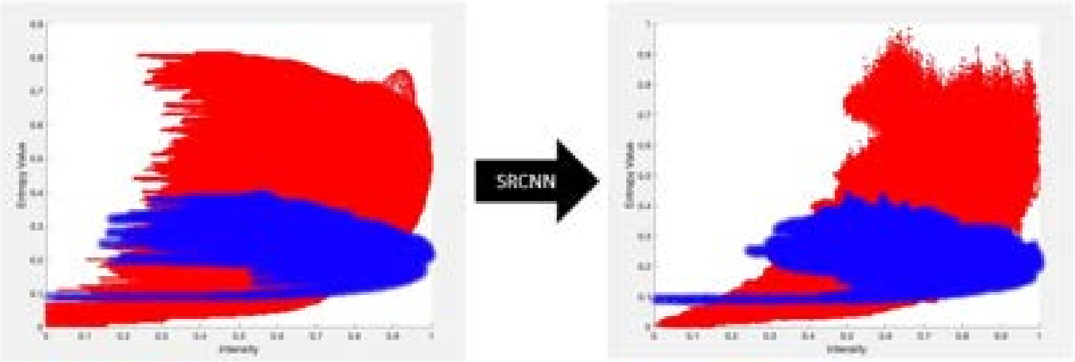


Figure 5.12: SRCNN's effect on the distribution of points corresponding to fracture regions.

map method with using a combination of decision tree (DT) and neural network (NN) [180] as well as using the combination of saliency map and SRCNNs.

Table 5.1: The comparison of our presented model with other methods.

Method	Accuracy	Precision	Sensitivity	Specificity
Saliency map	83.1%	42%	93%	72%
DT+NN [180]	85%	76.9%	100%	70%
Map+SRCNN	90.4%	49%	95%	78%

5.7 CONCLUSION

In terms of precision and sensitivity, the results of our model are not promising in comparison to NNs and decision trees, however, our model achieves better results in terms of accuracy and specificity. An evaluation based on accuracy, precision, sensitivity, and specificity can have a wide range if the size of the images is changed, because the number of TN pixels corresponding to the outside of the bone may vary. Further, our method does not require pixel-by-pixel labelling or extensive training. The use of SRCNN in conjunction with the saliency map can help to make our model more robust when dealing with a wide range of resolutions in X-ray images. It was our objective to demonstrate the capability of image processing in automated fracture diagnosis and demonstrate how super-resolution techniques can be utilised to enhance the capability of image processing.

The saliency map can be used to detect tumours or other abnormalities in medical images, and in conjunction with DNN models to increase accuracy. Saliency map is a powerful image processing technique and can also be utilised as a real-time system because of having low computational cost.

6 X-RAY TO MRI TRANSLATION USING VAE AND CGAN

6.1 INTRODUCTION

In some parts of the body, such as the wrist and ankle, there are many small bones. The shadows cast by these small bones on X-ray images may lead to many incorrect diagnoses. Moreover, X-ray images are not sufficient for diagnosing scaphoid fractures or calcaneal fractures, and the treatment requires close monitoring of the patient for a few weeks. In order to minimise pain and trauma, it is important to diagnose any bone fracture as quickly as possible. After an injury, an MRI can provide an immediate diagnosis if the initial radiographs are inconclusive. Due to the fact that MRIs provide tomographic views of bones, MRIs can assist in eliminating the shadows that result from overlapped bones, thereby reducing the possibility of diagnostic errors [181].

The most common fracture of the carpal bones is scaphoid fracture [182]. Avascular necrosis is caused by diagnostic errors in five to twelve percent of cases [183]. It is estimated that 16% of scaphoid fractures remain invisible on radiographs for the first six weeks of healing [184, 185].

Scaphoid fracture diagnosis from X-rays can be challenging [186]. In the cases of suspected scaphoid fracture, splinting the wrist after 10 – 14 days and taking a follow-up radiographs are recommended. Although the fracture may be visible as a result of resorption around the fracture line [187], the radiographs obtained after two weeks may not be reliable [188].

Calcaneus is the largest tarsal bone and is responsible for supporting the body [189]. A calcaneus fracture accounts for one to two percent of all fractures, and sixty percent of fractures of the tarsal bones [190–193]. Calcaneal fractures can be caused by a variety of factors, including high-energy traumas during falls and low-energy ones during sports. More than 60% of cases are caused by axial loads, such as falling from a height and landing on the feet [194].

A definite diagnosis is required in order to determine the appropriate treatment for calcaneal fractures. In most cases, inflammation is the most prominent symptom of fractures [195]. The ability to diagnose the patient effectively may also reduce the economic burden and speed up the delivery of proper care to the patient. The calcaneus can be diagnosed by radiographic scans and MRI, but the inherent shadows and noise in X-ray images may give an advantage to MRI in some cases.

Recent literature indicates that MRI can provide an early, accurate diagnosis more efficiently than X-rays [196]. A sample of X-ray images from the lateral and PA postero-anterior (PA) scaphoid bones are shown in Figure 6.1.

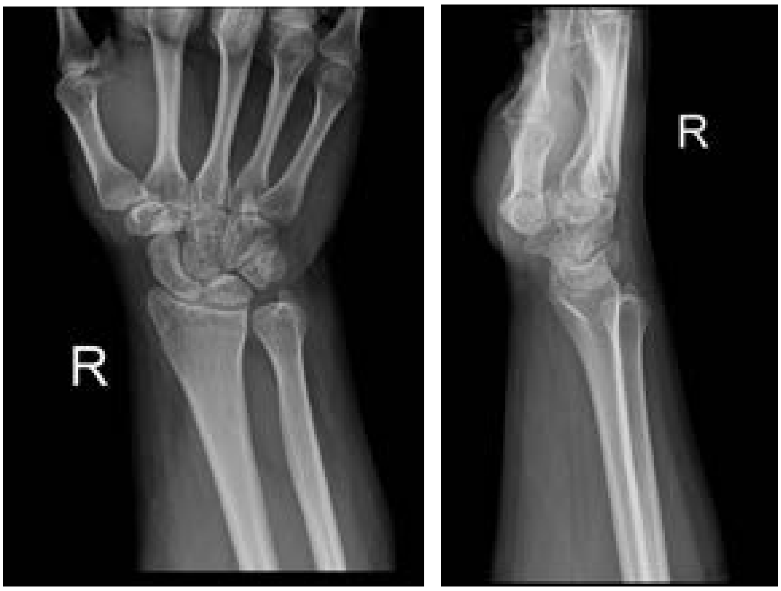


Figure 6.1: Example of a lateral and PA view of an x-ray taken from the scaphoid bones.

It is evident from the wrist image that skin shadows and overlapped bones make it impossible to diagnose a scaphoid fracture accurately. In a similar manner to the scaphoid, the structure of the bones in the ankle can be problematic when trying to diagnose calcaneal fractures. In Figure 6.2, images of the ankle are shown from a lateral and a PA view.

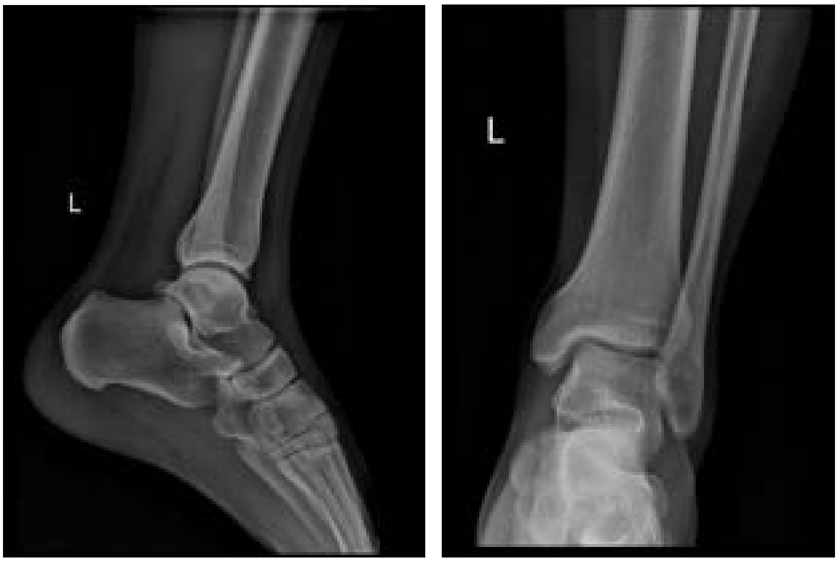


Figure 6.2: Example of a lateral and PA view of an x-ray taken from the calcaneal bones.

MRI is highly effective in diagnosing occult micro-fractures due to its excellent sensitivity, specificity, and reliability. MRI is not only used for the detection of scaphoid and calcaneal fractures, but is also useful in detecting other fractures in areas where the bone structure is complex [197]. An MRI can be used to diagnose these injuries early and prevent further damage from occurring. As a result, there is no need to follow up for several weeks and to conduct extensive reviews. MRI may also be used in some cases to exclude fractures and prevent overtreatment.

MRI should be considered the gold-standard imaging modality to investigate calcaneal or scaphoid fractures. In this way, overtreatment and undertreatment can be prevented by implementing an appropriate management system. In the United States, there are guidelines regarding when and where MRI should be used, but none exist in the United Kingdom [198]. For the management of occult scaphoid fractures, national guidelines should be implemented.

Although MRI offers many advantages over other imaging modalities, it is more expensive and the equipment required to capture MRI is not widely available in all clinics. A state-of-the-art deep neural network was used in this chapter to investigate whether it was possible to reconstruct MRI from X-ray images, which are much cheaper and more prevalent than MRIs. Reconstructed MRIs do not contain skin shadows or shadows caused by overlapping bones, and they do not require additional cost over conventional X-ray images. In other words, the X-ray images are enhanced by the underlying information visible only in MRIs.

The purpose of this chapter is to modify the variational autoencoder (VAE) and generative adversarial network (GAN) and design a new system to reconstruct MRIs based on the matched X-ray images. By using this end-to-end model, we are able to make more accurate and early diagnoses using X-ray images in the event that MRIs are not available, without incurring any additional costs. For these purposes we map the X-ray images to MRI, so once the training is complete, we have a dictionary to generate the corresponding MRI using only the X-ray images.

In order to reconstruct an MRI from an X-ray image, generative models are one of the methods that can be applied.

In non-parametric cases, machine learning may involve estimation of distributions $P(X)$, over some potentially high-dimensional space X, such as images. In each image, there are many pixels, and the generative model's job is to somehow capture the dependencies between those pixels, so nearby pixels can be used to estimate $P(X)$ [199].

Recently, generative models based on deep learning have gained increasing attention due to some impressive advances. Based on large amounts of data, well-designed network architectures, as well as smart training methods, a deep generative model is capable of producing highly quality realistic images, texts, and sounds. We would like to draw attention to two types of deep generative models in particular: generative adversarial networks (GANs) and variational autoencoders (VAEs).

6.2 VARIATIONAL AUTOENCODER

VAE has become one of the most popular approaches for learning how to generate complex probability distributions. VAEs are appealing because they exploit neural networks and can be trained using popular optimisation methods such as stochastic gradient descent. VAEs have already been used to generate a variety of complex data, including handwritten digits [200, 201], face recognition [202, 203], CIFAR images [204], shape models [203], segmentation [205], and prediction [206]. The purpose of this section is to discuss how VAE can be useful to our target and explain the mathematics involved.

An autoencoder consists of two parts: an encoder that reduces dimensionality of the data and a decoder that reproduces the data. The encoder model provides an approximation of the posterior probability of the latent random variables that need to be updated during training, which is used as input to the generative model. In contrast, the generative model aims to learn meaningful representations of the data from the latent space. Based on the Bayes rule, the encoder is approximately the inverse of the generative model.

To put it simply, a VAE is a type of autoencoder that has its encoding distribution regularised during the training process in order to ensure that its latent space has good properties so that we can generate some new data with its help. In addition, the term "variational" refers to the close relationship between regularisation and variational inference in statistics. As a way of gaining a better understanding of VAE, it would be beneficial to explain terms such as regularisation, dimensionality reduction, and variation.

In machine learning, dimension reduction refers to the reduction of the number of features used to describe the data. The need for low-dimensional data arises in a variety of situations, including storing, visualising, and preventing excessive computational load. If we want to reduce the dimension of the dataset, we can select some of the features from the dataset or extract a new set of features with a lower dimension from the existing set of data features. It is possible to reduce dimensionality in many ways, but we focused mostly on using encoders.

The encoders generate new sets of feature representations using the existing sets, and the decoders reverse this process. By reducing dimension, the data is compressed from the initial space to the encoded space, also known as latent space. The decoder, on the other hand, is responsible for decompressing them. Depending on the initial data distribution, the latent space dimension, and other factors, the decompression can be lossy. It means that a part of the information is lost during the encoding process and the decoder cannot recover it. A summary of lossless and lossy compression can be found in Figure 6.3.

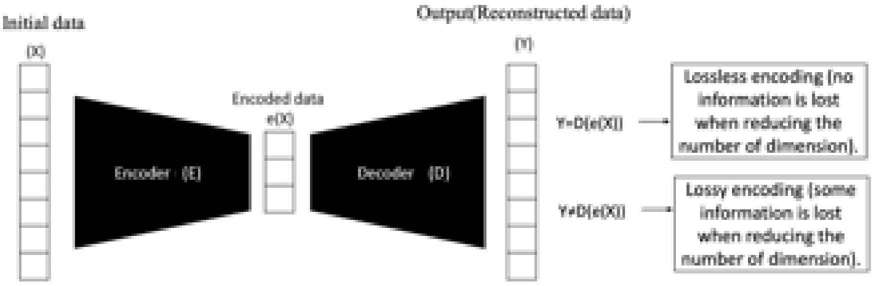

Initial data
(X)
Encoder (E)
Encoded data
e(X)
Decoder (D)
Output(Reconstructed data)
(Y)
Y=D(e(X))
Y≠D(e(X))
Lossless encoding (no information is lost when reducing the number of dimension).
Lossy encoding (some information is lost when reducing the number of dimension).

Figure 6.3: Lossless and lossy encoding schematics and equations.

Dimensionality reduction methods aim to reduce dimensions with the least amount of information loss by finding the best encoders and decoders. Essentially, we want the decoding and encoding models that preserve the most information when encoding, as well as have the least reconstruction error when decoding. The encoder is denoted by E, and the decoder by D. The dimensionality reduction loss function can be written as [207]:

$$(e^*, D^*) = \text{argmin}_\epsilon \big(x, D(e(x))\big) \tag{6.1}$$

where $e(x)$ is encoded data and ϵ is error. Autoencoders use neural networks in their encoder and decoder structures, and are trained using iterative optimisation. In order to update the weights of the networks, we feed the autoencoder with data, calculate the difference between output and input as an error or loss function, and then backpropagate the error through the model for training. As shown in Figure 6.4, the schematic of the autoencoder can be seen and also the symbols that are used in the loss function. Following is a representation of the loss function:

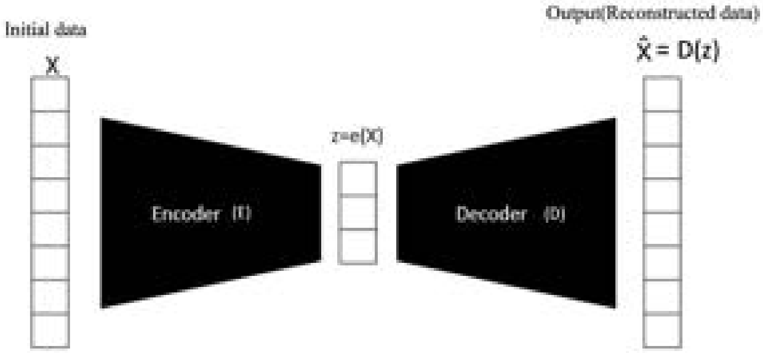


Figure 6.4: Schematic of autoencoder showing the symbols in loss function.

$$\text{loss} = \|x - \hat{x}\|^2 = \|x - d(z)\|^2 = \|x - d\big(e(x)\big)\|^2 \tag{6.2}$$

The bottleneck of an autoencoder will be at the point after the encoder where the latent data space (z) exists.

