UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Grado en Ingeniería de la Salud

Superresolución de imágenes de rodilla mediante modelos de Aprendizaje Profundo

Using Deep Learning models for super-resolution knee imaging

Realizado por

Álvaro Rey Blanes

Tutorizado por

Enrique Domínguez Merino

Departamento

Lenguajes y Ciencias de la Computación

MÁLAGA, febrero de 2024

UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA DE LA SALUD

Superresolución de imágenes de rodilla mediante modelos de Aprendizaje Profundo

Using Deep Learning models for super-resolution knee imaging

Realizado por
Álvaro Rey Blanes

Tutorizado por
Enrique Domínguez Merino

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, FEBRERO DE
2024

Fecha defensa: marzo de 2024

Resumen

En el ámbito de la salud y sanidad moderna, las pruebas de imagen juegan un papel fundamental, proporcionando información vital para el diagnóstico y tratamiento de pacientes. Entre estas técnicas, las imágenes por resonancia magnética (IRM) destacan por su capacidad para visualizar el interior del cuerpo humano utilizando campos magnéticos. El avance en la informática ha permitido mejorar la eficiencia y calidad de las imágenes obtenidas por las resonancias magnéticas en instalaciones médicas. Por ello durante este proyecto se ha investigado y desarrollado en el área de procesamiento de imágenes por resonancia magnética, con un enfoque particular en el uso de modelos de redes convolucionales para mejorar la resolución de estas imágenes. Se han propuesto dos modelos diferentes, junto con la implementación de uno, repartidos en 6 experimentos diferentes en total. Posteriormente se ha evaluado sus resultados a través de métricas objetivas, tanto de manera interna entre experimentos, como de manera externa con otros modelos propuestos por investigadores. A su vez, mediante la valoración subjetiva de profesionales médicos y estudiantes relacionados con la medicina, dando una valoración más visual y específica sobre la anatomía en los resultados. Esta aproximación busca optimizar el uso de las resonancias magnéticas, mejorando así la precisión diagnóstica y el tratamiento de los pacientes tratando de facilitar y reducir riesgos asociados a la resolución de las pruebas diagnósticas.

Palabras Clave: Superresolución, Resonancia Magnética, Aprendizaje Profundo, Redes Convolucionales

Abstract

In the context of modern health and healthcare, imaging tests play a crucial role, providing vital information for the diagnosis and treatment of patients. Among these techniques, magnetic resonance imaging (MRI) stands out for its ability to visualize the interior of the human body using magnetic fields. Advances in computing have enabled improvements in the efficiency and quality of images obtained from MRI scans in medical facilities. Over this final year dissertation, research and development have been conducted in the area of magnetic resonance imaging processing, with a particular focus on the use of convolutional neural network models to enhance the resolution of these images. Two different models have been proposed, along with the implementation of one, spread across a total of six different experiments and evaluating their results both through objective metrics internally among experiments, as well as externally with other models proposed by researchers. Additionally, through the subjective assessment of medical professionals and students related to medicine, providing a more visual and specific evaluation of the anatomy in the results. This approach seeks to optimize the use of MRIs, thus improving diagnostic accuracy and patient treatment by aiming to facilitate and reduce risks associated with the resolution of diagnostic tests.

Keywords: **Super-resolution, MRI, Biomedical Engineering, Convolutional Neural Networks, Deep Learning**

Contents

List of Figures

List of Tables

1 Introduction

1.1 Magnetic Resonance Imaging

MRI is an imaging technique, used mainly in the biomedical field and it is based on the physical phenom known as magnetic resonance (MR), which allows us to look into those particles that possesses spin and charge, given a magnetic field. It is different from other techniques as X-Ray (Rx) or computer tomography (CT), as it does not use ionizing radiation but radio-frequency (RF) signals, meaning MRI is a harmless method without losing information quality. All in all, with the computational processing and improvement and due to its utility to almost represent any anatomical part of the body (apart from the knee) makes it an useful technique extended worldwide.

1.1.1 Magnetic Resonance

It was first discovered by Isidor Isaac Rabi, a well-known physicist born in Poland, in 1938 after almost a decade of studying the magnetic fields and extended with the knowledge in nuclear magnetic resonance (NMR) in 1946 by Felix Bloch and Edward Purcell. We can define its principle as: when an atom (with an odd atomic number Z) is affected by a magnetic field $\overrightarrow{B_0}$, constant over time, this would be able to absorb the energy at a specific frequency (which may vary over time). This would cause what we know as nuclear spin, which in case of the ^{1}H (most used choice for MRI, as the body-tissue contain mostly H_2O) has a spin of $\frac{1}{2}$.[1]

This nuclear spin is the reason for the magnetic moment, induced around the nucleus and aligned parallelly to the rotation axis. Magnetic moment will remain constant as the nuclear spin. In order to investigate or register an MR it is needed a sum of spins on behalf of a single spin.

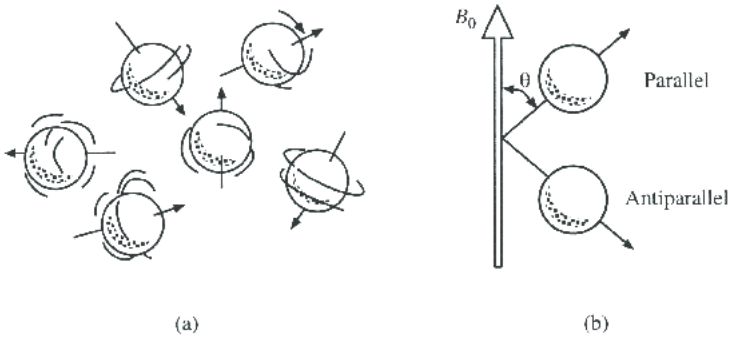


Figure 1.1.1: Illustration of how nuclear spin works, and if it is parallel (+ spin) or anti-parallel (- spin) The orientation angle θ = 54.7° for ^{1}H a) Random orientation due to no magnetic field. b) Organized orientation under a magnetic field. *Source: Multi subject study of motor sensory function in rat brain using fMRI (2005) John Lilja.* [2]

1.1.2 Larmor formula

Apart of the magnetic moment previously mentioned, the magnetic field will produce another spin (precession) around the field.

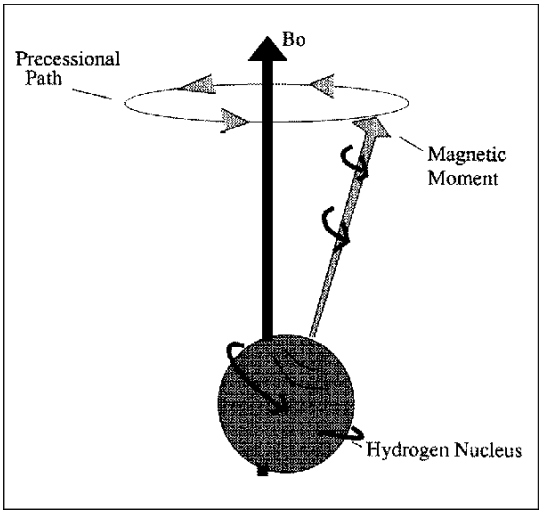


Figure 1.1.2: Illustration of how precessional path works along a magnetic moment. *Source: "HOSPITAL PHYSICS: Magnetic resonance imaging and spectroscopy" (1996) Andy Simmons . [3]*

The frequency associated to this precessional spin can be calculated using the Larmor equation,

$$\omega = \gamma B_0 \tag{1.1.1}$$

where ω is given by,

$$f = \frac{\omega}{2\pi} \tag{1.1.2}$$

ω is the known as "Larmor frequency" (MHz), B the value of the magnetic field (T) and finally γ which is the gyro-magnetic ratio ($\frac{MHz}{T}$), that will remain constant. For our interest, as we have indicated before, for the isotope ^{1}H, γ = 42,58 $\frac{MHz}{T}$.

There will be an added magnetic field, related to close by electrons, which may make the external field ($\overrightarrow{B_0}$) increase or decrease. So, the equation 1.1.1 might be rewritten as,

$$\omega = \gamma(1 - \sigma)B_0 \tag{1.1.3}$$

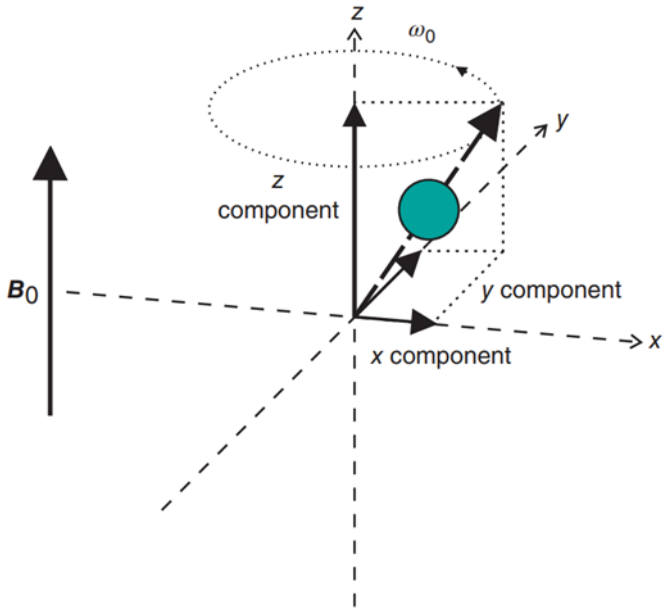


Figure 1.1.3: Visual definition on how the Larmor equation works. *Source: "MRI Basic Principles and Applications", (2015)* [1]l

Larmor frequency will vary between 42 to 170 MHz, which is inside the RF range, and this is basically the reason MRI does not affect living tissue, as it is much lower than Rx. [4]

1.1.3 Creating the MRI

Using the physics studied before, and applying a specific magnetic field which values are usually in the 0.5T (minimal value) to 11.7T (mainly used in research scanners) range and with an added $\overrightarrow{B_1}$ (static at a rotation frame) which will be created using radio-frequency transmitters. The precessing between x and y equals to 0 and $\overrightarrow{M}$ (vector sum) is static, before there is a RF pulse. After the RF pulse, it will no longer cancel, creating a preccesing M_t and a voltage with a Larmor frequency oscillation. This causes the nucleus to go at a maximum energy state, $\overrightarrow{M}$ to follow a spiral movement and divert α. [5]

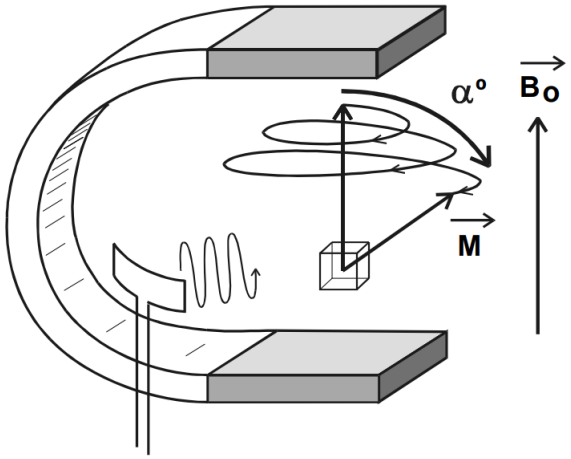


Figure 1.1.4: Visual explanation on how the momentum vector enters on a spiral movement after the RF Pulse. *Source: Introduccion biofisica a la resonancia magnetica (1993), Jaume Gili [6]*

After the excitation reaches its top, the pulse comes to an end and $\overrightarrow{M}$ goes to relaxation, recovering its initial position. The nucleus stops being excited (figure 1.1.5) and will come gradually to the lower energy state energy. This causes ΔE, and each voxel [1] will release an energetic signal (RF) which after being received and processed will be converted to an electric signal. Finally using Fourier inversion theorem (equations 1.1.4 & 1.1.5), [7]

[1] Voxel, as defined by Oxford Languages "each of an array of elements of volume that constitute a notional three-dimensional space, especially each of an array of discrete elements into which a representation of a three-dimensional object is divided." In MRI we use voxels to discretize the patient volume.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{1.1.4}$$

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{-i\omega t} dt \tag{1.1.5}$$

where, t equals the time, ω the frequency that has been already defined, and f(t) the sinusoidal signal, and using FT we will be able to obtain the planes making up the final MRI result.

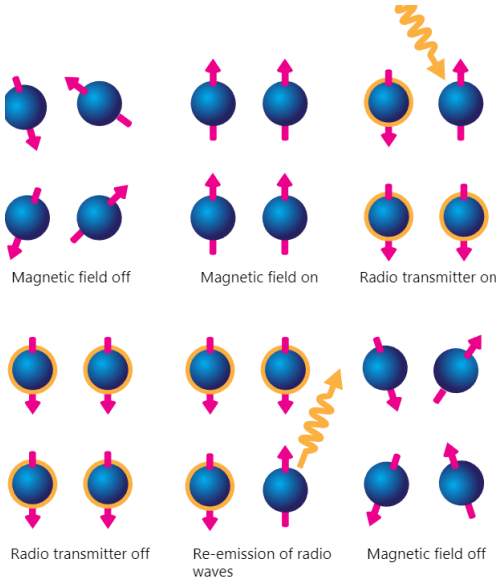


Figure 1.1.5: Nuclear alteration after the RF pulse goes off. *Source: MCAT, Khan Academy.*

1.1.4 MRI Weights

Depending on our medical intentions, MRI equipment can be configured in order to show different anatomical parts. This weights are used by working on the relaxation properties and we can distinguish: [8]

- **T1-Weighted MRI (T1)**: This weight uses a time constant given by the return to equilibrium of longitudinal magnetization. It has a great soft tissue contrast and can be useful on detecting vascularity of injuries. In addition, shorter T1 will be brighter. Longitudinal magnetization at a time t, $M_z(t)$, will be given by:

$$M_z(t) = M_0 \left(1 - e^{-\frac{t}{T_1}}\right) \tag{1.1.6}$$

  Being M_0 an initial magnetization and $T1$ the relaxation time.

- **T2-Weighted MRI (T2)**: T2-weighting relies on the relaxation time constant, specifically on the transverse magnetization affected by neighboring spins interactions. Commonly employed for evaluation anatomical structures like meniscus, ligaments, or tendons in knee imaging, T2-weighted MRI highlights brighter signals with longer T2 relaxation times, contrasting with T1-weighted imaging. Transverse magnetization on the x,y axis at t, $M_{xy}(t)$ is calculated by:

$$M_{xy}(t) = M_0 \left(1 - e^{-\frac{t}{T_2}}\right) \tag{1.1.7}$$

- **Proton Density Weighted MRI (PD)**: As its name indicates, it shows the hydrogen protons concentration on the tissue. The contrast will be given by variations on proton density. It can also differentiate ligaments or meniscus, meaning it is pretty common to use PD-weights on a knee MRI.

- **Short Tau Inversion Recovery MRI (STIR)**: This technique is useful to suppress signal from fat, which is usually unneeded on a knee study. The equation giving signal intensity S is:

$$S_{STIR} = \rho \left(1 - 2e^{-\frac{T_I}{T_{1\text{fat}}}} + e^{-\frac{T_R}{T_1}}\right) e^{-\frac{T_E}{T_2}} \tag{1.1.8}$$

  With TR and TE being repetition and echo time respectively. [8]

1.2 The Knee joint

The knee is classified as an hinge joint (inside function classification), being the most complex one in the human body, as it contains the widest range of extension and flexion among all sagittal plane joints. Its complexity also comes from the rotation movement which the knee is also capable of, but the range of rotation depends directly on the flexion degree the knee has. It can be more specifically classified as a synovial joint which means it contains a synovial cavity filled with synovial fluid and a membrane (synovium).
As an anatomical structure, the knee holds around 70% of body-weight while standing in bipedal position, allows the movement and provides great stability and shock absorber between bones. [9]

1.2.1 Knee anatomy

The knee is built up by different types of tissues.

Bone Tissue is a *specialized connective tissue* present on the knee precisely in the bones that make up the joint. Its main compounds are collagen type I (organic portion of the bone) in approximately 90% distributed in parallel fibrillar bundles, which contains substances as hyaluronic acid and keratin (proteoglycans), along osteonectin and osteopontin. The other portion is the inorganic, composed by hydroxyapatite. Bone's functions are to give support and protection, and are attached to each other by ligaments. The bones present in the articulation are:

- **Femur**: Distal femur (Lateral condyle, Medial condyle). Patellar surface. Articulates with patella and tibia.
- **Patella**. All parts of the bone are contained in the knee. Articulates with femur and tibia
- **Tibia**: Proximal tibia (Lateral condyle, Medial condyle). Articulates with femur, fibula and patella
- **Fibula**: Its apex head is an insertion side to ligaments and muscles of the knee.

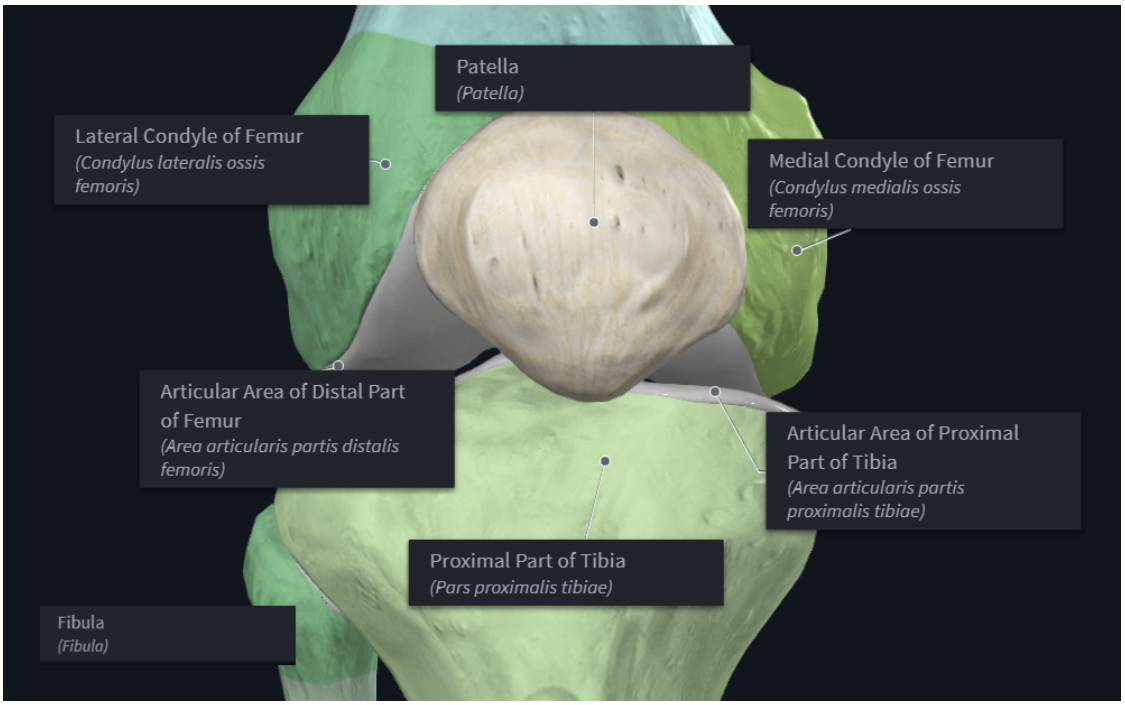


Figure 1.2.1: Bones in knee labeled. *Source: Complete Anatomy, 3D4Medical from ELSEVIER.* [10]