The limitation of autoencoders is that once they have been trained, they do not produce new content. Thus, generating new data is a challenging task, which has led to the development of state-of-the-art VAEs.

To make use of the decoder as a generative model to generate new content, the latent space of our autoencoder must be regular. A regularity refers to the ability to produce outputs that are consistent and coherent [208]. In order to achieve such regularity, it is possible to regularise the training process explicitly. As a result, VAEs are autoencoders whose training data is regularised to prevent overfitting and to ensure that their latent space possesses properties that enable them to generate new data with approximately similar distributions.

As part of the regularisation of the latent space, rather than creating an encoder that outputs a single value for each latent state attribute, we will create an encoder which describes a probability distribution for each latent state attribute. The following steps should be followed in order to train the VAE:

- First, the input is encoded as a probability distribution over the latent space.
- Second, sample a point from the probability distribution in the latent space.
- Third, the sampled point is decoded and the loss function is used to determine the error.
- At the end of the process, the reconstruction error is backpropagated to the model.

Figure 6.5 illustrates the difference between the workflow of autoencoders and VAEs.

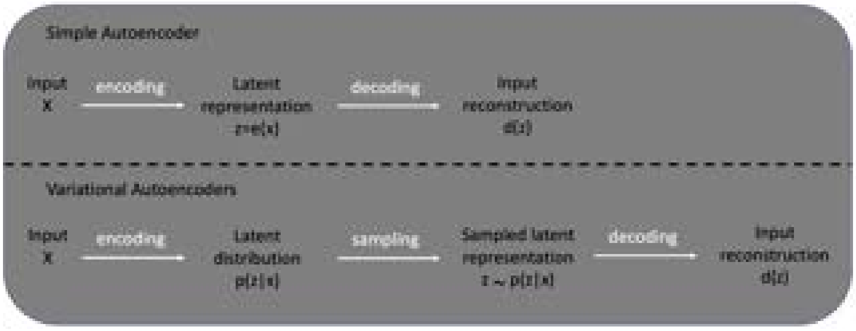

Simple Autoencoder
Input X
encoding
Latent representation z=e(x)
decoding
Input reconstruction d(z)
Variational Autoencoders
Input X
encoding
Latent distribution p(z|x)
sampling
Sampled latent representation z ~ p(z|x)
decoding
Input reconstruction d(z)

Figure 6.5: The diagram of the workflow of autoencoders and VAEs.

The schematic of VAE is represented in Figure 6.6 [209].

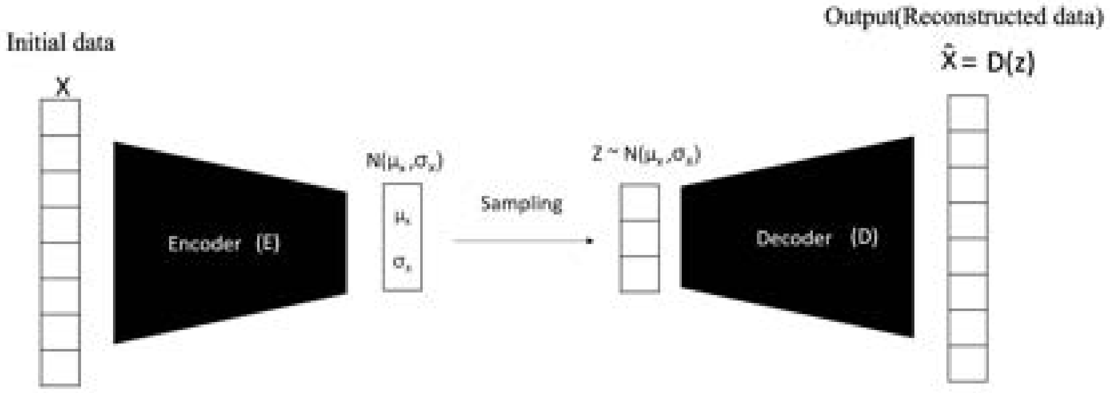


Figure 6.6: VAE schematic.

In the loss function of the VAE, there are two terms: the reconstruction term and the regularisation term. The reconstruction loss is used to determine the difference between the reconstructed data and the original data. The regularisation term makes the latent space regular. The regularisation term is Kulback-Leibler divergence, as a measure of similarity between the generated and target data distributions. The VAE loss function can be represented as:

$$\begin{aligned} \text{loss} &= \|x - \hat{x}\|^2 + KL[N(\mu_x, \sigma_x), N(0,1)] \\ &= \|x - d(z)\|^2 + KL[N(\mu_x, \sigma_x), N(0,1)] \end{aligned} \tag{6.3}$$

The VAE method has been used here to reconstruct MRIs from X-ray images.

6.3 IMAGE TO IMAGE TRANSLATION

Translation of one image to another is a common method of converting one data modality to another. Similarly to translating from Arabic to Spanish, different representations of images, such as RGB images, edge maps, and semantic labels, can be translated to each other. The process of automatic image-to-image translation is similar to that of automatic language translation in that one possible representation of a scene is translated into another possible representation. A wide range of methods are used to predict pixels from pixels in different fields of research, including texture translation in images [210], translating images [211], denoising [212], montaging or annotating [213],and edge detection [214]. Our purpose in this study was not only to predict pixels using a Pix2Pix system [215], but also to generate useful information based on the spatial information provided in the input.

Many image prediction problems have been solved through the use of CNNs. Although CNNs learn to minimise loss functions, which represent the quality of the results, much effort still goes into designing effective losses by fine tuning or regularisation, even though the learning process is automatic. But, we must still inform CNN of what we wish it to minimise. For example, if we ask CNN to minimise the Euclidean distance between predicted and ground truth pixels, we will receive blurry results in the images [216]. The reason for this is that Euclidean distance is minimised by averaging all plausible outcomes, which causes blurring. The challenge of using CNN to generate realistic images with sharp edges remains unsolved [215].

In order to generate realistic data with sharp edges, conditional generative adversarial networks (CGANs) is suggested [217]. For a better understanding of CGANs, it is essential to understand the standard GAN architecture first.

GAN is a generative model, which means that it is capable of creating new content. The goal of generative modelling is to discover patterns in input data and to learn them, so they can create new examples that could have been derived from that dataset. The GAN model is the most common and powerful generative model, consisting of two subnetworks: the Generator (G) and Discriminator (D).

While the G is trying to generate images that cannot be distinguished from ground truth, the D is trying to detect whether the generated images are real or fake, or whether they are reconstructed or ground truth images (Figure 6.7).

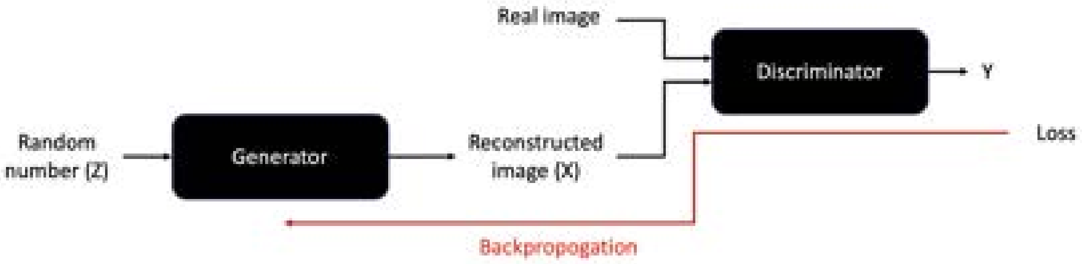


Figure 6.7: Workflow diagram of the GAN.

Random numbers are fed into G to generate data (images). Then, this generated image is used as input to D, which attempts to determine whether the images are real or artificially generated (fake).

If the input comes from the original dataset, then D should classify it as real, while if it comes from G, then it should be categorised as fake.

G initially produces images that contain only white noise without any content. Due to these scrap images, the loss function has a large value. Backpropagating the loss function results in updating the weight until D cannot distinguish real images from fake images. Therefore, G is strongly encouraged to produce plausible translations.

Since we are using D as a classifier, the weights of D should not be changed during the training of G. However, we also need to train D in order to train the GAN. So, a step-by-step training routine is therefore necessary to train the GAN. Following are the steps:

1. In order to produce some images, the latent sample or some noise should be used as input to G.
2. Data from both real and generated sources are used to train D.
3. In order to use D as a classifier, we freeze the weights of D. The latent sample should be used once again as an input to G. The loss function will be calculated based on the output from D to update the weights of G .
4. The loop should be repeated as many times as necessary to optimise both G and D.

The overall GAN loss function working based on equilibrium concept is defined as [64]:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_y\Big[\log D(y)\Big] + \mathbb{E}_{x,z}\Big[\log\big(1 - D(G(x,z))\big)\Big] \tag{6.4}$$

G tries to produce images which are most likely to be real, while D attempts to detect whether they are fake or real. It can be concluded that at the end, there is a stable point, where D has been trained enough to best distinguish between fake and real images while G is still able to fool D. Let's take a closer look at the CGAN now.

GAN is a generative model, and CGAN is a conditional generative model. This makes CGAN suitable for image-to-image translation, in which we implement conditions on an input image and generate an output image corresponding to those conditions. A CGAN has all the same functionalities as a GAN with the exception that a second condition should be fed into the network as an input.

To have output that corresponds to input, the input of the CGAN is also taken into account in the loss function. CGAN's loss function is as follows:

$$\mathcal{L}_{CGAN}(G, D) = \mathbb{E}_{x,y}\Big[\log D(x,y)\Big] + \mathbb{E}_{x,z}\Big[\log\big(1 - D(x, G(x,z))\big)\Big] \tag{6.5}$$

While the discriminator's role remains unchanged in CGAN, G must not only fool D, but also produce an output that is similar to ground truth with the desired condition. The L1 or L2 distance can be used as a loss function to determine whether the output is similar to the ground truth. The L2 distance does not check for blurriness, and since we desire sharp and realistic images, we should utilise the L1 distance:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}\Big[\|y - G(x,z)\|_1\Big]. \tag{6.6}$$

Therefore, the final objective should be [215]:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G) \tag{6.7}$$

Other unsupervised learning approaches for image translation exist, such as cycle GAN. Due to the importance of small details such as fractures in medical imaging, methods which focus predominantly on annotating images, which may result in details being missed, are not suitable. Consequently, we used a state-of-the-art GAN architecture that combines CGANs with pixels-to-pixels GANs. In our method, data is not only used as input, but also added through layers of G and concatenated before being fed into D.

In this chapter we have attempted to map X-ray images to MRIs using image-to-image translation DL methods. In our comparison of different methods, we have found that this translation can reduce the shadows of skin and overlapped bones for a more accurate diagnosis using the reconstructed image, rather than the original X-ray. Thus, instead of capturing time-consuming, expensive MRIs from patients, only X-rays are captured, and MRIs are generated by our mapping networks. Additionally, the reconstructed images would be useful in assisting in the diagnosis of micro fractures or fractures in the scaphoid or calcaneal bones by restoring data that are not visible in the X-ray image to the naked eye, but are clearly visible in the reconstructed image.

6.4 METHODOLOGY

MRI provides more information about the organs, which can help us to make a more accurate diagnosis. In this section, we attempt to convert X-ray images into MRI using VAE and CGAN. In the event that MRI is not available, our mapping dictionary can be used to convert X-ray images to MRIs. Reconstructed MRIs can be used to diagnose fractures that cannot be diagnosed through X-rays.

In this research, we have not only compared our state-of-the-art architecture with current models to map X-rays to MRIs, but also evaluated our system as an end-user product.

Let $X \in \mathbb{R}^{A \times B}$ and $Y \in \mathbb{R}^{C \times D}$ be respectively the X-ray (X) and the corresponding 2-D slice of MRI (Y), where A and B represent the size of X-ray images and C and D defined as the size of MRI. Our mapping model is trying to reconstruct the domain Y based on the domain X. The equation can be defined as:

$$X = f(Y) + \beta$$

where $f(\cdot)$ is mapping function and β is additive noise.

In this study, we have proposed two methods for mapping and compared them with various evaluation metrics, including content loss, MSE, and visual comparison.

We have used 40 pairs of matched MRI and X-ray images for our research. It should be noted that X-ray images and MRIs were taken less than a week apart from one another, from the same region. Our model has been evaluated using ankle images in order to determine how well it performs in complex regions such as the ankle, and also to see whether it removes shadows created by other bones on the calcaneal bone or not.

From the MRIs, we have chosen T1-weighted images. From the tomographic MRI view, three slides are selected whose shapes are most similar to those seen in the X-ray.

These 40 pairs of images have been augmented using different techniques, such as cropping, rotating, flipping, shifting colours, jittering colours, and blurring images, to make a dataset of 475 pairs of images (Figure 6.8). The size of all the images in our dataset is $512 \times 512 \times 3$.

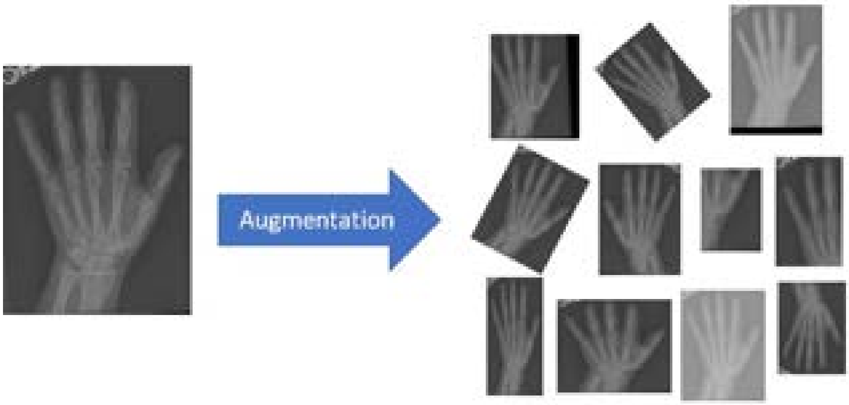


Figure 6.8: Samples of augmented dataset.

6.4.1 *VAE*

In this section, we use VAE to reconstruct MRI images from X-ray images.

A VAE imposes a prior probability distribution over the latent space p(z) to ensure that it is regularised and it has Gaussian distribution, to smooth transition between input and output.

$$z \sim N(0, 1)$$

Thus, rather than mapping data sample X directly to a single point in the latent space, the encoder network E maps X into a Gaussian distribution that can be defined using two different vectors, namely the mean and standard deviation of multivariate Gaussian distributions. Then, the Gaussian-distributed sample is scaled by the predicted mean(μ) and variance(σ) as follows:

$$z = \mu + \epsilon \odot \sigma,$$

To enforce the encoder to map the data sample X into a Gaussian distribution, and also generating new data and avoid collapsing the output into specific type and reconstruct realistic details, the KL divergence will be used as loss function. KL is calculated as follows:

$$\mathcal{L}_{DKL}(G) = D_{KL}(E(X) \parallel p(z)). \tag{6.8}$$

This is carried out by optimising the $p(z)$ of the Gaussian distribution to fit samples of the output of encoder $E(X)$ via maximum likelihood. Accordingly, it can be reparametrised as follows [218]:

$$\mathcal{L}_{DKL} \simeq -\frac{1}{2}\sum_{z=1}^{Z}(1+\log(\sigma_z^2)-\mu_z^2-\sigma_z^2), \tag{6.9}$$

In this case, Z is the dimension of the latent space. The decoder loss is calculated as the distance between the data sample X and the generated data $\hat{X}$. It can be computed using L1 or L2 distances (norms). In this study, the L1 distance is calculated as the decoder loss, in order to avoid having a blurry image as the output. Here is a representation of the L1 equation:

$$\mathcal{L}_1 = \sum |X-\hat{X}|. \tag{6.10}$$

As shown in Figure 6.9, the aforementioned loss function is used to map X-ray images to MRIs:

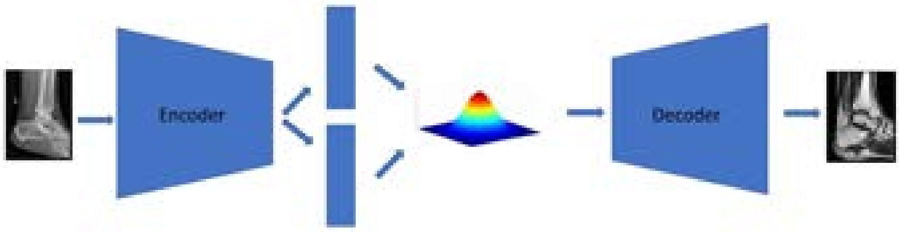


Figure 6.9: Diagram of our VAE workflow.

We have eight layers in our encoder architecture. Normally, 4 to 5 layers are used for encoding. However, the large images sizes allow us to use more layers. The following parameters have been used for each layer (Table 6.1):

Table 6.1: Encoder architecture of the variational auto-encoder.

Name	Type	Numbers of filters	Filter size	Stride	Padding	Activation function	Output
Conv2D_1	Convolutional	n1=32	3×3	2	same	ReLU	[256,256,32]
Conv2D_2	Convolutional	n2=64	3×3	2	Same	ReLU	[128,128,64]
Conv2D_3	Convolutional	n2=128	3×3	2	Same	ReLU	[64, 64, 128]
Conv2D_4	Convolutional	n4=256	3×3	2	Same	ReLU	[32,32,256]
Conv2D_5	Convolutional	n5=512	3×3	2	Same	ReLU	[16,16,512]
Conv2D_6	Convolutional	n6=512	3×3	2	same	ReLU	[8,8,512]
Flattening	Reshaping					ReLU	[3, 27,68]
Dense_1	Fully connected	n8=256				ReLU	[256]
Z_mean	Fully Connected_1	n9=256					[256]
Z_log_var	Fully Connected_2						[256]

After 6 convolutional layers, the output is flattened to be fed into two layers fully connected (FC) network. The first layer is a dense layer with 256 nodes followed by the ReLU activation function. But, the second layer, consists of two parallel dense layers with 256 nodes for each of them. One of them is responsible for calculating the mean of distribution in the latent space, and the other one calculates the variance of the distribution.

This diagram illustrates the workflow of our encoder:

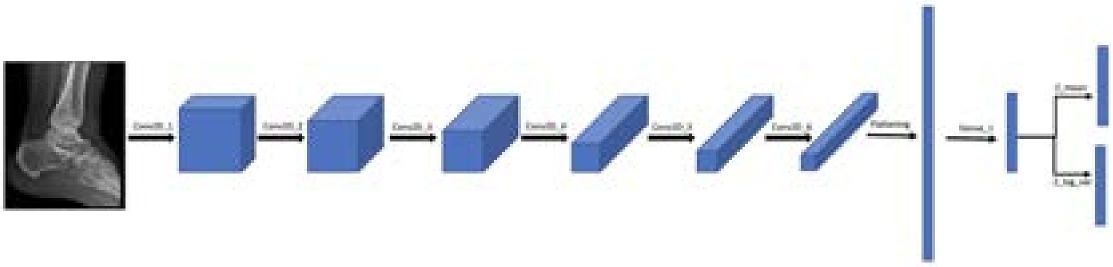


Figure 6.10: Encoder architecture in VAE.

For the bottleneck, we have used 256 as the latent dimension. For the decoder, we have used eight layers. The first layer is a fully connected network with $8 \times 8 \times 512$ nodes. The size of this FC layer is big, because it should have the same size as the output of the layer conv2D_6 after reshaping. Following this, seven layers of transpose convolutional layers are employed. The size of each layer is as follows (Table 6.2):

Table 6.2: Decoder architecture of the variational auto-encoder.

Name	Type	Numbers of filters	Filter size	Stride	Padding	Activation function	Output
Dense_1	Fully connected	32768				ReLU	[32768]
Flattening	Reshaping					ReLU	[8,8,512]
Conv2DT_1	Conv2D_Tanspose	n1=512	3×3	2	same	ReLU	[16,16,512]
Conv2DT_2	Conv2D_Tanspose	n2=512	3×3	2	Same	ReLU	[32,32,512]
Conv2DT_3	Conv2D_Tanspose	n3=256	3×3	2	Same	ReLU	[64,64,256]
Conv2DT_4	Conv2D_Tanspose	n4=128	3×3	2	Same	ReLU	[128,128,128]
Conv2DT_5	Conv2D_Tanspose	n5=64	3×3	2	Same	ReLU	[256,256,64]
Conv2DT_6	Conv2D_Tanspose	n6=32	3×3	2	Same	ReLU	[512,512,32]
Conv2DT_7	Conv2D_Tanspose	n7=3	3×3	1	Same	tanh	[512,512,3]

As our desired output image is RGB, the final layer has three channels. Figure 6.10 shows the architecture used for the decoder:

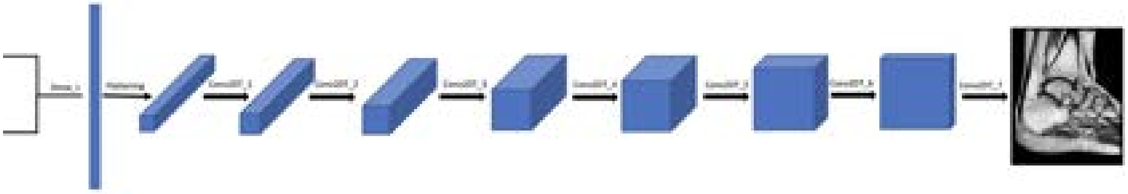


Figure 6.11: Decoder architecture.

We have further improved our model by defining new loss functions. In addition to calculating the loss function between the ground truth MRI and the reconstructed MRI, we have also included slides before and after the chosen MRI. With this new loss function, we have attempted to reconstruct more details in the output image. Thus, our model not only eliminates the shadows cast by skin and other overlapping bones, but also reconstructs detail from three slices, allowing clinicians to see more details. Although this model generates only one image, it contains useful information from three MRI slices. The L1 loss function can be defined as:

$$L_1 = \left(|X_{n-1} - \hat{X}| + |X_n - \hat{X}| + |X_{n+1} - \hat{X}|\right) \div 3 = \frac{1}{3} \sum_{\alpha=-1}^{1} |X_{n+\alpha} - \hat{X}| \tag{6.11}$$

Here, n is the slice number of the chosen MRI, X represents the MRI ground truth images, and $\hat{X}$ represents the constructed image. Due to the fact that there are three layers involved, we have divided it by 3.

In order to have the loss function, we have used KL loss function and L_1 loss function alongside each other. In our cases, the detail in the reconstructed image is important for us, that is why we have used two coefficients for each of them to increase the importance of L_1 loss function. The loss function can be represented as:

$$\text{Loss} = \gamma L_1 + \beta L_{KL} \tag{6.12}$$

where β and γ are the coefficients of the loss function. In this case, we have used 10 for γ and 0.01 for β.

Using MSE and visual evaluation, we compared our new model with the model with the same architecture but with the old loss function. Even though our new loss function may improve the result, this model directly backpropagates the difference between the reconstructed MRI and ground truth MRI to train the VAE-based model. In order to achieve a better result, we have added D to our model in order to have an architecture similar to CGAN and evaluate that model as well.

6.4.2 *GAN*

In GAN, G maps a noise vector into latent space, and then it uses a prior probability distribution p(z) to reconstruct the data. D predicts a binary class of real or fake data samples based on both real and generated data samples. In order to train G and D, we use an adversarial loss function [217, 219] and the binary cross entropy as the loss function from the original GAN as in equation 6.4 [68]. In our method, instead of binary cross entropy, the logistic loss is employed here as an adversarial loss.

Furthermore, in our research instead of feeding the noise vector into the GAN, X-ray images have been used as inputs to reconstruct the MRI. Our state-of-the-art GAN uses VAE as G and adds D to determine whether the reconstructed data is real or fake. The principle of our method is derived from the image-to-image translation methods [215, 220] and the Speech Enhancement GAN [221], which are based on CGANs. Our research focuses on both high frequency details and low frequency details. As a result of using only CGAN, some detail could be lost. For example, we would like to implement an image-to-image translation to map a zebra to a horse, and as long as the output looks real for us, that is sufficient for us, and we do not care if the horse's face has a scar that does not appear in the zebra image. However, in medical images, these scars or fractures include significant details for accurate diagnosis and cannot be missed. To ensure that these details are not missed, we have used conditional (de)normalisation in decoder design. So, in our research we have used X-ray images instead of noise as input for the G to produce MRI instead of synthesised images. To reconstruct all the details, including important information, our network must be able to reconstruct all the small details. In addition to including all the high frequency details in the reconstructed MRI, D should not be able to distinguish between the reconstructed MRI and the original MRI. In this case, we need to generate a synthesising MRI based on the X-ray condition.

The conditional image synthesis process can be implemented in many different ways with different types of input data. As an example, class-conditional models [222, 223] learn how to synthesise images according to their category labels. Image-to-image translation is another widely used method based on CGANs [224, 225], where both input and output are images. In our model, we used conditional normalisation not to miss any detail. So, we have utilised spatially-adaptive (de)normalisation (SPADE) [226].

SPADE is a type of conditional normalisation. In contrast to standard normalisation methods, conditional (de)normalisation requires external data in image form or contain some extra information by re-feeding X-ray images as extra information. An illustration of the SPADE block can be found in Figure 6.12.

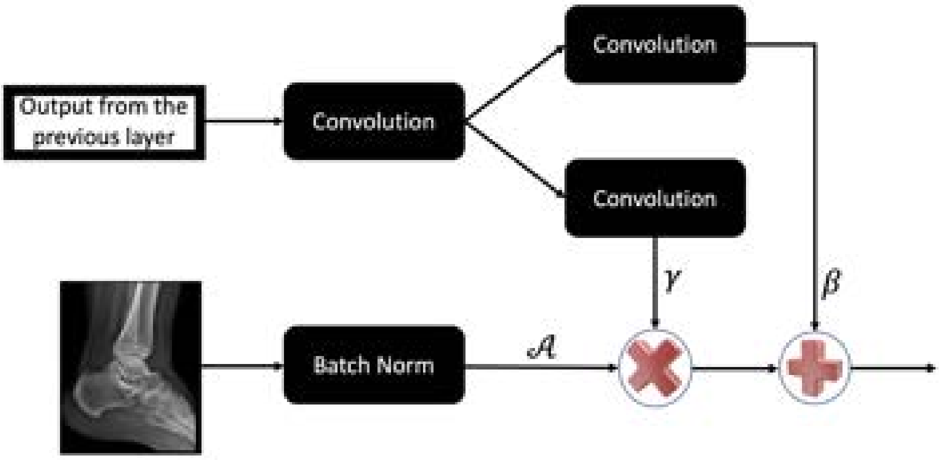


Figure 6.12: The schematic of SPADE block.

In the same way as batch normalisation [217], SPADE normalises activation channel-by-channel. So, each channel is multiplied by γ and added to β element-wise that can be represented as follows:

$$\text{Output} = \gamma \times A + \beta$$

where A is the output of the previous layer after channel-by-channel batch normalisation, and γ and β are the learned modulation parameters of the normalisation layer from the external added data.

Using SPADE, we are able to preserve spatial information better than common normalisation methods in reconstructed images, which helps to prevent the loss of small details such as fractures. Almost all state-of-the-art conditional image synthesis models using CGAN require normalisation blocks, such as the InstanceNorm [227], but they tend to eliminate some of the spatial information. In contrast, the input image in the SPADE generator is fed through spatially adaptive modulation without normalisation. Only activations from the previous layer are normalised. Hence, the SPADE generator can better preserve spatial information.

As the SPADE block is susceptible to gradient vanishing, they are used in the ResNet approach, called SPADE ResNet, shown in Figure 6.13. A SPADE ResNet consists of two SAPDE blocks followed by hyperbolic tangent activation (Tanh) and convolutional layers [226]. In the Resnet SPADE block, there is also a skip layer.

In order to utilise SPADE, which requires additional data for regularisation, we have fed the X-ray (input) into SPADE ResNet in the decoder. The architecture of our model can be seen in Figure 6.14.

Figure 6.15 shows G architecture. So we have used latent space from the output of the encoder to feed into the decoder and we have also used Resnet SPADE as normalisation and skip connection to feed X-ray images directly to decoder. CGAN architecture is mostly using Upsampling and downsampling layer in the encoder and decoder for

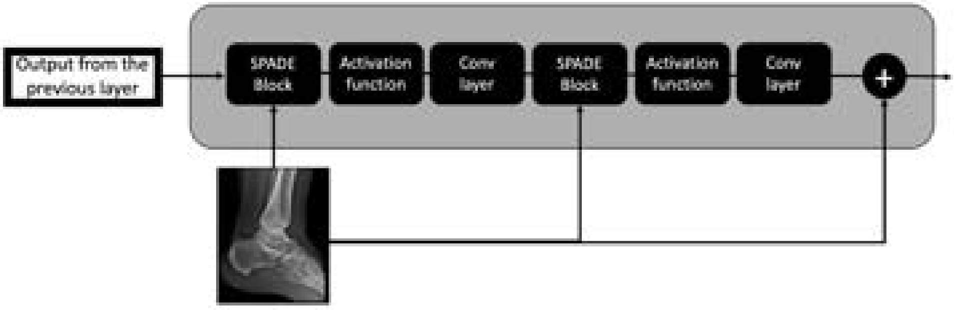


Figure 6.13: SPADE Resnet architecture.

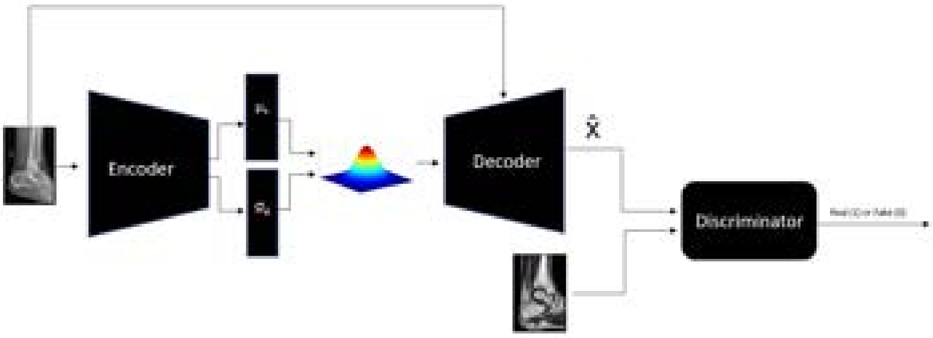


Figure 6.14: Architecture of the proposed X-ray-to-MRI translation model.

some layers, but in our case, as we want to generate the exact MRI corresponding to the input X-ray. So, we have changed upsamling layers with transpose convolutional layer and convolutional layer as downsampling to have more learnable parameters. In order to add SPADE resnet block and transpose convolutional layer is used as following architecture is used (Figure 6.15).

The following filter sizes and parameters are used for the encoder (Table 6.3).

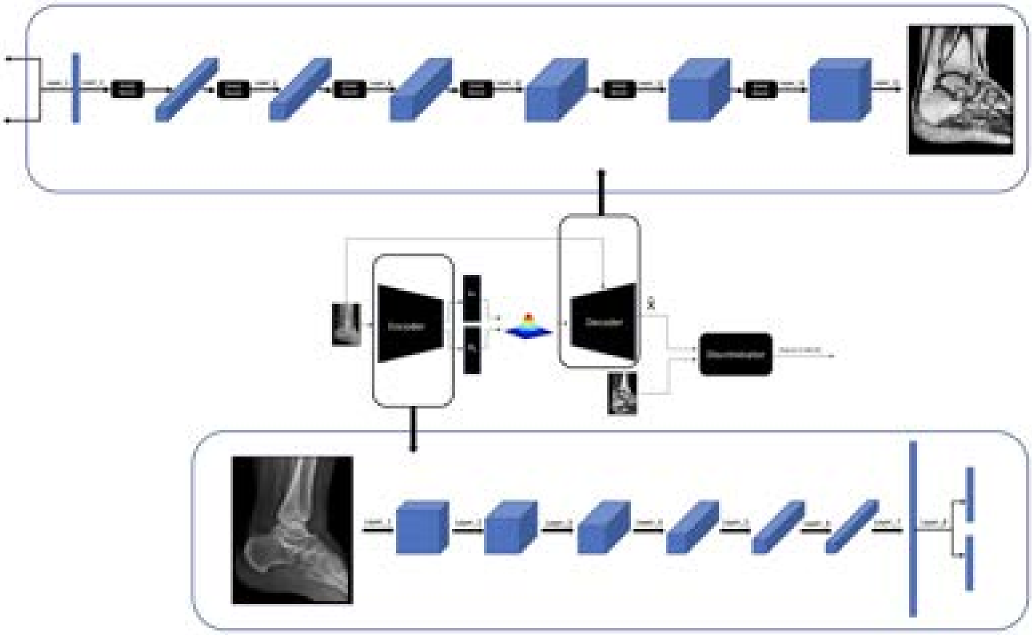


Figure 6.15: The entire design for our generative Network.