Ligaments and tendons (figures 1.2.2 & 1.2.3) which are *regular dense connective tissue*, meaning their fibers are parallel-oriented to resist high level of tension on a determined direction. Ligaments are connective tissues that join two different bones whereas tendons connect a muscle with a bone. The main fibers component is collagen type I as well, which packed together forms collagen fibrils.

There are numerous ligaments in the knee but focus is usually set on the **four main**, as they are the most critical in order to stabilise the knee, those ligaments are: [11]

- **ACL - Anterior cruciate ligament**: Femur-tibia. Prevents a knee hyper-extension as it does not allow the femur to slide over the tibia in a sagittal plane (front/backwards).

- **PCL - Posterior cruciate ligament**: Femur-tibia. Stronger and thicker than the ACL and with its same function (stabilize, not allowing bone sliding) .

- **LCL - Fibular/Lateral collateral ligament**: Femur-fibula. Forms a co-joint tendon with the biceps femoris. As the other ligaments described, its function is to stabilize the knee, but in this case in the coronal plane (sideways, to the outer side).

- **MCL- Tibial/Medial collateral ligament**: Femur-tibia (fused to medial meniscus). Stabilizes the knee as the LCL, in a coronal plane (inner side) but its connection to the meniscus increases the injury possibility.

Furthermore, there are more ligaments with fewer importance than the mentioned before:

- **Patellar ligament**: Tibia-patella.
- **Anterior and Posterior meniscofemoral ligament**: Gives support to the cruciate ligaments ACL and PCL.
- **Arcuate popliteal ligament**: Fibula-femur. Prevents knee excessive rotation.

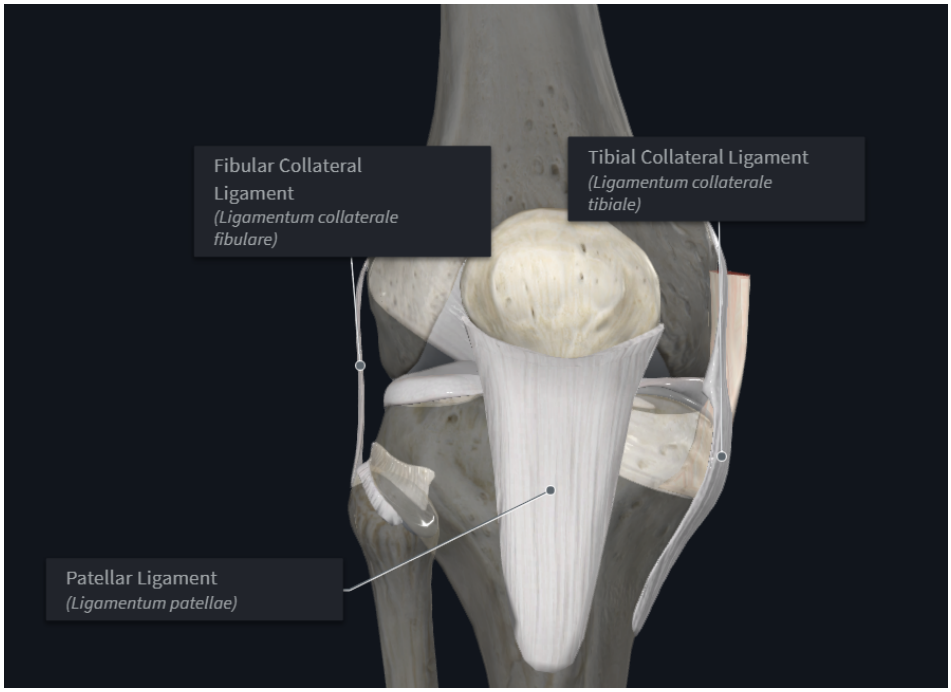


Figure 1.2.2: Anterior coronal view of knee ligaments. *Source: Complete Anatomy, 3D4Medical from ELSEVIER.* [10]

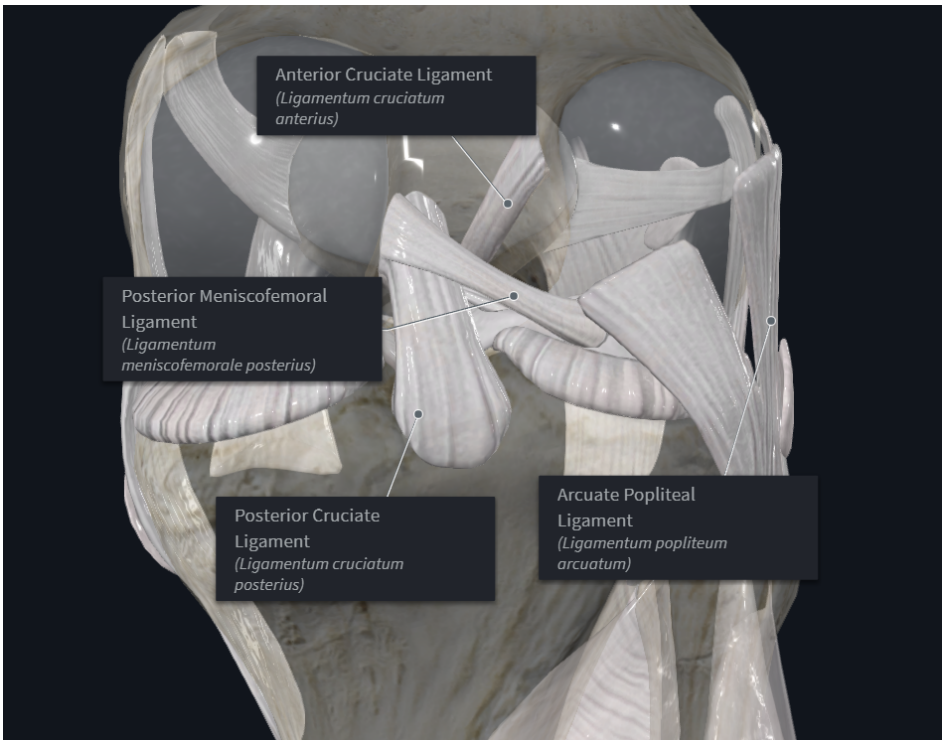


Figure 1.2.3: Posterior coronal view of knee ligaments *Source: Complete Anatomy, 3D4Medical from ELSEVIER.* [10]

When it comes to *tendons* (figure 1.2.4) we must highlight:

- **Quadriceps tendon**: Joins to the patella. Essential for leg extension.
- **Patellar tendon**: Is the same that the patella ligament as it also joins two bones, but at the same time it is a continuation for the quadriceps tendon. From now on, we will refer to it as a tendon following the proper medical terminology for injuries and treatments.
- **Semitendinous muscle**.
- **Iliotibial tract/band**. This tendon goes from the gluteus to the tibia.
- **Biceps Femoris Tendon**. Works on the knee flexion.