Table 6.3: Encoder architecture in the state of art GAN design.

Name	Type	Numbers of filters	Filter size	Stride	Padding	Activation function	Output
Layer_1	Conv2D	n1=16	3×3	2	same	Leaky ReLU (0.2)	[256,256,16]
Layer_2	Conv2D	n2=32	3×3	2	Same	Leaky ReLU (0.2)	[128,128,32]
Layer_3	Conv2D	n3=64	3×3	2	Same	Leaky ReLU (0.2)	[64,64,64]
Layer_4	Conv2D	n4=128	3×3	2	Same	Leaky ReLU (0.2)	[32,32,128]
Layer_5	Conv2D	n5=256	3×3	2	Same	Leaky ReLU (0.2)	[16,16,256]
Layer_6	Conv2D	n6=256	3×3	2	Same	Leaky ReLU (0.2)	[8,8,256]
Layer_7	Flattening						16384
Layer_8	Fully Connected_1 Fully Connected_2	256 Nodes					256

In the last fully connected layer, there are two parallel layers that each have 256 nodes for calculating the mean and variance of the prior distribution. Since 256 dimensions are used in the latent space, it is better to use 256 nodes in the last layer of G.

The decoder uses the filter sizes and parameters in Table 6.4. For the reconstruction of the MRI, we have used transpose convolutional layers rather than upsampling layers. In contrast to an upsampling layer, the transpose convolutional layer has learnable parameters, whereas upsampling does not.

Table 6.4: Decoder architecture in the state of art GAN design.

Name	Type	Numbers of filters	Filter size	Stride	Padding	Activation function	Output
Layer_1	Fully Connected	nodes=512					512
Layer_2	Reshape						
Layer_3	Spade_ResBlock	n3=512	3×3		Same	Tanh	
Layer_4	Conv2D_Tanspose	n2=512	3×3	2	Same	Leaky ReLU (0.2)	[8,8,512]
Layer_5	Spade_ResBlock	n3=512	3×3		Same	Tanh	
Layer_6	Conv2D_Tanspose	n2=512		2	Same	Leaky ReLU (0.2)	[16,16,512]
Layer_7	Spade_ResBlock	n3=256	3×3		Same	Tanh	
Layer_8	Conv2D_Tanspose	n2=256	3×3	2	Same	Leaky ReLU (0.2)	[32,32,256]
Layer_9	Spade_ResBlock	n3=128	3×3		Same	Tanh	
Layer_10	Conv2D_Tanspose	n2=128	3×3	2	Same	Leaky ReLU (0.2)	[64,64,128]
Layer_11	Spade_ResBlock	n3=64	3×3		Same	Tanh	
Layer_12	Conv2D_Tanspose	n2=64	3×3	2	Same	Leaky ReLU (0.2)	[128,128,64]
Layer_13	Spade_ResBlock	n3=32	3×3		Same	Tanh	
Layer_14	Conv2D_Tanspose	n2=32	3×3	2	Same	Tanh	[256,256,32]
Layer_15	Conv2D	n2=3	4×4	2	Same	Tanh	[512,512,3]

In our study the design of D is based on a Markovian discriminator [228]. D classifies the output patch-by-patch. In other words, we divide the images into patches, and D determines whether each patch is fake or real. Instead of having 0s or 1s as outputs of D, we have a matrix of 0s (fakes) and 1s (reals) that corresponds to patches. The Markovian discriminator is more suitable for image or image translation since it can check each region of the image.
The descriminator loss function is defined as [68]:

$$\mathcal{L}_D(G, D) = -\mathbb{E}_Y\Big[\min(0, -1 + \mathcal{D}(Y))]-\mathbb{E}_{x,z}\Big[\min(0, -1 - \mathcal{D}(\mathcal{G}(X, z)))\Big] \qquad (6.13)$$

The easiest method of incorporating the condition and small details into the network is to combine the input image, in our case the X-ray image, with the latent space vector or the output of G. As shown in Figure 6.16, this method has been intrduced used in the first paper on CGAN [205].

In D, we have concatenated our input (X-ray) with the MRI and reconstructed the MRI in order to combine the information from the X-ray and the MRI while D attempts to differentiate between real and fake.

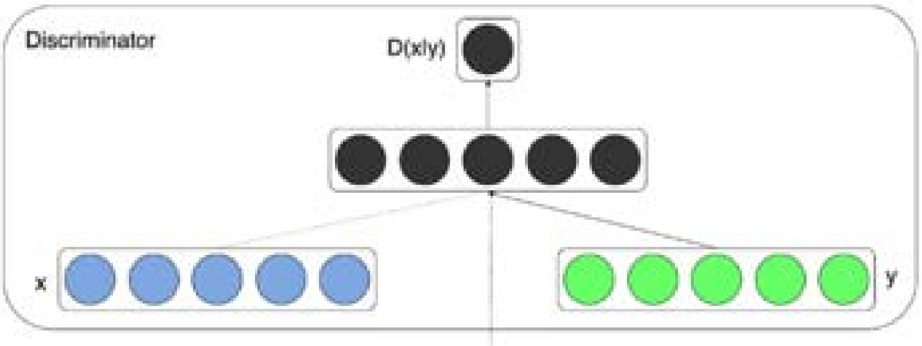


Figure 6.16: The concatenation process in the discriminator network.

D learns to distinguish real patches from inappropriately fake patches. Our D network is made up of the sequence of convolutional layers to distinguish reconstructed MRI from real MRI. D architecture can be seen in Figure 6.17:

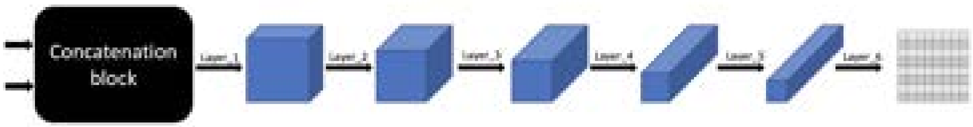


Figure 6.17: Discriminator architecture.

Table 6.3 shows the filter sizes and parameters used for D in Figure 6.17.

Table 6.5: Parameters of the discriminator.

Name	Type	Numbers of filters	Filter size	Stride	Padding	Activation function	Output
Layer_1	Conv2D	n1=32	4×4	2	same	Leaky ReLU (0.2)	[256,256,32]
Layer_2	Conv2D	n2=64	4×4	2	Same	Leaky ReLU (0.2)	[128,128,64]
Layer_3	Conv2D	n3=128	4×4	2	Same	Leaky ReLU (0.2)	[64,64,128]
Layer_4	Conv2D	n4=256	4×4	2	Same	Leaky ReLU (0.2)	[32,32,256]
Layer_5	Conv2D	n5=512	4×4	2	Same	Leaky ReLU (0.2)	[16,16,512]
Layer_6	Conv2D	n6=1	4×4	2	Same	Leaky ReLU (0.2)	[8,8,1]

As it is mentioned before, D and G are trained using the adversarial loss approach. D loss used in our model is in equation 6.13 and the G loss is:

$$\mathcal{L}_{\mathcal{G}_h} = \quad -\mathbb{E}_{x,z}\Big[\mathcal{D}(\mathcal{G}(X,z))\Big]. \tag{6.14}$$

Aside from the G adversarial loss $\mathcal{L}_{\mathcal{G}_h}$, two other loss functions $-$KL divergence loss, L1 loss are also combined and applied to train G:

$$\mathcal{L}_{\mathcal{G}} = \mathcal{L}_{\mathcal{G}_h} + \lambda_1 LKL + \lambda_2 L_1 \tag{6.15}$$

where λ_1 and λ_2 are the penalty terms.

For the purpose of training our model, we have used 40 pairs of ankle X-ray images and MRIs taken over a period of less than two weeks. We have augmented the dataset to have 450 pairs of images. As part of the training process, we used a batch size of four to train our model for 110 epochs over 11000 iterations.

6.5 EXPERIMENTS

The goal is to create a model that can accurately translate X-ray image to MRI without losing any information (Figure 6.18).

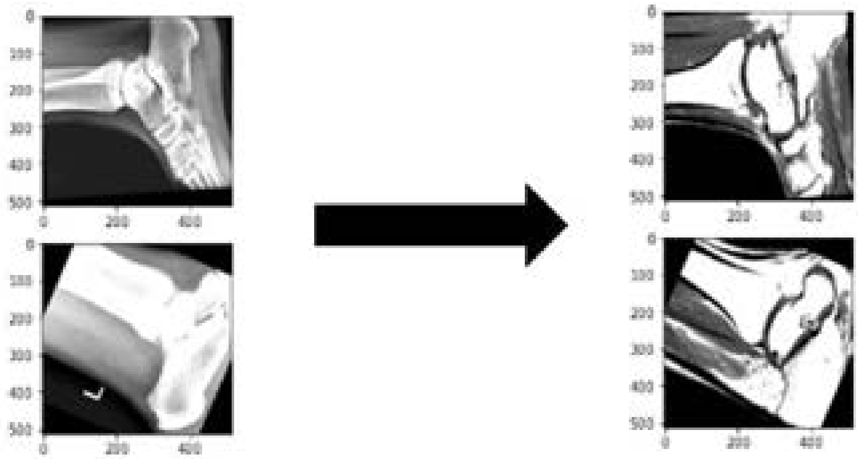


Figure 6.18: The workflow of translating X-ray image to MRI.

We use a combination of KL and MSE losses together with visual inspection to compare our model with others.

In order to evaluate the importance of using KL divergence and L1 loss, we initially trained the VAE model using MSE using L2. As expected, the results were more blurry than realistic. The loss function in equation 6.12 has been used here.

The "normal VAE", "customised VAE", "CGAN", and "CGAN with concatenation and convolution layer instead of upsampling" have been compared. The sub-figures of Figure 6.19 illustrate the results of these models.

The first model we used was VAE to map X-ray images to MRIs. Figure 6.19 illustrates how our VAE models perform when it is trained using L2+MSE loss.

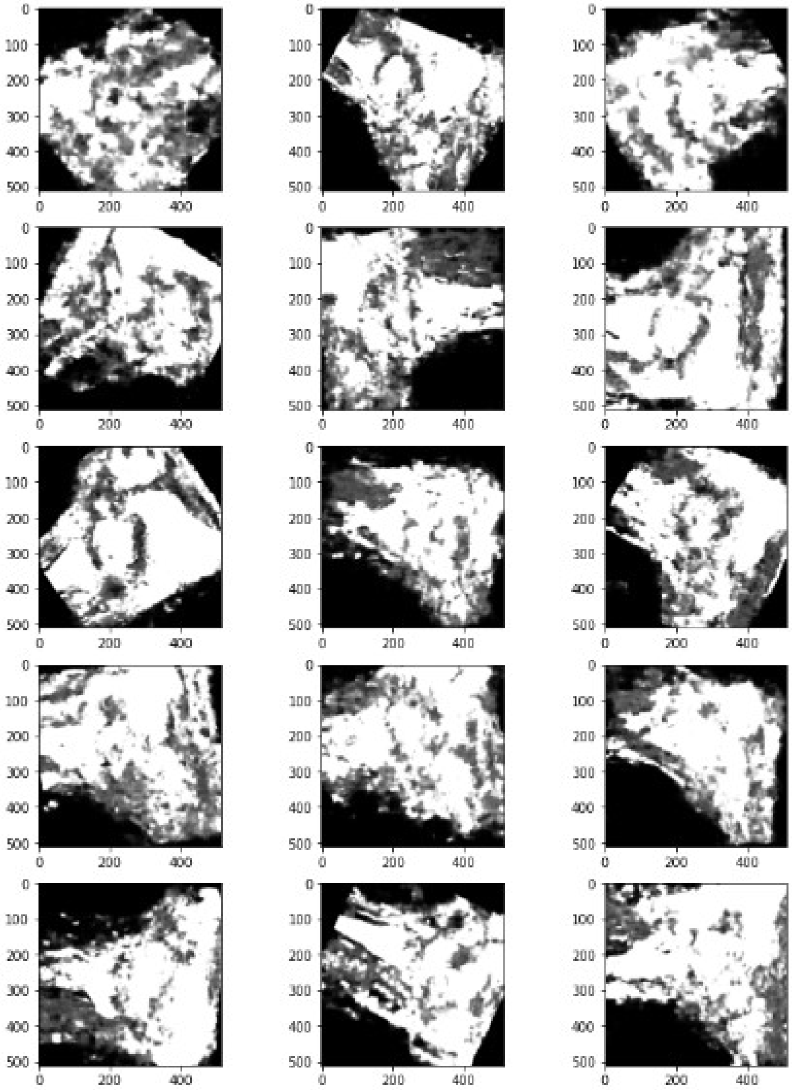


Figure 6.19: The result of VAE using the L2 MSE to train.

Figure 6.20 shows the result of VAE using the new loss function consisting of L1 and KL divergence loss. As it is obvious the result of VAE by itself is not promising. Comparison

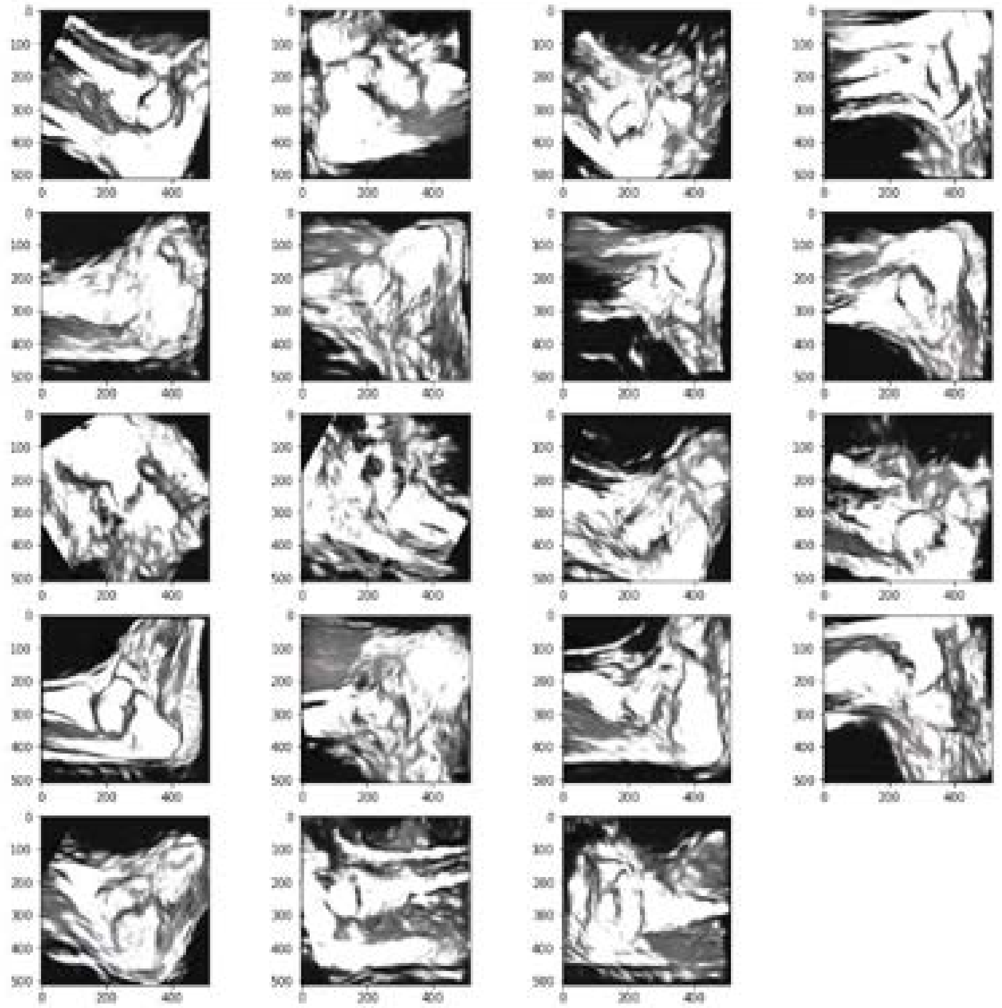


Figure 6.20: The result VAE with KL-divergence and L1 loss.

Moreover, we also show the results of our CGAN model and the state-of-the-art model in Figures 6.21 and 6.22.

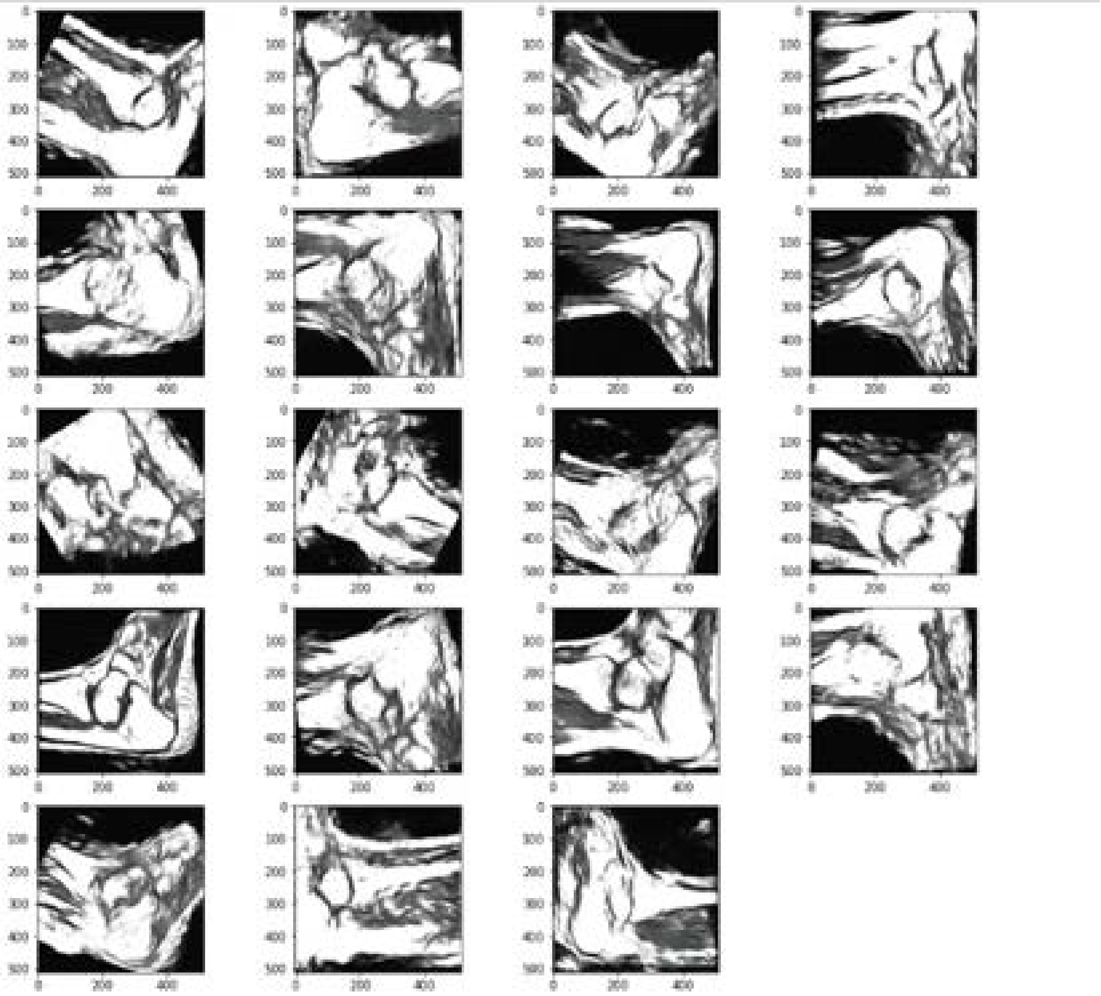


Figure 6.21: The result of conditional generative adversarial network.

In the Figure 6.23, the loss is shown over the number of epochs:

- The VAE architecture described above uses the new loss function.
- A VAE with a customised loss function over three layers of MRI data.
- It is based on the CGAN architecture described in [226].
- A state-of-the-art design of CGAN that utilises convolutional layers rather than upsampling, concatenation, and a customised loss function

As shown in Figure 6.23, training our model requires more time than training other models since we have more parameters to train and our model tends to overfit faster, but shows better performance than other models.

Figure 6.24 provides a visual comparison between the results of each model and the ground truth.

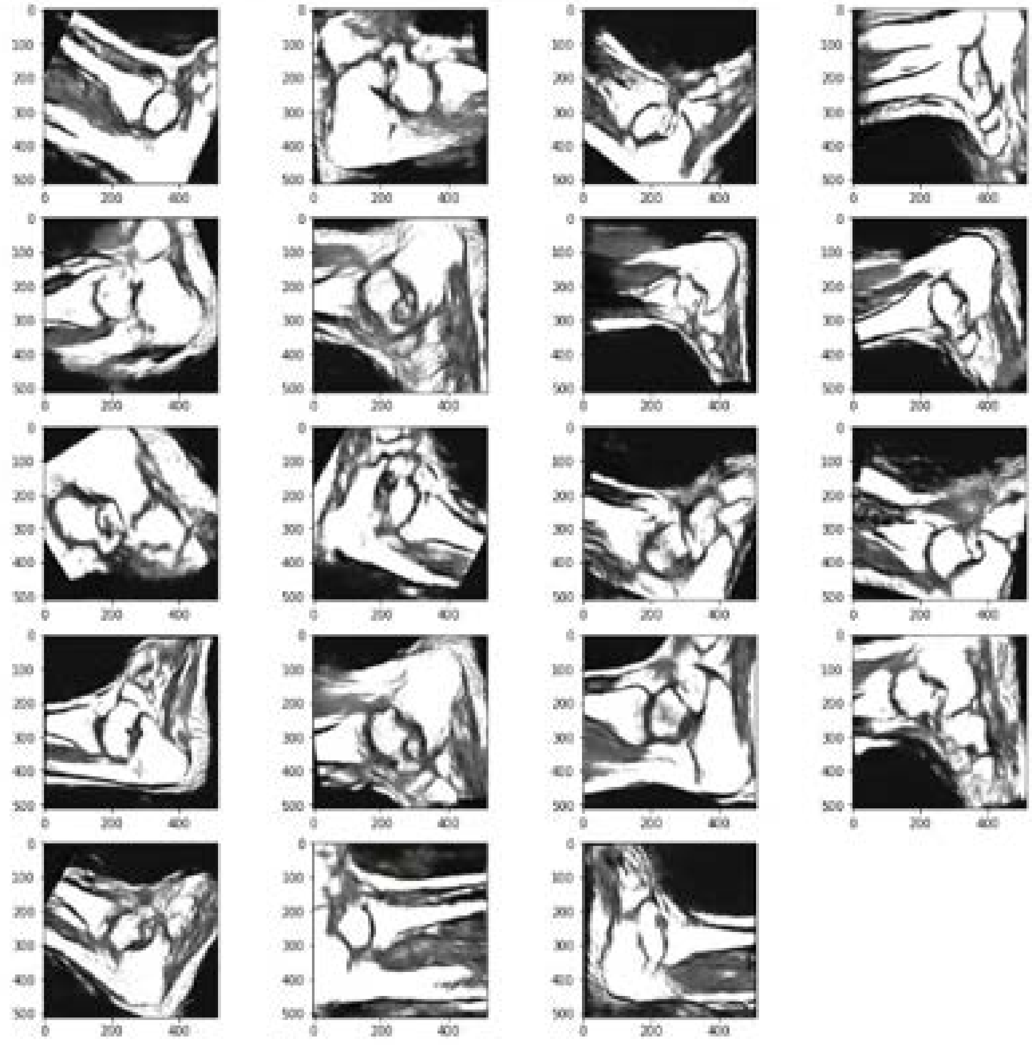


Figure 6.22: The result of our model to eliminate the shadow of overlapping bones on calcaneal bone.

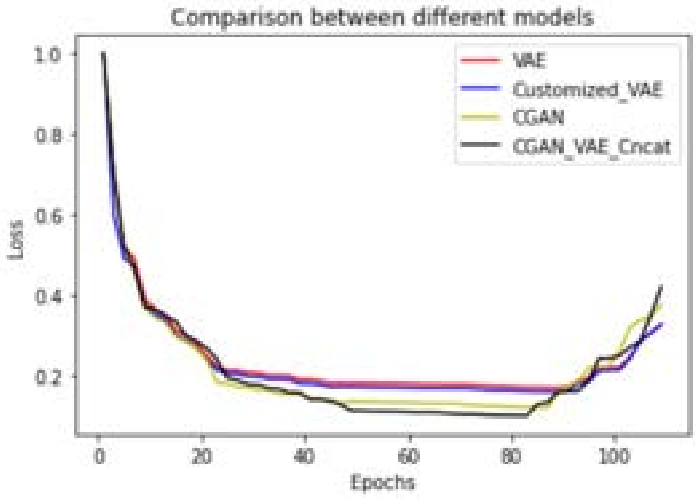


Figure 6.23: Comparison of loss between 4 models over number of epochs.

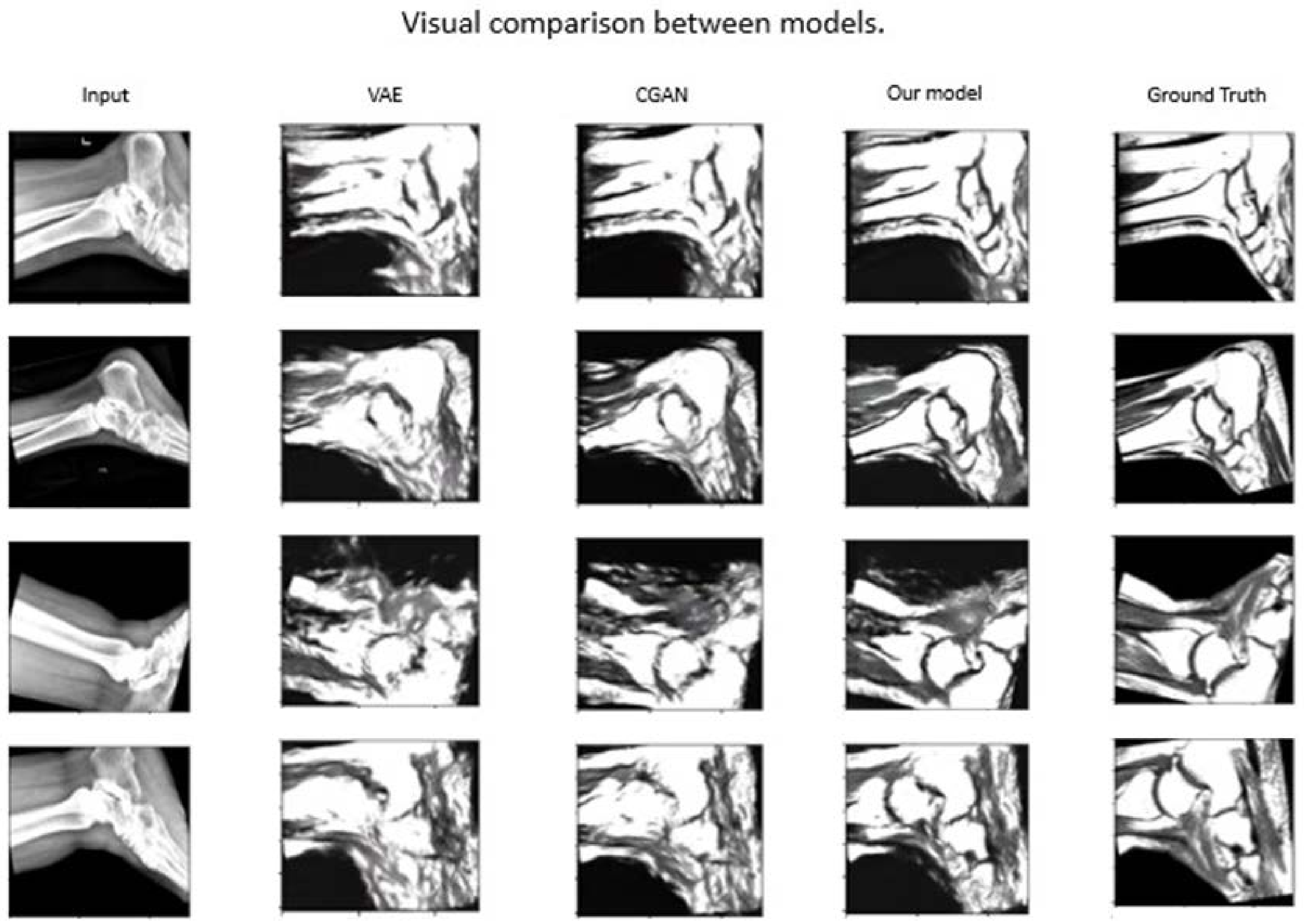


Figure 6.24: The visual comparison betweenthe results of VAE, CGAN and our design.

6.6 CONCLUSION

A translation of X-ray image to MRI is described in this chapter. Our dataset contained 40 pairs of X-ray images and MRIs that were augmented to have approximately 500 pairs of images. Our model was trained for 110 epochs using a batch size of 4. In addition, we have used KL and L1 loss functions in order to train the model, as well as three layers of MRI to improve its performance.

We found that our models performed better than VAE and CGAN. Thus, not only can our model be used for other image-to-image translation tasks, but also it has shown significant potential for use as a computer-aided design system to increase the accuracy of bone fracture diagnosis by providing tomographic images of that organ and omitting shadows of skin and overlapping bones (Figure 6.25).

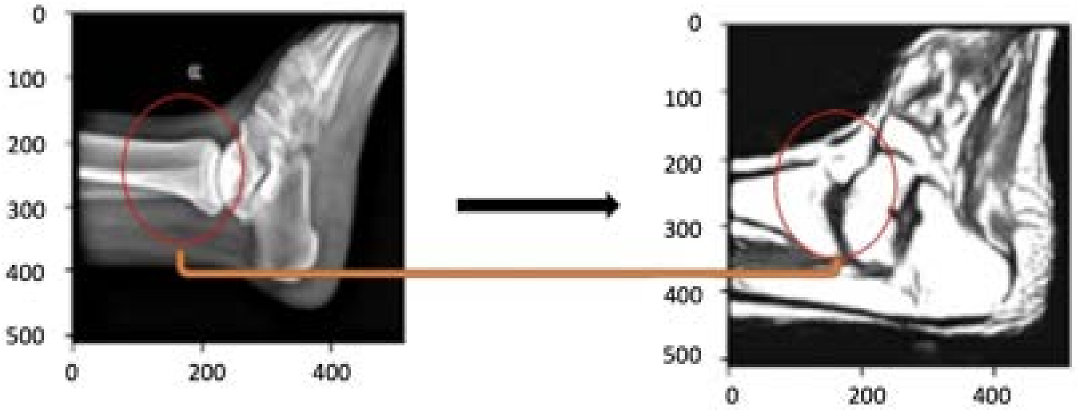


Figure 6.25: Mitigating shadows of overlapping bones.

The outcome of this research verifies the high potential of the proposed method in translating X-ray images to MRIs including sufficient bone diagnostic details even with such small dataset. It is believed that, having more pairs of matched datasets, there will be considerable improvement in the results. This method also paves the path for a full 3-D MRI to 2-D X-ray image translation, which requires more research.