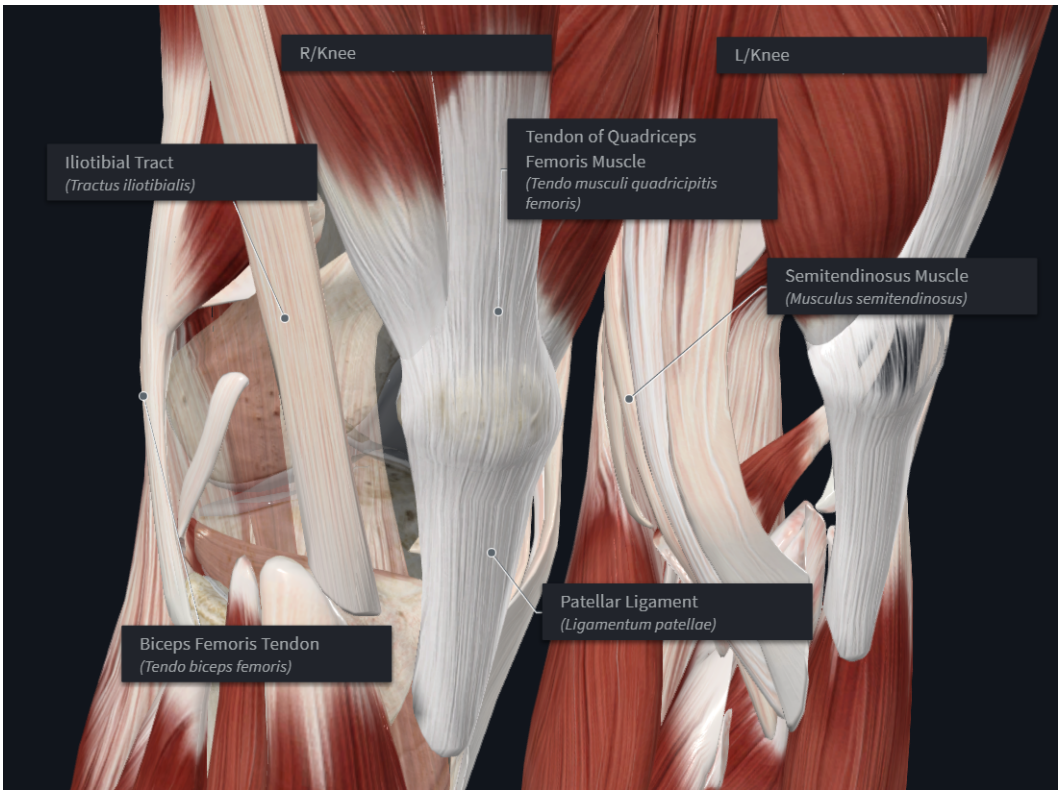


Figure 1.2.4: Anterior coronal/sagittal view of knee tendons *Source: Complete Anatomy, 3D4Medical from ELSEVIER.* [10]

Meniscus, classified as fibrocartilage, is made of Collagen Type I and II (figure 1.2.5). It is key to reduce knee wear, distribute the forces transmitted on a normal knee usage (transforming axial compression forces to circumferential-tensile) and lubricating the articulation. It is consider almost an avascular and acellular anatomical structure, making it one of the hardest tissues to regenerate. [12] We can distinguish two inside each knee:

- **Medial meniscus**: Located in the inner part of the knee, C-shaped. Wider in its posterior part than in the anterior part and attached to **MCL**.
- **Lateral meniscus**: Located in the outer part of the knee, close to an O-shaped structure. Its wideness is uniform. Has a bigger mobility than the medial meniscus.

Articular Cartilage which is composed of hyaline articular cartilage and has no perichondrium nor blood vessels, leading to a complex regeneration. It has superficial, medial and deep zones [13], varying their collagen type II direction and density in function of the amount/orientation of forces. [14]

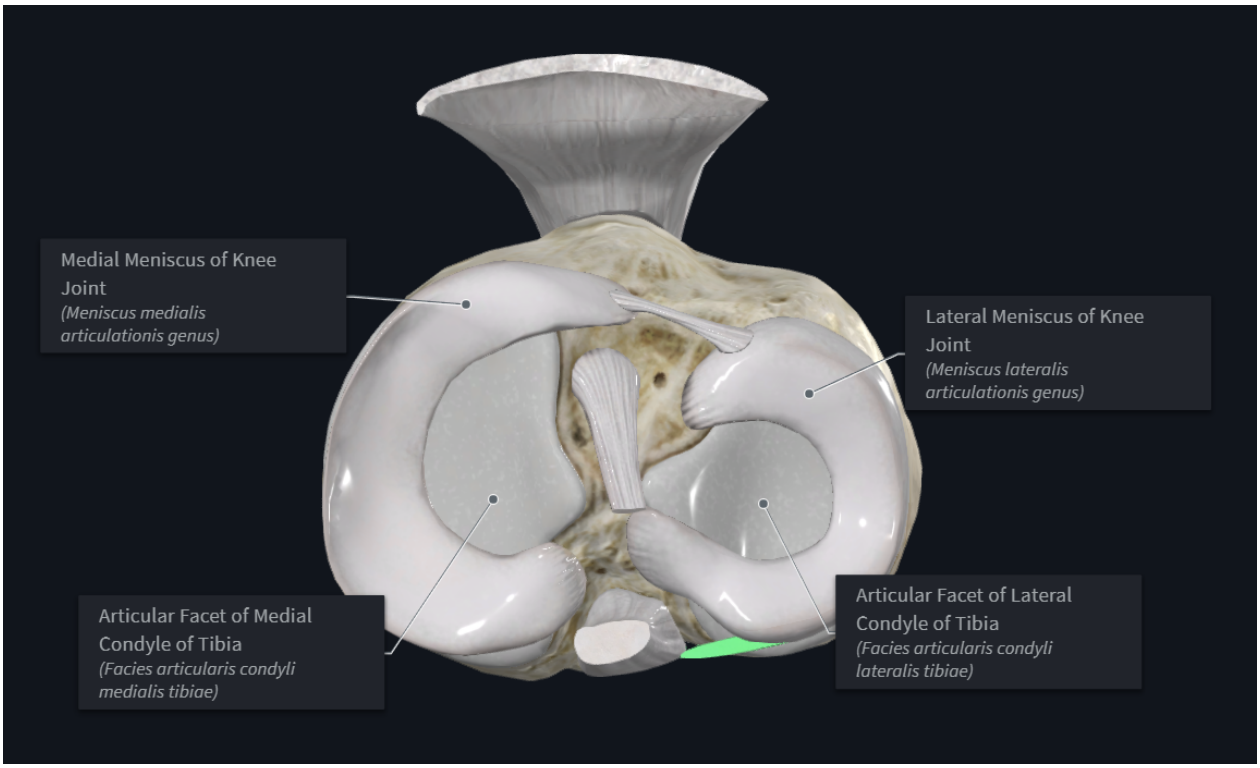


Figure 1.2.5: Axial view of knee meniscus and Articular Cartilage placement *Source: Complete Anatomy, 3D4Medical from ELSEVIER.* [10]

Muscular tissue, is a completely different type of tissue than the connective tissue mentioned before. *Striated (skeletal) muscle* is the specific type of muscle involved in the knee, although most of them are not placed in the knee but in its proximal side (upper leg, closer to the trunk) or in its distal side (lower leg, further to the trunk). However they do have, as previously discussed, their tendons attached to parts of the knee. This means these muscles are responsible of articulating and providing stability to the joint (the main function of muscular tissue is to produce forces, by shortening or extending fibers). In addition they also play a key role to prevent possible damage or injuries in the tissues located in the knee [15]. The main muscles related are:

- **Quadriceps femoris**: Consisting of 4 specific muscles (vastus lateralis, vastus intermedius, vastus medialis and rectus femoris). Those are knee extenders in the proximal part of the leg.
- **Hamstrings**: Consisting of 3 specific muscles (semitendinosus, semimembranosus, biceps femoris). Are knee flexors and the biggest muscle group along with the quadriceps

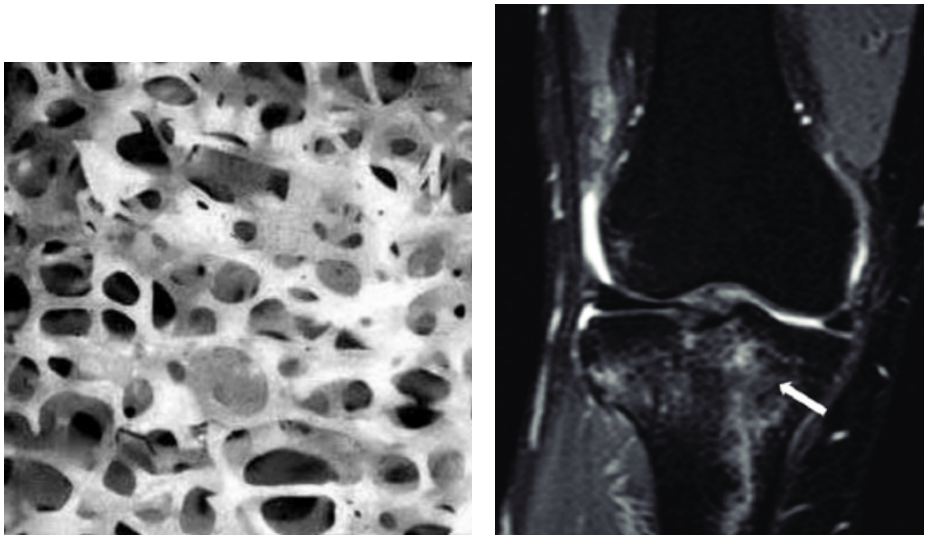
femoris to actuate in the join.

- **Sartorius**: Crosses the knee on its medial side, to attach the proximal tibia, being the longest muscle in the human body. Main function is to rotate the joint when the knee is flexed and also helps hamstrings on the knee flexion.
- **Popliteus**: Goes from the medial/anterior proximal tibia to medial distal femur, crossing the knee on its posterior side. Helps as well to the knee rotation (medially) while flexed and femur rotation on the tibia.
- **Gastrocnemius**: Medial/lateral, mostly known as calf. Similar to what popliteus muscle does, crosses through the posterior part of the knee, this time to join the distal head of the femur. Even though its main function is related to the ankle join, it helps on the knee flexion.
- **Adductor magnus**: Although is not related to any knee movement, it is attached to the distal femur and helps on the stability of the articulation.

1.2.2 Injuries in the knee

The main usage of the MRI in the knee is to detect and evaluate possible damage done to the tissues. While almost all injuries related to ligaments or cartilage/meniscus needs a complementary MRI to observe the status of a malfunctioning tissue, bones and muscles can also be precisely evaluated with other imaging techniques, requiring MRI in some specific trauma cases. [15] [16]

Bone: Usually, the most common damage a bone can have is a fracture (bone crack that goes through) or fissure (just broken on the surface). In that situation, the diagnose would be done using *Rx* or *CT*. In spite of that, there are medical situations (figure 1.2.6) that requires of a MRI to evaluate further, as it can not be seen in the previous mentioned techniques. Bone trabeculae fractures[2] or an unusual cellular growth are examples. [17]



(a) (b)

Figure 1.2.6: a) Bone trabeculae. b) PD MRI with a marrow oedema. *Source: a)Tissue Engineering, University of Malaga, Andrades,J.A. B)Chemical shift imaging: An indispensable tool in diagnosing musculoskeletal pathology's (May 2021, Vandana,J.)* [18]

[2]Bone trabeculae are structures placed on the cancellous bone, which is placed on the extremes of a bone. Are made of interstitial lamellae, containing osteocytes inside and osteoblast, osteoclast on the outside

Ligaments and tendons, must be the anatomical part most in need of MRI, in order to confirm or dismiss an injury. The case of ligaments partial/complete tears are the most significant trauma, as time taken to recovery goes from a month (grade 1 tear, only visible on a microscopy and unable to see on a MRI) to 12+ months (complete rupture). Some specific cases:

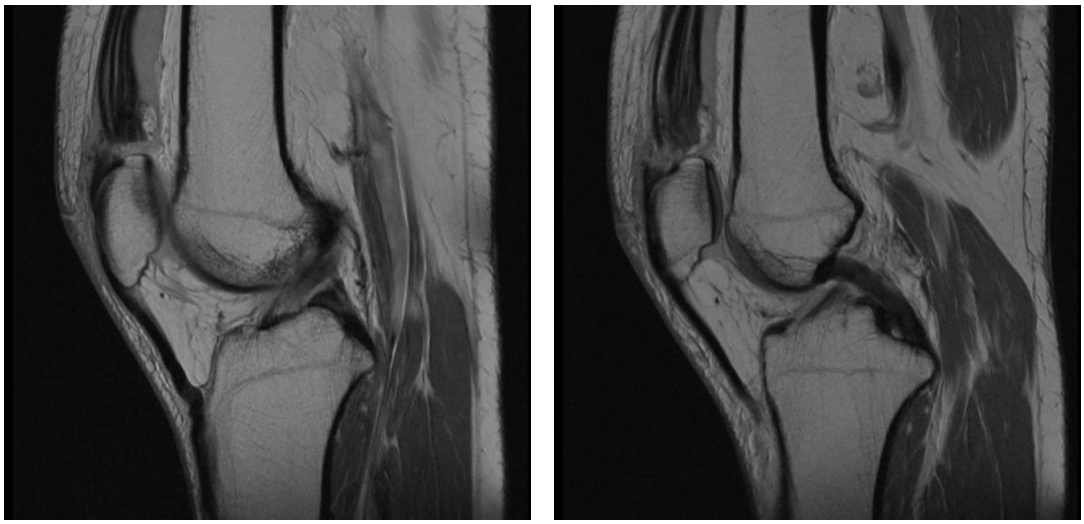
- **ACL/PCL rupture**: Being the ACL tear (figure 1.2.2) much more common, obtaining good quality images are essential in order to choose and select a treatment/recovery plan. This injuries are usually repaired via surgery and requires from a graft [19], usually autografts from patellar/hamstrings/quadriceps tendons or even the iliotibial band, all discussed in the previous subsection 1.2.1. A good image also helps to create a surgery plan, and evaluate the possible anatomical part to use for an autograft.
- **MCL/LCL**: MCL damage represents approximately 40% of all knee ligaments injuries and also much more common than LCL torn. Its diagnose and treatment is quite similar as the one seen before.
- **Meniscofemoral/arcuate ligaments** injuries are also required of a MRI to evaluate, however this ligaments are pretty much less common to suffer from a tear.[20]

(a) (b)

Figure 1.2.7: (a) ACL tear seen in a PD MRI. (b) ACL tear seen on a T2 MRI. Both using a sagittal view *Source: Case courtesy of Hani Makky Al Salam, Radiopaedia.org, rID: 12175* [21]

For tendons, tears (figure 1.2.8) are also the most common injury in order to need a medical image diagnose. In addition, tendons may suffer from inflammations or degenerative processes which causes the known as tendinopathy or tendinitis. Specific cases:

- **Patellar/quadriceps tendon tendinopathy**. Might be treated without need of a surgical procedure, rest and region-specific temperature changes.
- **Patellar/quadriceps tendon rupture**. Is more likely to need surgery and in case, suture (stitches and anchors) between the broken parts or an autograft.[22]



(a) (b)

Figure 1.2.8: Quadriceps tendon tear on a sagittal view. The rupture can be seen as the black line discontinuity on the upper left-hand side of the image. PD MRI. *Source: Case courtesy of Dr.Ajay C Desai, Radiopaedia.org, rID: 10826* [21]

Meniscus and articular cartilage: Besides ligaments and tendons, the other main tissue which damage can be precisely seen on a MRI.
Meniscus injuries, can be caused due to trauma (knee ligaments tears can also cause damage due to the released energy and attachments) or degenerative due to constant pressure and damage along years. The meniscal tears can be complex, as there are different types as showed on figure 1.2.9

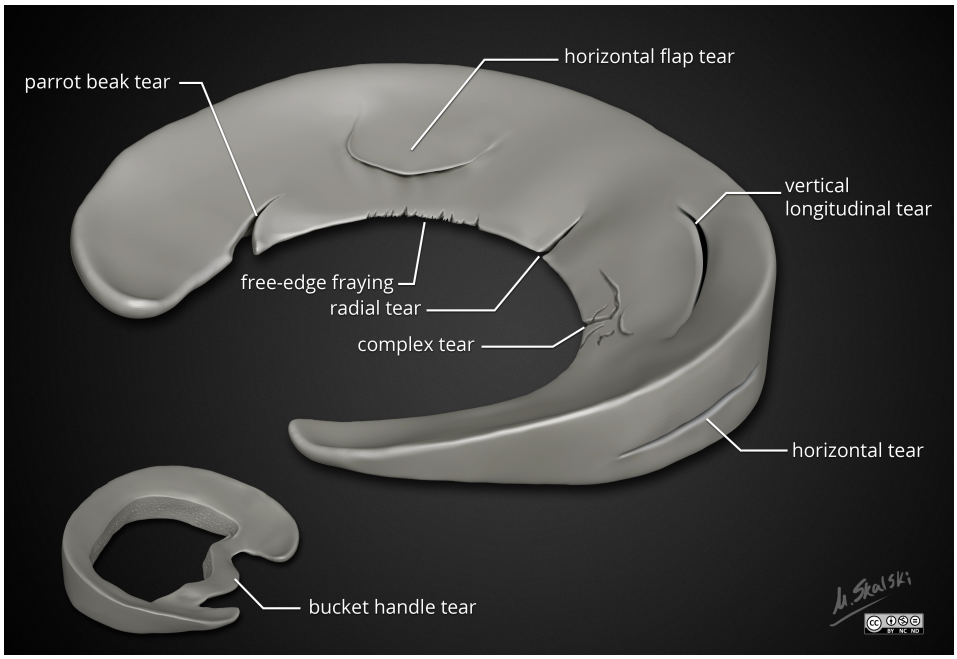


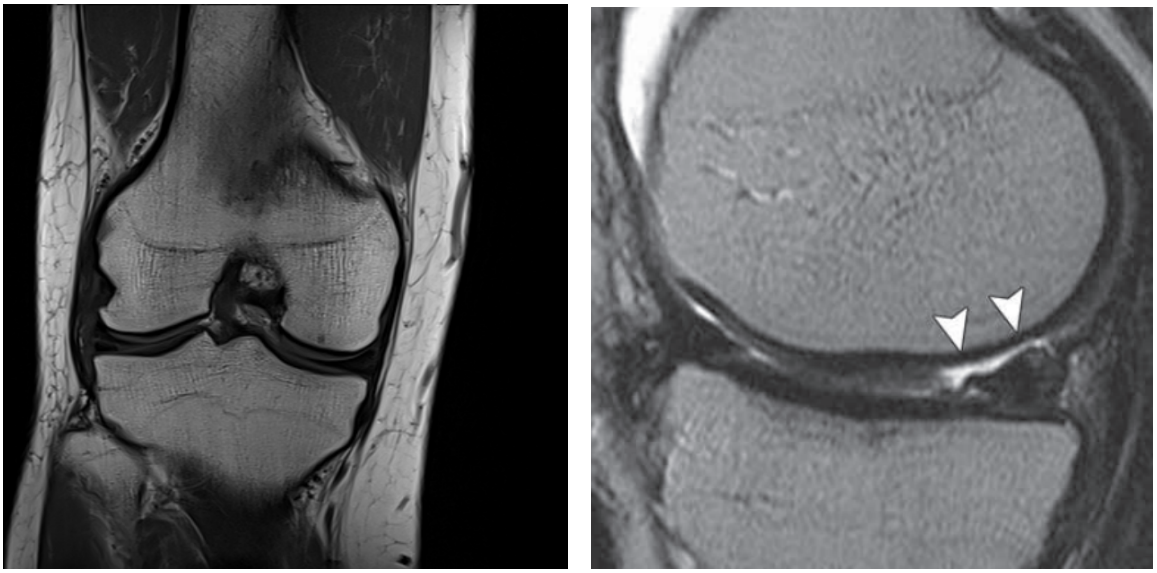
Figure 1.2.9: Different meniscal tears. *Source: Case courtesy of Matt Skalski, Radiopaedia.org, rID: 55569* [21]

Torn or degenerated meniscus usually causes a big pain in the patient, the inability to articulate the knee (locked feeling) or stiffness articulation. Additionally, as previously said, the meniscus is a very-low vascular and very-low cellular tissue, causing that conservative treatments are rarely used and that surgery is usually a better option (which is in need of MRI to asses the current state of the tissue before incise).