7 CONCLUSION

7.1 SUMMARY

In this thesis, the focus is on the development of an automated clinical assistant diagnosis system with application to bone fracture diagnosis. During this research we have had access to a limited set of matched X-ray and MRI mostly taken from hand. X-rays, fast retrieved, and are widely accessible, cheap, and relatively learned by many clinicians. On the other hand, MRIs are expensive, less accessible, not so popular among clinicians and time demanding. Several machine learning methods have been used, including CNN, GAN, Faster R-CNN, and image super-resolution, as well as image processing methods, such as entropy and intensity measurement. As a result, we have designed an end-to-end model that is more efficient and requires less training and computation compared with the traditional approaches.

This study aims to develop a model that is applied to X-ray images to mitigate bone fracture misdiagnosis, decrease the workload of specialists, and reduce patient costs associated with bone fracture diagnosis. Three different approaches could be employed to reach these three targets. The first approach is to build a CAD system that shows some regions of interest to specialists. As a result, a specialist is not only able to focus his/her attention on regions where there is a high likelihood of existing fractures, but also can be asked to double-check those regions for better diagnosis. The second approach involves the use of a completely autonomous system that is capable of detecting fractures without any human involvement. Nevertheless, an expert checks the results again to confirm them. Lastly, we can build models that can generate higher resolution images or even translate the medical images into other formats and extract more information to assist in faster and more accurate diagnosis by clinicians. The aforementioned issues in fracture diagnosis are caused by the presence of artefacts in low-resolution X-ray images. The use of low resolution X-ray images may cause misdiagnosis in the case of occluded microfracture. It is more expensive and time-consuming to capture other modalities of medical images, such as MRIs, which can be useful for accurate diagnosis.

There is a need for accurate bone segmentation in all fracture diagnosis systems. An effective X-ray image segmentation should alleviate the effect of soft tissue in fracture region segmentation. To segment the bone perfectly, we used a hybrid model of Faster R-CNN and DRLSE in the Chapter 3 of this thesis. Due to the use of transfer learning and a hybrid model of NNs and image processing techniques, our method is not only more accurate than current methods but also requires less data for training and lower computational cost. The accuracy and precision of our model are significantly higher than those of X-net model. Also, it only requires region-based labelling rather than pixel-level labelling. In addition, the proposed approach best tackles localisation of dislocated bones as the result of fracture.

Low-resolution of X-ray images is the main cause of most missed diagnoses. To resolve this issue, in Chapter 4, we used pairs of high-resolution and low-resolution images to train the single-frame super-resolution model using CNNs and GANs. We compared the results with blind models such as bicubic interpolation using subjective and objective methods. Additionally, we have evaluated the model by implementing it on regions containing fractures. We have achieved better results than blind methods both in terms of MOS and PSNR.

Chapter 5 discusses the use of saliency maps based on entropy and intensity for fracture detection. The advantage of our model is that it does not require any training for it to work. A comparison was made between this model and feature-based method. Saliency maps are generated by analyzing the entropy and intensity of images, and then the fractures are detected using silhouettes and K-means clustering. For a more accurate diagnosis, the effect of super-resolution techniques is also evaluated on the saliency map. Based on the developed model in this section, we can define some regions of interest for clinicians to double-check in order to reduce human error. The results of this mode are not promising enough for use as an independent diagnosis system; however, it can be used as a CAD system.

In Chapter 6, we have described a model used to translate X-ray images into MRIs. X-rays cannot be used to diagnose some micro fractures or fractures in regions with complex bone structures. For example, If there is a suspicion of a fracture in the scaphoid or calcaneal bone, the patient should be monitored for a few weeks, or more expensive medical imaging such as MRI should be obtained. In our proposed network, X-ray images can be translated into MRIs, which will reduce the demand for taking MRIs. Moreover, our model can be used for other regions as well since it eliminates shadows caused by overlapping bones, soft tissue and skin. We have not only shown that our proposed model has significant potential in terms of subjective tests (by visualisation), but also compared our novel architecture to other available designs through comparing the reconstruction error.

7.2 LIMITATIONS AND FUTURE WORK

In this section, we discuss some limitations of this thesis and some problems, that machine learning faces in medical imaging. We then provide some discussions and suggestions for future research directions. It is noteworthy that we tried, in this thesis, to overcome some of these limitations through the use of novel approaches.

7.2.1 *Model Generalisation*

Among the most important limitations of machine learning approaches in fracture diagnosis and medical imaging in general is that these algorithms need to be generalised for different types of fractures and abnormalities and for different patients. A neural network might have trouble generalising to fracture types that weren't sufficiently represented in the training data. Such cases may result in misdiagnosis or under-performance. Due to their limited generalization, deep-learning models cannot be used in real-world applications. Furthermore, hyper-parameter tuning needs to be done in a way that these methods are not constrained by data distribution and can be used with other data sources. Moreover, this limitation can occur not only if the data domain changes, but also if data is collected by different instruments or if data is collected under different conditions with different amounts of noise. Assuring that NNs perform equally well on different datasets and have good performance on similar datasets will help them to be more practical.

7.2.2 *Data Limitation*

In order for neural networks to be effective, a comprehensive and diverse dataset is required. Datasets with limitations or biases may lead to suboptimal performance and hinder generalization. It is also important to take into account large class imbalances when segmenting medical images. The disadvantage of loss functions that take into account class imbalances is that they require additional parameters to adjust the class weighting and this can affect performance. Additionally, in some cases, neural network training requires specific types of data, such as pair of X-ray images and MRIs, which is extremely difficult to obtain in the real world. The future of medical imaging research should focus on gathering larger, more diverse, and accurately annotated datasets by specialist. The performance and generalization of neural networks will be enhanced as a result. The use of models with less complexity can also help reduce the need for large datasets.

7.2.3 *Interpreting Neural Networks*

It is common to consider deep learning models to be "black boxes," making it difficult to understand the rationale behind their decisions. To gain trust and acceptance, interpretability is crucial in medical decisions. On the other hand, the lack of interpretability may also lead to unpredictability and unreliability of the model. Furthermore, due to the complex and abstract nature of the learned features, explanation of NN-generated results to medical professionals is challenging. Medical professionals will be able to trust neural networks more if we develop methods to make them easier to interpret and provide visual or textual explanations for their decisions.

As the main objective of this research, we have tried to design an image translator to learn from a limited number of matched MRI - X-ray images so it can be later applied to many widely available cheap X-rays. The focus here has been on hand and ankle images due to big demands by A&E hospital departments and the availability of data. Clearly, the applications can be extended to many other body radiographs. On the other hand, by adding more data and computational resources the proposed image translation method can be extended to a full 3-D MRI to 2-D X-ray image mapping which can benefit a wider clinical applications.

As a conclusion, this thesis has demonstrated the potential of neural networks in medical imaging and fracture detection, while also highlighting challenges that must be addressed in order for their successful integration into clinical practice. A future research program in these areas will pave the way for safer, more accurate, and ethical AI-driven healthcare.

BIBLIOGRAPHY

[1] Peter Hallas and Trond Ellingsen. "Errors in fracture diagnoses in the emergency department–characteristics of patients and diurnal variation." In: *BMC Emergency Medicine* 6.1 (2006), pp. 1–5.

[2] Katharina Breininger. "Machine learning and deformation modeling for workflow-compliant image fusion during endovascular aortic repair." PhD thesis. Friedrich-Alexander-Universitaet Erlangen-Nuernberg (Germany), 2021.

[3] Jianchao Yang and Thomas Huang. "Image super-resolution: historical overview and future challenges." In: *Super-Resolution Imaging*. CRC Press, 2017, pp. 1–34.

[4] Lyndsey C Pickup. "Machine learning in multi-frame image super-resolution." PhD thesis. Oxford University, UK, 2007.

[5] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. "Image super-resolution as sparse representation of raw image patches." In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[6] Shah Hussain, Iqra Mubeen, Niamat Ullah, Syed Shahab Ud Din Shah, Bakhtawar Abduljalil Khan, Muhammad Zahoor, Riaz Ullah, Farhat Ali Khan, and Mujeeb A Sultan. "Modern Diagnostic Imaging Technique Applications and Risk Factors in the Medical Field: A Review." In: *BioMed Research International* 2022 (2022).

[7] Jacob Beutel, Harold L Kundel, Yongmin Kim, Richard L Van Metter, and Steven C Horii. *Handbook of medical imaging*. Vol. 3. Spie Press, 2000.

[8] MJ Yaffe and JA Rowlands. "X-ray detectors for digital radiography." In: *Physics in Medicine & Biology* 42.1 (1997), p. 1.

[9] Peter A Rinck. *Magnetic resonance in medicine: a critical introduction*. BoD–Books on Demand, 2019.

[10] Peter A Rinck. *Magnetic resonance in medicine: a critical introduction*. BoD–Books on Demand, 2019.

[11] Boer JA Vlaardingerbroek MT. *Magnetic resonance imaging theory and practice*. 1997.

[12] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features." In: *Scientific Data* 4.1 (2017), pp. 1–13.

[13] Joanna M Wardlaw, Eric E Smith, Geert J Biessels, Charlotte Cordonnier, Franz Fazekas, Richard Frayne, Richard I Lindley, John T O'Brien, Frederik Barkhof, Oscar R Benavente, et al. "Neuroimaging standards for research into small vessel disease and its contribution to ageing and neurodegeneration." In: *The Lancet Neurology* 12.8 (2013), pp. 822–838.

[14] Susan Brooks, Anita E Wluka, S Stuckey, and F Cicuttini. "The management of scaphoid fractures." In: *Journal of Science and Medicine in Sport* 8.2 (2005), pp. 181–189.

[15] Darshan D Ruikar, KC Santosh, and Ravindra S Hegadi. "Automated fractured bone segmentation and labeling from CT images." In: *Journal of medical systems* 43 (2019), pp. 1–13.

[16] Pierre Dodin, Johanne Martel-Pelletier, Jean-Pierre Pelletier, and François Abram. "A fully automated human knee 3D MRI bone segmentation using the ray casting technique." In: *Medical & biological engineering & computing* 49 (2011), pp. 1413–1424.

[17] Yun Pei, Wenzhuo Yang, Shangqing Wei, Rui Cai, Jialin Li, Shuxu Guo, Qiang Li, Jincheng Wang, and Xueyan Li. "Automated measurement of hip–knee–ankle angle on the unilateral lower limb X-rays using deep learning." In: *Physical and Engineering Sciences in Medicine* 44 (2021), pp. 53–62.

[18] Waqar Ahmad, Hazrat Ali, Zubair Shah, and Shoaib Azmat. "A new generative adversarial network for medical images super resolution." In: *Scientific Reports* 12.1 (2022), p. 9533.

[19] Dwarikanath Mahapatra, Behzad Bozorgtabar, and Rahil Garnavi. "Image super-resolution using progressive generative adversarial networks for medical image analysis." In: *Computerized Medical Imaging and Graphics* 71 (2019), pp. 30–39.

[20] Yuchong Gu, Zitao Zeng, Haibin Chen, Jun Wei, Yaqin Zhang, Binghui Chen, Yingqin Li, Yujuan Qin, Qing Xie, Zhuoren Jiang, et al. "MedSRGAN: Medical images super-resolution using generative adversarial networks." In: *Multimedia Tools and Applications* 79 (2020), pp. 21815–21840.

[21] Graeme J Awcock and Ray Thomas. *Applied image processing*. Springer, 1995.

[22] Thomas S Huang, William F Schreiber, and Oleh J Tretiak. "Image processing." In: *Proceedings of the IEEE* 59.11 (1971), pp. 1586–1609.

[23] Maria MP Petrou and Costas Petrou. *Image processing: the fundamentals*. John Wiley & Sons, 2010.

[24] Bruno Artacho and Andreas Savakis. "Waterfall atrous spatial pooling architecture for efficient semantic segmentation." In: *Sensors* 19.24 (2019), p. 5361.

[25] Laurent Najman and Michel Schmitt. "Watershed of a continuous function." In: *Signal Processing* 38.1 (1994), pp. 99–112.

[26] Manuel Pinheiro and José L Alves. "A new level-set-based protocol for accurate bone segmentation from CT imaging." In: *IEEE Access* 3 (2015), pp. 1894–1906.

[27] Marcel Krvcah, Gabor Szekely, and Remi Blanc. "Fully automatic and fast segmentation of the femur bone from 3D-CT images with no shape prior." In: *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2011, pp. 2087–2090.

[28] V Burdin and C Roux. "Surface segmentation of long bone structures from 3D CT images using a deformable contour model." In: *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 1. IEEE. 1994, pp. 512–513.

[29] Stuart SC Burnett, George Starkschall, Craig W Stevens, and Zhongxing Liao. "A deformable-model approach to semi-automatic segmentation of CT images demonstrated by application to the spinal canal." In: *Medical Physics* 31.2 (2004), pp. 251–263.

[30] A Franzle, M Sumkauskaite, J Hillengass, T Bauerle, and R Bendl. "Fully automated shape model positioning for bone segmentation in whole-body CT scans." In: *Journal of Physics: Conference Series*. Vol. 489. 1. IOP Publishing. 2014, p. 012029.

[31] Abdel-Razzak Natsheh, Prasad VS Ponnapalli, Nader Anani, Dalil Benchebra, Atef El-kholy, and Peter Norburn. "Segmentation of bone structure in sinus CT images using self-organizing maps." In: *IEEE International Conference on Imaging Systems and Techniques*. IEEE. 2010, pp. 294–299.

[32] Haoyan Guo, Sicong Song, Jinke Wang, Maozu Guo, Yuanzhi Cheng, Yadong Wang, and Shinichi Tamura. "3D surface voxel tracing corrector for accurate bone segmentation." In: *International Journal of Computer Assisted Radiology and Surgery* 13.10 (2018), pp. 1549–1563.

[33] Neeraj Sharma, Lalit M Aggarwal, et al. "Automated medical image segmentation techniques." In: *Journal of Medical Physics* 35.1 (2010), p. 3.

[34] Tanuj Puri, Glen M Blake, Kathleen M Curran, Hamish Carr, Amelia EB Moore, Niall Colgan, Martin J O'Connell, Paul K Marsden, Ignac Fogelman, and Michelle L Frost. "Semi-automatic region-of-interest validation at the femur in 18f-fluoride PET/CT." In: *Journal of Nuclear Medicine Technology* 40.3 (2012), pp. 168–174.

[35] James N Morgan and John A Sonquist. "Problems in the analysis of survey data, and a proposal." In: *Journal of the American Statistical Association* 58.302 (1963), pp. 415–434.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: convolutional networks for biomedical image segmentation." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

[37] David Marr and Ellen Hildreth. "Theory of edge detection." In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 207.1167 (1980), pp. 187–217.

[38] Josien PW Pluim and J Michael Fitzpatrick. "Image registration." In: *IEEE transactions on medical imaging* 22.11 (2003), pp. 1341–1343.

[39] Medha V Wyawahare, Pradeep M Patil, Hemant K Abhyankar, et al. "Image registration techniques: an overview." In: *International Journal of Signal Processing, Image Processing and Pattern Recognition* 2.3 (2009), pp. 11–28.

[40] Joseph V Hajnal and Derek LG Hill. *Medical image registration*. CRC press, 2001.

[41] Derek LG Hill, Philipp G Batchelor, Mark Holden, and David J Hawkes. "Medical image registration." In: *Physics in Medicine & Biology* 46.3 (2001), R1.

[42] Corinna Cortes and Vladimir Vapnik. "Support vector machine." In: *Machine Learning* 20.3 (1995), pp. 273–297.

[43] Warren S Mcculloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." In: *Bulletin of Mathematical Biology* 52.1 (1990), pp. 99–115.

[44] Paul Werbos. "Beyond Agression: "New Tools for Prediction and Analysis in the Behavioral Sciences." In: *Ph. D. Dissertation, Harvard University* (1974).

[45] PA Mirau, LW Jelinski, FA Bovey, TE Long, JE McGrath, and S Richard. "Encyclopedia of physical science and technology." In: *Polymers* (2003), pp. 857–901.

[46] Mingyuan Xin and Yong Wang. "Research on image classification model based on deep convolution neural network." In: *EURASIP Journal on Image and Video Processing* 2019 (2019), pp. 1–11.

[47] Ceren Güzel Turhan and Hasan Sakir Bilge. "Recent trends in deep generative models: a review." In: *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. IEEE. 2018, pp. 574–579.

[48] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." In: *Insights into Imaging* 9.4 (2018), pp. 611–629.

[49] Keiron O'Shea and Ryan Nash. "An introduction to convolutional neural networks." In: *ArXiv Preprint ArXiv:1511.08458* (2015).

[50] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." In: *Insights into Imaging* 9.4 (2018), pp. 611–629.

[51] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[52] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[54] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." In: *ArXiv Preprint ArXiv:1409.1556* (2014).

[55] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. "The shattered gradients problem: If resnets are the answer, then what is the question?" In: *International Conference on Machine Learning*. PMLR. 2017, pp. 342–350.

[56] Andreas Veit, Michael J Wilber, and Serge Belongie. "Residual networks behave like ensembles of relatively shallow networks." In: *Advances in Neural Information Processing Systems* 29 (2016).

[57] Masoud Abdi and Saeid Nahavandi. "Multi-residual networks: improving the speed and accuracy of residual networks." In: *ArXiv Preprint ArXiv:1609.05672* (2016).

[58] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. "Polynet: A pursuit of structural diversity in very deep networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 718–726.

[59] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *Nature* 521.7553 (2015), pp. 436–444.

[60] Sepp Hochreiter and Jurgen Schmidhuber. "Long short-term memory." In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[61] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. "Recurrent neural network based language model." In: *Interspeech*. Vol. 2. 3. Makuhari. 2010, pp. 1045–1048.

[62] Yann LeCun et al. "LeNet-5, convolutional neural networks." In: *URL: http://yann. lecun. com/exdb/lenet* 20.5 (2015), p. 14.

[63] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. "The history began from alexnet: a comprehensive survey on deep learning approaches." In: *ArXiv Preprint ArXiv:1803.01164* (2018).

[64] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved techniques for training GANs." In: *Advances in neural information processing systems* 29 (2016).

[65] Clara M Barreira, Mehdi Bouslama, Diogo C Haussen, Jonathan A Grossberg, Blaise Baxter, Thomas Devlin, Michael Frankel, and Raul G Nogueira. "Abstract WP61: automated large artery occlusion detection IN stroke imaging-ALADIN study." In: *Stroke* 49.Suppl_1 (2018), AWP61–AWP61.

[66] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. "A guide to deep learning in healthcare." In: *Nature Medicine* 25.1 (2019), pp. 24–29.

[67] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

[68] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In: *Advances in Neural Information Processing Systems* 27 (2014).

[69] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." In: *Journal of big Data* 8 (2021), pp. 1–74.

[70] Tom Dietterich. "Overfitting and undercomputing in machine learning." In: *ACM Computing Surveys (CSUR)* 27.3 (1995), pp. 326–327.

[71] Gregory Paris, Denis Robilliard, and Cyril Fonlupt. "Exploring overfitting in genetic programming." In: *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer. 2003, pp. 267–277.

[72] David D Jensen and Paul R Cohen. "Multiple comparisons in induction algorithms." In: *Machine Learning* 38.3 (2000), pp. 309–338.

[73] Xue Ying. "An overview of overfitting and its solutions." In: *Journal of Physics: Conference Series*. Vol. 1168. 2. IOP Publishing. 2019, p. 022022.

[74] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. "Early stopping and non-parametric regression: an optimal data-dependent stopping rule." In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 335–366.

[75] Johannes Fürnkranz. "Pruning algorithms for rule learning." In: *Machine Learning* 27.2 (1997), pp. 139–172.

[76] Christian Garbin, Xingquan Zhu, and Oge Marques. "Dropout vs. batch normalization: an empirical study of their impact to deep learning." In: *Multimedia Tools and Applications* 79.19 (2020), pp. 12777–12815.

[77] Yohannes Kassahun, Bingbin Yu, Abraham Temesgen Tibebu, Danail Stoyanov, Stamatia Giannarou, Jan Hendrik Metzen, and Emmanuel Vander Poorten. "Surgical robotics beyond enhanced dexterity instrumentation: a survey of machine learning techniques and their role in intelligent and autonomous surgical actions." In: *International Journal of Computer Assisted Radiology and Surgery* 11.4 (2016), pp. 553–568.

[78] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems* 27 (2014).

[79] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "Imagenet large scale visual recognition challenge." In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[80] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. "Dermatologist-level classification of skin cancer with deep neural networks." In: *nature* 542.7639 (2017), pp. 115–118.

[81] Jie-Zhi Cheng, Dong Ni, Yi-Hong Chou, Jing Qin, Chui-Mei Tiu, Yeun-Chung Chang, Chiun-Sheng Huang, Dinggang Shen, and Chung-Ming Chen. "Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans." In: *Scientific Reports* 6.1 (2016), pp. 1–13.

[82] Issam El Nakadi, Christian Mélot, Jean Closset, Véronique De Moor, Kahina Bétroune, Pascale Feron, Pierre Lingier, and Michel Gelin. "Evaluation of da Vinci Nissen fundoplication clinical results and cost minimization." In: *World Journal of Surgery* 30.6 (2006), pp. 1050–1054.

[83] Holger A Haenssle, Christine Fink, Roland Schneiderbauer, Ferdinand Toberer, Timo Buhl, Andreas Blum, A Kalloo, A Ben Hadj Hassen, Luc Thomas, A Enk, et al. "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition

in comparison to 58 dermatologists." In: *Annals of Oncology* 29.8 (2018), pp. 1836–1842.

[84] Ryan Poplin, Avinash V Varadarajan, Katy Blumer, Yun Liu, Michael V McConnell, Greg S Corrado, Lily Peng, and Dale R Webster. "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning." In: *Nature Biomedical Engineering* 2.3 (2018), pp. 158–164.

[85] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. "Clinically applicable deep learning for diagnosis and referral in retinal disease." In: *Nature Medicine* 24.9 (2018), pp. 1342–1350.

[86] Thijs Kooi, Geert Litjens, Bram Van Ginneken, Albert Gubern-Mérida, Clara I Sánchez, Ritse Mann, Ard den Heeten, and Nico Karssemeijer. "Large scale deep learning for computer aided detection of mammographic lesions." In: *Medical Image Analysis* 35 (2017), pp. 303–312.

[87] Amir Jamaludin, Timor Kadir, and Andrew Zisserman. "SpineNet: automatically pinpointing classification evidence in spinal MRIs." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 166–175.

[88] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 2503–2510.

[89] Samuel W Hasinoff. "Saturation (imaging)." In: *Computer Vision: A Reference Guide*. Springer, 2021, pp. 1107–1109.

[90] Albertus Cotton. "The limitation of the X-ray in the diagnosis of certain bone and joint disease." In: *JBJS* 2.2 (1915), pp. 217–240.

[91] V Zharkova, S Ipson, J Aboudarham, and B Bentley. "Survey of image processing techniques, EGSO internal deliverable." In: (2002).

[92] Ding Feng. "Segmentation of bone structures in X-ray images." In: *School of Computing National University of Singapore* (2006).

[93] Geoff Dougherty. *Medical image processing: techniques and applications*. Springer Science & Business Media, 2011.

[94] John C Russ. *The image processing handbook sixth edition*. 2011.

[95] Lian Ding, Kai Zhao, Xiaodong Zhang, Xiaoying Wang, and Jue Zhang. "A lightweight U-net architecture multi-scale convolutional network for pediatric hand bone segmentation in X-ray image." In: *IEEE Access* 7 (2019), pp. 68436–68445.

[96] Cosmin Cernazanu-Glavan and Stefan Holban. "Segmentation of bone structure in X-ray images using convolutional neural network." In: *Adv. Electr. Comput. Eng* 13.1 (2013), pp. 87–94.

[97] Laurent Najman and Michel Schmitt. "Watershed of a continuous function." In: *Signal Processing* 38.1 (1994), pp. 99–112.

[98] Nobuyuki Otsu. "A threshold selection method from gray-level Histograms." In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[99] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm." In: *Procedia Computer Science* 54 (2015), pp. 764–771.

[100] Nils Plath, Marc Toussaint, and Shinichi Nakajima. "Multi-class image segmentation using conditional Random Fields and Global Classification." In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 817–824.

[101] J-L Starck, Michael Elad, and David L Donoho. "Image decomposition via the combination of sparse representations and a variational approach." In: *IEEE Transactions on Image Processing* 14.10 (2005), pp. 1570–1582.

[102] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: active contour models." In: *International Journal of Computer Vision* 1.4 (1988), pp. 321–331.

[103] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. "Image segmentation using deep learning: a survey." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[104] Cristina Stolojescu-Crisan and Stefan Holban. "A comparison of X-ray image segmentation techniques." In: *Advances in Electrical and Computer Engineering* 13.3 (2013), pp. 85–92.

[105] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. "Simultaneous detection and segmentation." In: *European Conference on Computer Vision*. Springer. 2014, pp. 297–312.

[106] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. "Hypercolumns for object segmentation and fine-grained localization." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 447–456.

[107] Pedro O O Pinheiro, Ronan Collobert, and Piotr Dollár. "Learning to segment object candidates." In: *Advances in Neural Information Processing Systems* 28 (2015).

[108] Jifeng Dai, Kaiming He, and Jian Sun. "Instance-aware semantic segmentation via multi-task network cascades." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3150–3158.

[109] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich *f*eature hierarchies for accurate object detection and semantic segmentation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587.

[110] Ross Girshick. "Fast R-CNN." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1440–1448.

[111] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. "Selective search for object recognition." In: *International Journal of Computer Vision* 104.2 (2013), pp. 154–171.

[112] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: towards real-time object detection with region proposal networks." In: *Advances in Neural Information Processing Systems* 28 (2015).

[113] Joseph Redmon and Ali Farhadi. "Yolov3: an incremental improvement." In: *arXiv preprint arXiv:1804.02767* (2018).

[114] Qiang Yang and Xindong Wu. "10 challenging problems in data mining research." In: *International Journal of Information Technology & Decision Making* 5.04 (2006), pp. 597–604.

[115] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning." In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2009), pp. 1345–1359.

[116] Stanley Osher and James A Sethian. "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations." In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49.

[117] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." In: *Journal of Big Data* 3.1 (2016), pp. 1–40.

[118] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. "Distance regularized level set evolution and its application to image segmentation." In: *IEEE Transactions on Image Processing* 19.12 (2010), pp. 3243–3254.

[119] Kuan-Hung Shih, Ching-Te Chiu, Jiou-Ai Lin, and Yen-Yu Bu. "Real-time object detection with reduced region proposal network via multi-feature concatenation." In: *IEEE Transactions on Neural Networks and Learning Systems* 31.6 (2019), pp. 2164–2173.

[120] Joseph Bullock, Carolina Cuesta-Lázaro, and Arnau Quera-Bofarull. "XNet: A convolutional neural network (CNN) implementation for medical X-ray image segmentation suitable for small datasets." In: *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Vol. 10953. SPIE. 2019, pp. 453–463.

[121] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Image super-resolution using deep convolutional networks." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2015), pp. 295–307.

[122] Laura Palagi. "Global optimization issues in deep network regression: an overview." In: *Journal of Global Optimization* 73.2 (2019), pp. 239–277.

[123] Leon Bottou and Olivier Bousquet. "The tradeoffs of large scale learning." In: *Advances in Neural Information Processing Systems* 20 (2007).

[124] Dianyuan Han. "Comparison of commonly used image interpolation methods." In: *Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Atlantis Press. 2013, pp. 1556–1559.

[125] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Learning a deep convolutional network for image super-resolution." In: *European Conference on Computer Vision*. Springer. 2014, pp. 184–199.

[126] Zheng Hui, Xiumei Wang, and Xinbo Gao. "Fast and accurate single image super-resolution via information distillation network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 723–731.

[127] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. "Deep back-projection networks for super-resolution." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1664–1673.

[128] Assaf Shocher, Nadav Cohen, and Michal Irani. ""Zero-shot" super-resolution using deep internal learning." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3118–3126.

[129] Yuchen Fan, Jiahui Yu, Ding Liu, and Thomas S Huang. "Scale-wise convolution for image restoration." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 10770–10777.

[130] William Harrison, Ashley W Newton, and Graham Cheung. "The litigation cost of negligent scaphoid fracture management." In: *European Journal of Emergency Medicine* 22.2 (2015), pp. 142–143.

[131] Jonathan McConnell, Renata Eyres, and Julie Nightingale. *Interpreting trauma radiographs*. John Wiley & Sons, 2008.

[132] Michael Smith, Gregory I Bain, Perry C Turner, and Adam C Watts. "Review of imaging of scaphoid fractures." In: *ANZ journal of surgery* 80.1-2 (2010), pp. 82–90.

[133] Jennifer J Donald and Stuart A Barnard. "Common patterns in 558 diagnostic radiology errors." In: *Journal of medical imaging and radiation oncology* 56.2 (2012), pp. 173–178.

[134] Michal Aharon, Michael Elad, and Alfred Bruckstein. "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation." In: *IEEE Transactions on Signal Processing* 54.11 (2006), pp. 4311–4322.

[135] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines." In: *ICML*. 2010.

[136] Marc Claesen and Bart De Moor. "Hyperparameter search in machine learning." In: *arXiv preprint arXiv:1502.02127* (2015).

[137] Saeid Sanei, Delaram Jarchi, and Anthony G Constantinides. *Body sensor networking, design and algorithms*. John Wiley & Sons, 2020.

[138] ER Davies and Matthew Turk. *Advanced methods and deep learning in computer vision*. Elsevier, 2021.

[139] Saeid Sanei and Jonathon A Chambers. *EEG signal processing and machine learning*. John Wiley & Sons, 2021.

[140] Jithin Saji Isaac and Ramesh Kulkarni. "Super resolution techniques for medical image processing." In: *2015 International Conference on Technologies for Sustainable Development (ICTSD)*. IEEE. 2015, pp. 1–6.

[141] Kawther Aarizou and Abdelhamid Loukil. "Self-similarity single image super-resolution based on blur kernel estimation for texture reconstruction." In: *International Journal of Computational Science and Engineering* 25.1 (2022), pp. 64–73.

[142] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. "Photo-realistic single image super-resolution using a generative adversarial network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4681–4690.

[143] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. "Dictionary learning algorithms for sparse representation." In: *Neural Computation* 15.2 (2003), pp. 349–396.

[144] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. "Photo-realistic single image super-resolution using a generative adversarial network." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 4681–4690.

[145] Pishtiwan HS Kalmet, Sebastian Sanduleanu, Sergey Primakov, Guangyao Wu, Arthur Jochems, Turkey Refaee, Abdalla Ibrahim, Luca v Hulst, Philippe Lambin, and Martijn Poeze. "Deep learning in fracture detection: a narrative review." In: *Acta Orthopaedica* 91.2 (2020), pp. 215–220.

[146] Tian Tai Peng et al. "Detection of femur fractures in X-ray images." In: *Master of Science Thesis, National University of Singapore* (2002).

[147] Dennis Wen-Hsiang Yap, Ying Chen, Wee Kheng Leow, Tet Sen Howe, and Meng Ai Png. "Detecting femur fractures by texture analysis of trabeculae." In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. IEEE. 2004, pp. 730–733.

[148] Tai Peng Tian, Ying Chen, Wee Kheng Leow, Wynne Hsu, Tet Sen Howe, and Meng Ai Png. "Computing neck-shaft angle of femur for X-ray fracture detection." In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2003, pp. 82–89.

[149] DM Ryder, SL King, CJ Oliff, and E Davies. "A possible method of monitoring bone fracture and bone characteristics using a noninvasive acoustic technique." In: *International Conference on Acoustic Sensing and Imaging, 1993.* IET. 1993, pp. 159–163.

[150] JJ Kaufman, A Chiabrera, M Hatem, NZ Hakim, M Figueiredo, P Nasser, S Lattuga, AA Pilla, and RS Siffert. "A neural network approach for bone fracture healing assessment." In: *IEEE Engineering in Medicine and Biology Magazine* 9.3 (1990), pp. 23–30.