This type of injuries are also approachable with 3D-MRI sequencing or MRI arthrography (usage of a contrast agent to improve conventional MRI).

Articular cartilage damage (figure 1.2.10), is quite similar to meniscus injuries in symptoms, joint limitation and professional check. It can be graded using *Outerbridge classification of cartilage, 1961* from grade 0 (no apparent damage, cartilage is on a good condition) to grade 4 (no remainder cartilage, bone is completely exposed).

Muscles injuries, mainly caused by overuse or an excessive muscle extension breaking the muscular fibers. Muscles, just as ligaments and tendons, can suffer a tear (figure 1.2.11). They can also be injuries related to an inflammatory nature (myopathies) or contusions induced by trauma. All this events can be evaluated with ultrasound, being the MRI useful in deeper injuries or tendon-related muscles injuries.[23]



(a) (b)

Figure 1.2.10: (a)Coronal T1 MRI, showing complex horizontal and radial tears. (b)Sagittal T2 MRI with a grade 4 AC damage. *Source: Case courtesy of Ahmed Abdrabou, Radiopaedia.org, rID: 33036* [21] *(a)- Articular Cartilage in the Knee: Current MR Imaging Techniques and Applications in Clinical Practice and Research (2011). Crema, Michel D. RNSA (b)* [14]

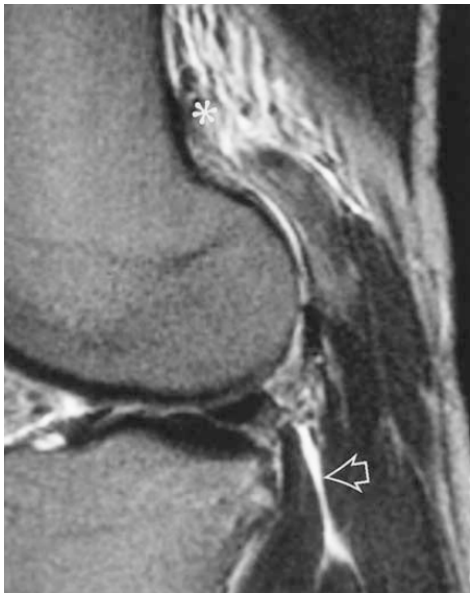


Figure 1.2.11: Sagittal T2 MRI, with a partial gastrocnemius tear(*). *Source:Traumatic Musculotendinous Injuries of the Knee: Diagnosis with MR Imaging (2000). Bencardino JT. Rafiographics.* [24]

1.3 Super-resolution and Deep Learning

1.3.1 Super-resolution

Super-resolution as a group of techniques to increase an image resolution, has been used for almost half a century now. However, it has been increasing over the recent years due to the computing improving and accessibility in conjunction to the newest technologies both in hardware and software. Its main purpose is to construct high-resolution images ("HR") from low-resolution images ("LR") [25] in order to obtain finer details or in the specific medical field, better anatomical detail.

Algorithms in SR: Depending on the approach to enhance the image resolution we can discriminate into various classes of algorithms :

- **Interpolation-Based**: Bilinear/Bicubic interpolation. *Bilinear interpolation* uses the average of 2x2 of known pixels around the unknown ones. One of the simplest SR algorithms. *Bicubic interpolation* consist of the same but using 4x4 neighborhood, which gives improvement from the bilinear, with an higher computational cost.

- **Reconstruction-Based**: *Iterative Back-Projection (IBP)* consist on a continuous iteration comparing the downgraded image with the LR and uses this to correct the HR. *Maximum a posteriori (MAP)* uses an initially guess of HR to regularize the posterior solution.

- **Learning-Based**: *Convolutional Neural Network (CNN)*. Deep Learning models with end-to-end mappings. Starting with LR patches to end on a HR result. Examples are: **SRCNN** or **EDRS**. *Generative Adversarial Networks (GAN)*: Uses a generator to setup HR images and discriminator to evaluate the result. Examples are **SRGAN** or **ESRGAN**.

- **Patch-Based**: *Anchored Neighborhood Regression*, uses embedding and ridge regression of local neighbours to improve resolution. *Sparse Coding*: Patches can be represented as sparse linear combination of elements.

- **Frequency-Domain**: *Wavelet-Based*, applies the wavelet transforms to decompose images into different frequency bands then applies SR techniques.

- **Hybrid**: *Deep Learning and Classical Methods*, uses a combination of both, trying to exploit the main advantages of both.

Category	Algorithm	Results	CompCost [3]
Interpolation-Based	BilinearInt	Low resolution improvements	L
	BicubicInt	Smoother, works for minor artifacts	L
Reconstruction-Based	(IBP)	Better detail and resolution, reduces artifacts	M
	(MAP)	Higher-quality, great for edge preservation	M
Learning-Based	(CNNs)	High-quality results	H-VH
	(GANs)	High-quality results	H-VH
Patch-Based	(ANR)	Efficient, moderate upscaling	U
	Sparse Coding	Detailed texture preservation, varied image types	U
Frequency Domain	W-B SR	Multi-scale enhancement and edge preservation	M
Hybrid	DL+CM	Detailed but simple	M-H

Table 1.3.1: Brief summary of the key elements for the different options to super-resolute.

- L: Low computational cost.
- M: Moderate computational cost.
- H: High computational cost.
- VH: Very high computational cost.
- U: Variable computational cost, depends highly on algorithm configuration. For ANR and Sparse Coding, computational cost may vary on number of patches and their complexity to process.

[3] All computational calculations are approximate and depends on the investigator application of each algorithm.

1.3.2 Deep Learning

As defined by Oxford Languages, deep learning is ”a type of machine learning based on artificial neural networks in which multiple layers of processing are used to extract progressively higher level features from data”. Basically, its the usage of non-linear transformation to identify and extract certain characteristics of specific data. We can differentiate on two main groups of machine learning:

- **Supervised Learning**: Systems learning data already given contains solution or labeled classes.
- **Unsupervised Learning**: Data used to learn does not contain an specific answer so learning is done based on data and criteria.

Inside proper deep learning (DL) algorithms we usually encounter the Artificial Neural Networks (ANN) which as it indicates, tries to imitate the nervous system neural configuration.

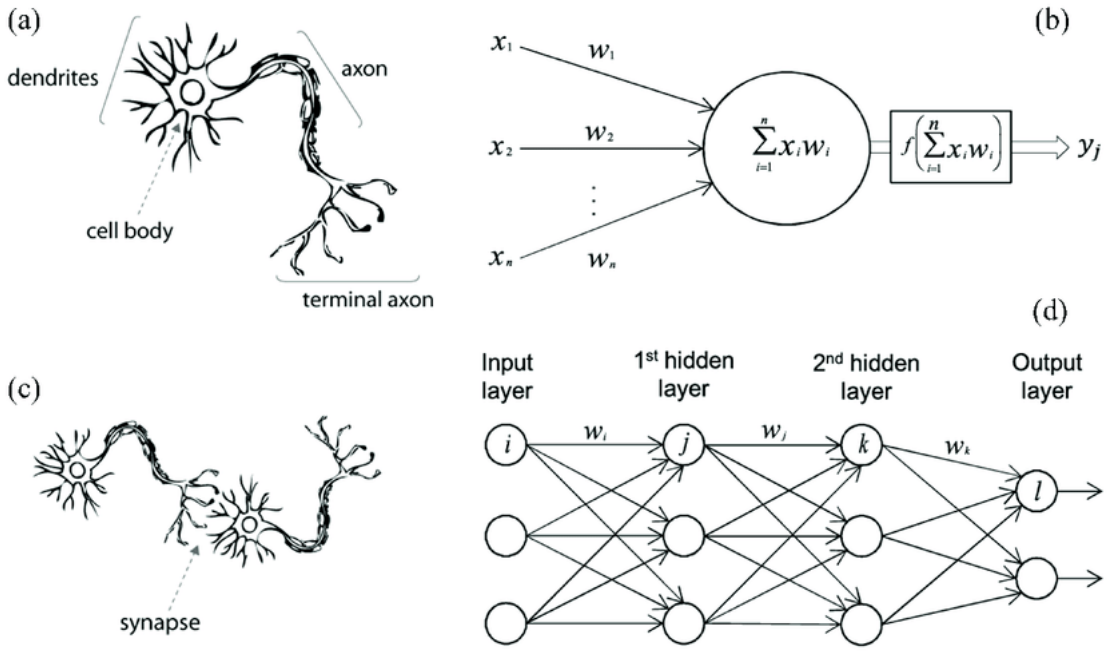


Figure 1.3.1: a) A biological monopolar neuron example and how it relates with other neurons to form a system c). b) Is an artificial neuron, d) shows ANN schema. *Source: Using a Data Driven Approach to Predict Waves Generated by Gravity Driven Mass Flows. (2020). Zhenzhu Meng* [26]

Artificial Neuron, as we have seen in figure 1.3.1, an AN is the basic structure of an ANN, and is composed by: [27]

- **Inputs and Weights**: Arriving values to the neuron, either from data input or other neuron. Weights are adjustable parameters, and constantly update to reduce network mistake.
- **Bias**: Following the input-weights, provides a degree of freedom so that the data can have a better fit.
- **Summation and Activation Functions**: The neuron calculates the sum of the inputs received and goes through a posterior activation function, which introduces non-linearity. This allow the AN and ANN to create complex links.
- **Output**: Can be an output for other AN, or to a final output layer of ANN.

We can mathematically model an neuron with the following equation:

$$a_k = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1.3.1}$$

where a is the k^{th} neuron, f , is the activation function, w_i, weight associated to i^{th} input, x_i, i^{th} input and b, bias.

Of all components, the main element is the *activation layer*, which apart from what was mention before, its crucial to process the input and produce an output data. Its key for the **decision making** in the ANN as it defines the type of output, **training efficiency** with some functions like *ReLu*[4] which are able to accelerate training process via gradient descent and for **complex patter learning**.

The most common activation functions that are used are:

[4]ReLu stands for Rectified Linear Unit.

Activation Function	Mathematical Expression	Usage
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	Probability/Binary Classification
Tanh	$f(x) = \tanh(x)$	Normalized Data
ReLU	$f(x) = \max(0, x)$	General Purpose, Hidden Layers
Leaky ReLU	$f(x) = \max(\alpha x, x)$	Dying ReLU Problem
Softmax	$f(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$	Multiclass Classification
ELU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}$	Improved Learning Characteristics
Swish	$f(x) = x \cdot \text{sigmoid}(\beta x)$	Self-Gated function

Table 1.3.2: Summary table of Main Activation Functions and theirs mathematical functions

As referred to in the previous subsection 1.3.1, the two main DL neural networks are CNN and GANs.

Convolutional Neural Networks - CNN As its name indicates, CNN consists of different convolutional layers or operations. Are adequately useful in topology grid-like tasks as images. We can distinguish 5 specific layers after an input:

- **Convolutional Layers**: Primary layer for a CNN, forming the core of the network. Each layer applies kernels (learnable filters) to the input. Usually extracts high-importance features (edges,textures).
- **Pooling Layers**: Goes after a Convolutional layer, reduce spatial dimensions. Most frequent pooling are Max pooling and average pooling.
- **ReLu Layers**: As we have discussed, ReLu activation functions consist on non-linear transformation which helps on solving the vanishing gradient. Reduces complexity and computing time.
- **Fully-Connected Layers**: Connects all activation of the previous layer. More usual on classification problems, at the end of the neural network model.
- **Normalization Layers**: Similar to batch normalization, useful to stabilize and improve training velocity.[28]

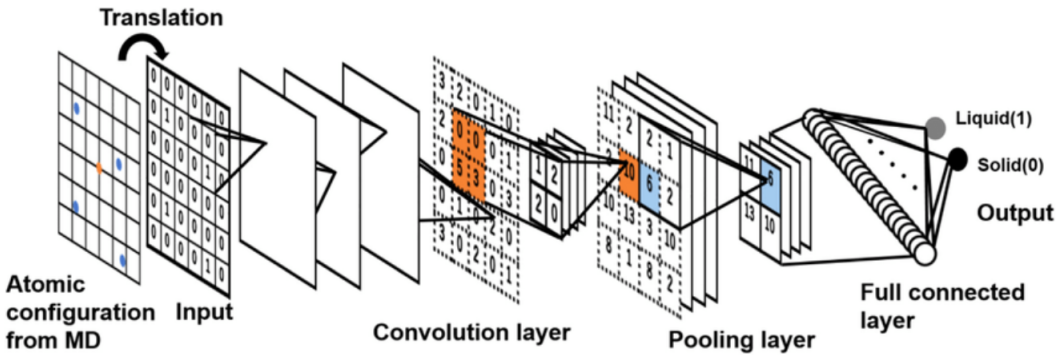


Figure 1.3.2: Example of a CNN schema. *Source: Machine learning approach to automated analysis of atomic configuration of molecular dynamics simulation (2020). Teppei Fukuya* [29]

Convolutional Autoencoders consist of a combination of autoencoders [5] and CNNs, creating a powerful while efficient implementation.

Autoencoders have, an **encoder**, which is the part of convolutional/pooling layers to progressively reduce the spatial dimensions of the input and compress the data into a latent-space representation. Captures the most salient characteristics on the input. **Latent space**, represents the compressed data ("the bottleneck"). Its the most crucial structure of a convolutional autoencoder as they determine the capacity to encode input variances. Finally the **decoder** parts, mirrors the encoder, applying convolutional and unsampling layers. The purpose is to reconstruct the input from the previous latent space layer. In essence, is a highly effective technique for image processing (reconstruction, generation or denoising).

Other known CNN variants/evolution are **LeNet**, **AlexNet**, **DenseNet** or the popular **GoogleNet**.

CNNs applications, apart from the SR problems, are **image/video classification**, **image segmentation**, **facial recognition** or **anomalies detection**.

[5] Autoencoders, defined as "an unsupervised learning technique in which we leverage neural networks for the task of representation learning. Specifically, we'll design a neural network architecture such that we impose a bottleneck in the network which forces a **compressed knowledge representation** of the original input", by Jeremy Jordan on https://www.jeremyjordan.me/

General Adversarial Networks/Nets (GANs), being one of the newest DL models as it was introduced by Ian Goodfellow and collaborators just 10 years ago. Gained attention due to its powerful results and ability to generate high-detailed images. [30] GANs feature unsupervised training, implemented by two networks as seen, the generator and discriminator:

- **Generator**: Creates the data trying to mimic the desired output, learning from a map from a latent space to particular data distribution.

- **Discriminator**: Evaluates the samples the generator network made, and compares to real data.

Other applications they have, are in **data augmentation**, advantageous for those training where data is scarce to obtain. As weaknesses, GANs could face high-level **training instability** since they are notoriously hard to train, specially in the *generator-discriminator* equilibrium. Furthermore, limited *generator* output ("mode collapse"), where generator variety is reduced.
Other publicized GANs featuring biomedical data are **FA-GAN** [31] and **BiGAN**. [32]

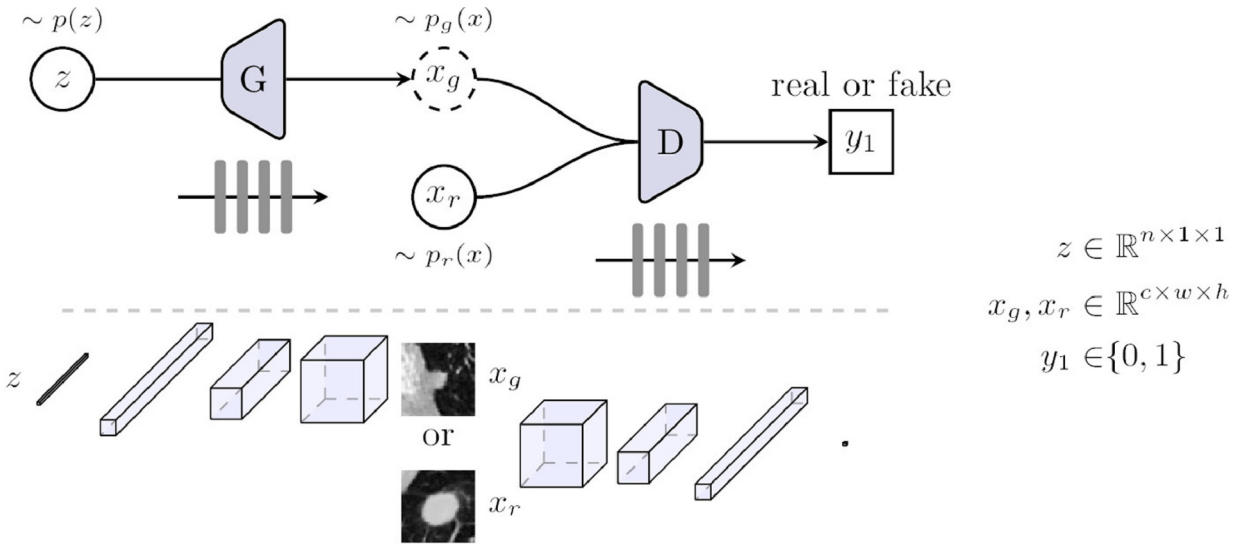


Figure 1.3.3: Schematic example of a GAN and medical usage. *Source: "Generative adversarial network in medical imaging: A review" (2019). Xin Yi* [32]

1.3.3 Deep Learning training and parameters

Following the previous section, we need to deepen on how a training can be done and how critical user-selection can be in order to obtain good results.
Training process steps:

- **Data Preparation:**
  - *Collection:* Gather a large, diverse, and representative dataset.
  - *Pre-processing:* Normalize, scale, and possibly augment the data to make it suitable for training.
- **Model Design:**
  - *Architecture Selection:* Choose a neural network architecture (e.g., CNN for image processing, RNN for sequential data).
  - *Layer Configuration:* Define the number and types of layers (e.g., convolutional, pooling, fully connected layers).
- **Model Training:**
  - *Forward Propagation:* Input data is passed through the network to get an output.
  - *Loss Calculation:* A loss function calculates the difference between the model's predictions and the actual values.
  - *Back-propagation:* The gradient of the loss function is calculated backwards through the network to determine how much each parameter should be adjusted.
  - *Optimizer Use:* An optimization algorithm (like SGD, Adam) adjusts the weights to minimize the loss.
- **Iteration:**
  - *Epochs and Batches:* Training usually occurs over multiple epochs and in batches iteratively.