[151] Thomas M Link, Sharmila Majumdar, Werner Konermann, Norbert Meier, John C Lin, David Newitt, Xiaolong Ouyang, Peter E Peters, and Harry K Genant. "Texture analysis of direct magnification radiographs of vertebral specimens: correlation with bone mineral density and biomechanical properties." In: *Academic Radiology* 4.3 (1997), pp. 167–176.

[152] Xiaolong Ouyang, Sharmila Majumdar, Thomas M Link, Ying Lu, Peter Augat, John Lin, David Newitt, and Harry K Genant. "Morphometric texture analysis of spinal trabecular bone structure assessed using orthogonal radiographic projections." In: *Medical Physics* 25.10 (1998), pp. 2037–2045.

[153] Andrzej Materka, Piotr Cichy, and Jerzy Tuliszkiewicz. "Texture analysis of X-ray images for detection of changes in bone mass and structure." In: *Texture Analysis in Machine Vision*. World Scientific, 2000, pp. 189–195.

[154] JF Veenland, TM Link, W Konermann, N Meier, JL Grashuis, and ES Gelsema. "Unraveling the role of structure and density in determining vertebral bone strength." In: *Calcified Tissue International* 61.6 (1997), pp. 474–479.

[155] Philip Caligiuri, Maryellen L Giger, and Murray Favus. "Multifractal radiographic analysis of osteoporosis." In: *Medical Physics* 21.4 (1994), pp. 503–508.

[156] Andrzej Materka, Michal Strzelecki, et al. "Texture analysis methods–a review." In: *Technical University of Lodz, Institute of Electronics, COST B11 Report, Brussels* 10.1.97 (1998), p. 4968.

[157] David Palma, Otto Visser, Frank J Lagerwaard, Jose Belderbos, Ben Slotman, and Suresh Senan. "Treatment of stage I NSCLC in elderly patients: a population-based matched-pair comparison of stereotactic radiotherapy versus surgery." In: *Radiotherapy and Oncology* 101.2 (2011), pp. 240–244.

[158] AH Mir, M Hanmandlu, and SN Tandon. "Texture analysis of CT images." In: *IEEE Engineering in Medicine and Biology Magazine* 14.6 (1995), pp. 781–786.

[159] Isaiah J Fidler. "Tumor heterogeneity and the biology of cancer invasion and metastasis." In: *Cancer Research* 38.9 (1978), pp. 2651–2660.

[160] Omar S Al-Kadi, D Watson, et al. "Texture analysis of aggressive and nonaggressive lung tumor CE CT images." In: *IEEE Transactions on Biomedical Engineering* 55.7 (2008), pp. 1822–1830.

[161] Sugama Chicklore, Vicky Goh, Musib Siddique, Arunabha Roy, Paul K Marsden, and Gary JR Cook. "Quantifying tumour heterogeneity in 18F-FDG PET/CT imaging by texture analysis." In: *European Journal of Nuclear Medicine and Molecular Imaging* 40.1 (2013), pp. 133–140.

[162] Sugama Chicklore, Vicky Goh, Musib Siddique, Arunabha Roy, Paul K Marsden, and Gary JR Cook. "Quantifying tumour heterogeneity in 18F-FDG PET/CT imaging by texture analysis." In: *European Journal of Nuclear Medicine and Molecular Imaging* 40.1 (2013), pp. 133–140.

[163] Ronica H Nanda, Yuan Liu, Theresa W Gillespie, John L Mikell, Suresh S Ramalingam, Felix G Fernandez, Walter J Curran, Joseph Lipscomb, and Kristin A Higgins. "Stereotactic body radiation therapy versus no treatment for early stage non-small cell lung cancer in medically inoperable elderly patients: a national cancer data base analysis." In: *Cancer* 121.23 (2015), pp. 4222–4230.

[164] Franko Hržić, Ivan Štajduhar, Sebastian Tschauner, Erich Sorantin, and Jonatan Lerga. "Local-entropy based approach for X-ray image segmentation and fracture detection." In: *Entropy* 21.4 (2019), p. 338.

[165] Du-Yih Tsai, Yongbum Lee, and Eri Matsuyama. "Information entropy measure for evaluation of image quality." In: *Journal of Digital Imaging* 21.3 (2008), pp. 338–347.

[166] Lihua Min, Xiaoping Yang, and Changfeng Gui. "Entropy estimates and large-time behavior of solutions to a fourth-order nonlinear degenerate equation." In: *Communications in Contemporary Mathematics* 15.04 (2013), p. 1250066.

[167] William J Frawley, Gregory Piatetsky-Shapiro, and Christopher J Matheus. "Knowledge discovery in databases: an overview." In: *AI Magazine* 13.3 (1992), pp. 57–57.

[168] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. "From data mining to knowledge discovery in databases." In: *AI Magazine* 17.3 (1996), pp. 37–37.

[169] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. "Cluster analysis and display of genome-wide expression patterns." In: *Proceedings of the National Academy of Sciences* 95.25 (1998), pp. 14863–14868.

[170] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[171] Amir Ahmad and Lipika Dey. "A k-mean clustering algorithm for mixed numeric and categorical data." In: *Data & Knowledge Engineering* 63.2 (2007), pp. 503–527.

[172] Xiaohong W Gao, Rui Hui, and Zengmin Tian. "Classification of CT brain images based on deep learning networks." In: *Computer Methods and Programs in Biomedicine* 138 (2017), pp. 49–56.

[173] Qi Dou, Lequan Yu, Hao Chen, Yueming Jin, Xin Yang, Jing Qin, and Pheng-Ann Heng. "3D deeply supervised network for automated segmentation of volumetric medical images." In: *Medical Image Analysis* 41 (2017), pp. 40–54.

[174] Andrew G Taylor, Clinton Mielke, and John Mongan. "Automated detection of moderate and large pneumothorax on frontal chest X-rays using deep convolutional neural networks: a retrospective study." In: *PLoS Medicine* 15.11 (2018), e1002697.

[175] Alejandro Rodriguez-Ruiz, Elizabeth Krupinski, Jan-Jurre Mordang, Kathy Schilling, Sylvia H Heywang-Köbrunner, Ioannis Sechopoulos, and Ritse M Mann. "Detection of breast cancer with mammography: effect of an artificial intelligence support system." In: *Radiology* 290.2 (2019), pp. 305–314.

[176] Robert Lindsey, Aaron Daluiski, Sumit Chopra, Alexander Lachapelle, Michael Mozer, Serge Sicular, Douglas Hanel, Michael Gardner, Anurag Gupta, Robert Hotchkiss, et al. "Deep neural network improves fracture detection by clinicians." In: *Proceedings of the National Academy of Sciences* 115.45 (2018), pp. 11591–11596.

[177] Takaaki Urakawa, Yuki Tanaka, Shinichi Goto, Hitoshi Matsuzawa, Kei Watanabe, and Naoto Endo. "Detecting intertrochanteric hip fractures with orthopedist-level accuracy using a deep convolutional neural network." In: *Skeletal Radiology* 48.2 (2019), pp. 239–244.

[178] Seok Won Chung, Seung Seog Han, Ji Whan Lee, Kyung-Soo Oh, Na Ra Kim, Jong Pil Yoon, Joon Yub Kim, Sung Hoon Moon, Jieun Kwon, Hyo-Jin Lee, et al. "Automated detection and classification of the proximal humerus fracture by using deep learning algorithm." In: *Acta Orthopaedica* 89.4 (2018), pp. 468–473.

[179] Yang Yang, Lin-Feng Yan, Xin Zhang, Yu Han, Hai-Yan Nan, Yu-Chuan Hu, Bo Hu, Song-Lin Yan, Jin Zhang, Dong-Liang Cheng, et al. "Glioma grading on conventional MR images: a deep learning study with transfer learning." In: *Frontiers in Neuroscience* 12 (2018), p. 804.

[180] TC Anu and R Raman. "Detection of bone fracture using image processing methods." In: *Int J Comput Appl* 975 (2015), p. 8887.

[181] Saket Tibrewal, Prakash Jayakumar, Sujit Vaidya, and Swee Chai Ang. "Role of MRI in the diagnosis and management of patients with clinical scaphoid fracture." In: *International Orthopaedics* 36.1 (2012), pp. 107–110.

[182] Saket Tibrewal, Prakash Jayakumar, Sujit Vaidya, and Swee Chai Ang. "Role of MRI in the diagnosis and management of patients with clinical scaphoid fracture." In: *International Orthopaedics* 36.1 (2012), pp. 107–110.

[183] IJ Leslie and RA Dickson. "The Fractured carpal scaphoid. Natural history and factors influencing outcome." In: *The Journal of Bone and Joint Surgery. British volume* 63.2 (1981), pp. 225–230.

[184] M Waizenegger, NJ Barton, TRC Davis, and ML Wastie. "Clinical signs in scaphoid fractures." In: *Journal of Hand Surgery* 19.6 (1994), pp. 743–747.

[185] John C Hunter, Eva M Escobedo, Anthony J Wilson, Douglas P Hanel, Gordon C Zink-Brody, and FA Mann. "MR imaging of clinically suspected scaphoid fractures." In: *AJR. American Journal of Roentgenology* 168.5 (1997), pp. 1287–1293.

[186] M Waizenegger, NJ Barton, TRC Davis, and ML Wastie. "Clinical signs in scaphoid fractures." In: *Journal of Hand Surgery* 19.6 (1994), pp. 743–747.

[187] Bo Munk, Johnny Frokjaer, Claus Falck Larsen, Hans Gad Johannsen, Lars Lundager Rasmussen, Annette Edal, and Lilla Demeney Rasmussen. "Diagnosis of scaphoid fractures: a prospective multicenter study of 1,052 patients with 160 fractures." In: *Acta Orthopaedica Scandinavica* 66.4 (1995), pp. 359–360.

[188] JOSEPH J Dias, JOHN Thompson, NJ Barton, and PJ Gregg. "Suspected scaphoid fractures. The value of radiographs." In: *The Journal of Bone and Joint Surgery. British volume* 72.1 (1990), pp. 98–101.

[189] Aditya Daftary, Andrew H Haims, and Michael R Baumgaertner. "Fractures of the calcaneus: a review with emphasis on CT." In: *Radiographics* 25.5 (2005), pp. 1215–1226.

[190] David W Stoller, Phillip FJ Tirman, Miriam A Bredella, Salvador Beltram, Robert M Branstetter, and Simon Blease. "Diagnostic imaging orthopaedics." In: *Diagnostic Imaging Orthopaedics*. 2004, pp. lxxvi–933.

[191] M De Filippo, A Corsi, L Evaristi, C Bertoldi, N Sverzellati, R Averna, P Crotti, G Bini, O Tamburrini, M Zompatori, et al. "Critical issues in radiology requests and reports." In: *La Radiologia Medica* 116.1 (2011), pp. 152–162.

[192] Massimo De Filippo, Emanuele Azzali, Antonella Pesce, Luca Saba, Maurizio Mostardi, Daniele Borgia, Antonio Barile, Raffaella Capasso, Fabio De Palmi, and Francesco Caravaggio. "CT arthrography for evaluation of autologous chondrocyte and chondral-inductor scaffold implantation in the osteochondral lesions of the talus." In: *Acta Biomedica* 87.3 (2016), pp. 51–6.

[193] M De Filippo, C Rovani, JJ Sudberry, F Rossi, F Pogliacomi, and M Zompatori. "Magnetic resonance imaging comparison of intra-articular cavernous synovial hemangioma and cystic synovial hyperplasia of the knee." In: *Acta Radiologica* 47.6 (2006), pp. 581–584.

[194] Antonio Pinto, Luca Brunese, Fabio Pinto, Ciro Acampora, and Luigia Romano. "E-learning and education in radiology." In: *European Journal of Radiology* 78.3 (2011), pp. 368–371.

[195] Enrique Guerado, María Luisa Bertrand, and Juan Ramón Cano. "Management of calcaneal fractures: what have we learnt over the years?" In: *Injury* 43.10 (2012), pp. 1640–1650.

[196] Charles Peterfy and Manish Kothari. "Imaging osteoarthritis: magnetic resonance imaging versus X-ray." In: *Current Rheumatology Reports* 8.1 (2006), pp. 16–21.

[197] Martin J Breitenseher, Viktor M Metz, Louis A Gilula, Christian Gaebler, Christian Kukla, Dominik Fleischmann, Herwig Imhof, and Siegfried Trattnig. "Radiographically occult scaphoid fractures: value of MR imaging in detection." In: *Radiology* 203.1 (1997), pp. 245–250.

[198] AD De Zwart, FJP Beeres, D Ring, LM Kingma, EG Coerkamp, SAG Meylaerts, and SJ Rhemrev. "MRI as a reference standard for suspected scaphoid fractures." In: *The British Journal of Radiology* 85.1016 (2012), pp. 1098–1101.

[199] Carl Doersch. "Tutorial on variational autoencoders." In: *ArXiv Preprint ArXiv:1606.05908* (2016).

[200] Max Welling and Diederik P Kingma. "Auto-encoding variational bayes." In: *ICLR* (2014).

[201] Tim Salimans, Diederik Kingma, and Max Welling. "Markov chain monte carlo and variational inference: bridging the gap." In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1218–1226.

[202] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models." In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1278–1286.

[203] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. "Deep convolutional inverse graphics network." In: *Advances in Neural Information Processing Systems* 28 (2015).

[204] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. "Draw: a recurrent neural network for image generation." In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1462–1471.

[205] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models." In: *Advances in Neural Information Processing Systems* 28 (2015).

[206] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. "An uncertain future: forecasting from static images using variational autoencoders." In: *European Conference on Computer Vision*. Springer. 2016, pp. 835–851.

[207] Nikolaus Kriegeskorte and Pamela K Douglas. "Interpreting encoding and decoding models." In: *Current opinion in neurobiology* 55 (2019), pp. 167–179.

[208] Aladin Virmaux and Kevin Scaman. "Lipschitz regularity of deep neural networks: analysis and efficient estimation." In: *Advances in Neural Information Processing Systems* 31 (2018).

[209] Diederik P Kingma, Max Welling, et al. "An introduction to variational autoencoders." In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[210] Alexei A Efros and William T Freeman. "Image quilting for texture synthesis and transfer." In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 2001, pp. 341–346.

[211] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. “Image analogies.” In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. 2001, pp. 327–340.

[212] Antoni Buades, Bartomeu Coll, and J-M Morel. “A non-local algorithm for image denoising.” In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 2. IEEE. 2005, pp. 60–65.

[213] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. “Sketch2photo: internet image montage.” In: *ACM Transactions on Graphics (TOG)* 28.5 (2009), pp. 1–10.

[214] Saining Xie and Zhuowen Tu. “Holistically-nested edge detection.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1395–1403.

[215] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. “Image-to-image translation with conditional Adversarial Networks.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1125–1134.

[216] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. “Context encoders: feature learning by inpainting.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2536–2544.

[217] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets.” In: *ArXiv Preprint ArXiv:1411.1784* (2014).

[218] James M Joyce. “Kullback-leibler divergence.” In: *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 720–722.

[219] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. “High-resolution image synthesis and semantic manipulation with conditional GANs.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8798–8807.

[220] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. “Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks.” In: *ArXiv Preprint ArXiv:1704.00849* (2017).

[221] Santiago Pascual, Antonio Bonafonte, and Joan Serra. “SEGAN: speech enhancement generative adversarial network.” In: *ArXiv Preprint ArXiv:1703.09452* (2017).

[222] Takeru Miyato and Masanori Koyama. “CGANs with projection discriminator.” In: *ArXiv Preprint ArXiv:1802.05637* (2018).

[223] Augustus Odena, Christopher Olah, and Jonathon Shlens. "Conditional image synthesis with auxiliary classifier GANs." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2642–2651.

[224] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2223–2232.

[225] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. "Toward multimodal image-to-image translation." In: *Advances in Neural Information Processing Systems* 30 (2017).

[226] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. "Semantic image synthesis with spatially-adaptive normalization." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2337–2346.

[227] Reda Oulbacha and Samuel Kadoury. "MRI to CT synthesis of the lumbar spine from a pseudo-3D cycle GAN." In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2020, pp. 1784–1787.

[228] Chuan Li and Michael Wand. "Precomputed real-time texture synthesis with markovian generative adversarial networks." In: *European Conference on Computer Vision*. Springer. 2016, pp. 702–716.