- **Evaluation:**
  - *Validation:* Original data is splitted to a validation subset to tune hyperparameters and prevent overfitting.
  - *Testing:* After training, test the model on unseen data to evaluate its performance.

Inside loss functions we have:

Loss Function	**Mathematical Formula**	**Use Cases**
Mean Squared Error (MSE)	$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Regression tasks
Mean Absolute Error (MAE)	$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	Regression, robust to outliers
Cross-Entropy Loss	Binary classification: $L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$ Multi-class classification: $L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij})$	Classification tasks
Hinge Loss	$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i \cdot \hat{y}_i)$	SVM [6], binary classification
Perceptual Loss	$L_{\text{perc}}(\phi, y, \hat{y}) = \frac{1}{C_j H_j W_j}\sum_{c=1}^{C_j}\sum_{h=1}^{H_j}\sum_{w=1}^{W_j} (\phi_j(y)_{chw} - \phi_j(\hat{y})_{chw})^2$	Image SR

Table 1.3.3: Main Loss Functions, how they are calculated and usage

[6] Support Vector Machine

1.3.4 Deep Learning Problems-Solution and PSNR, SSIM as evaluators

In the field of deep learning, particularly in applications like image processing or super-resolution, various problems are frequently encountered. Solutions to these issues often involve specific techniques or approaches.

Over-fitting, one of the most recurrent complication. Means that the model performs good on training data but encounter difficulties on new data. *Solutions* are regularization techniques mentioned before or early training stopping.

Under-fitting, ANN is too simple to understand data complexity. *Solution*, create a more complex model, more epochs or improve feature engineering.

Vanishing/Exploding Gradients, while training, gradients can vanish or explode, hindering learning. *Solution*, use initialization/batch normalization or architectures as long short-term memory (LSTM).

Computational Constraints, computational resources are not enough for the given model. *Solution*, optimize the actual model, reduce inefficient layers and improve computational hardware (introduce GPUs, augmented RAM memory).

Other solution approaches to possible deep learning obstacles are **data augmentation** ,**balance model complexity**, **dropout** or **Fine-tuning**.

In our SR problem we are trying to solve with deep learning, there are 2 main objective evaluators, key to develop conclusions and review. **Peak Signal-to-Noise Ratio (PSNR)** estimates the reconstruction quality with signal-rate proportion. The comparison is done with the SR reconstruction and a given HR image. PSNR calculations (equation 1.3.2) have decibels (dB) as units. Increased PSNR means a better result. Range [0,+∞).

On the other hand, we have **Structural Similarity Index Measure (SSIM)**, which range is [-1,1] and the closer it is to 1, the better. SSIM implements direct comparison between two given images, and compares three aspects of the images: structure, luminance and contrast. Is computed by equation 1.3.3.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) \tag{1.3.2}$$

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{1.3.3}$$

Where MSE is calculated using table 1.3.3 formula in PSNR. MAX_I is the maximum possible pixel value in the image. (In uint8, 8 bit-grayscale, would be 255).
In SSIM x,y are common-sized images:

- The mean values of x and y are represented as μ_x and μ_y respectively.
- The variances of x and y are denoted as σ_x^2 and σ_y^2 respectively.
- The covariance between x and y is denoted as σ_{xy}.
- Constants c_1 and c_2 are used to stabilize the division with weak denominators. Usually defined as $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, where L is the dynamic range of pixel values, and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

1.4 Objectives and Motivation

1.4.1 Objectives

Regarding the main objective of this Final Year Dissertation is to provide and evaluate methods and techniques which are able to perform super-resolution on those medical test regarding MRI. In addition, further objectives are to demonstrate the capabilities and potential of deep learning usage in a medical field and how can health engineers provide tools and technical improvements to physicists and any health services professional.
For a personal perspective the objectives are:

- Expand programming languages (*Python, MATLAB*) knowledge and usage, as well as, their libraries and toolboxes for deep learning.
- Learn about the existing and potential algorithms and techniques in deep learning.
- Gain an understanding of the architectures, parameters and configuration.
- Create own response to programming/DL problems, face and understand the possible errors while developing a solution.
- Comprehend the usage of different MRI weights on knee images, along with understanding the anatomical structure and how the injuries are shown on patient testing techniques.

1.4.2 Motivation

From a technical view, improving MR images, can result on better diagnosis and treatment while using lower resources, technologies and time. Knee MRI SR is a much emptier field than other body parts MRI SR as brain. With a quick articles research via *scholar.google.com* we can observe brain SR contains almost 57,000 while knee SR less than 10,000 which is approximately an 82.5% difference. As a sport-person I have experienced live ACL tears or meniscus injuries and although this injuries are much less important (and are far of supposing a patient death) that the possible diseases a brain MRI can find (such as Alzheimer or Cancer), they require a quite long treatment and follow-up. Being able to improve treatment planification and tracking with a better MRI is a big motivation to go through this final year dissertation.

2

Technologies Used

2.1 Data

All MRI data used is property of Stanford Machine Learning Group, published as MRNet Dataset for MRNet competition and research-only purpose. The dataset contains 1,370 exams (5.7GB[7]) with different injuries as ACL tears (23.3%) and meniscus tears (37.1%). A total of 1.088 patients were examined on GE scanners (1.5T/3.0T magnetic field) and the following weighted MRI:

- **Coronal T1.**
- **Coronal T2 + fat saturation.**
- **Sagittal PD.**
- **Sagittal T2 + fat saturation.**
- **Axial PD + fat saturation.**

The dataset also provides labels to each examination in order to work on a classification problem. [33]

[7] GB does reference to Gigabyte

2.2 Software

Inside the programming languages, python and MATLAB will be used. **Python**, developed and published by Guido van Rossum in 1991, is known for being an high-level programming language. Contains an open source license (free to use and distribute). To further development, we have used Spyder[8] on its 5.4.3 version. This software is acknowledged as an integrated development environment (IDE) and as main perks contains **interactive console**, **variable explorer** or **syntax highlighting and code completion**. Conda has been used to manage a specific environment with all libraries needed to execute each Python file. Those libraries are:

- **PyTorch, TensorFlow**: Both stand as deep learning open-source libraries, each one containing unique capabilities and features. PyTorch, developed by Facebook's AI Research lab (FAIR) has strong support for CUDA[9], leveraging NVIDIA Graphics processing units (GPUs). 2.2.0 version was used with CUDA 11.8 as compute platform. On the other hand, TensorFlow is developed by Google Brain team and has also the capability to leverage CUDA for NVIDIA's GPUs. 2.15.0 version was used.

- **OpenCV**: Open Source Computer Vision Library. Containing over 2,500 algorithms to perform various tasks, mainly in computer vision and machine learning. Version to be used is cv2 Python interface on OpenCV 4.8.1 version.

- **NumPy**: Or Numerical Python is one fundamental library for python scientific development. Allows numerical and mathematical operations in Python and is used as a base for many other libraries. Has significant integration with DL libraries.

- **SciPy**: Scientific Python with usability in technical and scientific computing. Builds on the previous seen, NumPy.

- **os**: Standard library to provide operation system-depend functionality. Involves file and directory management.

[8] Spyder stands for "*Scientific PYthon Development EnviRonment*"

[9] "Compute Unified Device Architecture" is a group of tools developed by NVIDIA allowing parallel computing and direct GPU access

- **PythonTelegramBot**.

On the other hand, **MATLAB**[10] is a licensed, high-level programming language. It is mainly a numeric computing environment (matrix manipulation, data visualization, etc). Provides its own graphic user interface (GUI), which is very intuitive. Python libraries equivalent in MATLAB are toolboxes which are generally included in paid license. Toolboxes used:

- **Deep Learning Toolbox**: Made by MathWorks Toolbox. Provides algorithms, apps to create/train/visualize/validate models and pre-trained models. Provides data pre-processing and augmentation functionalities. Version 14.1.
- **Image Processing Toolbox**: Which authors are MathWorks too, provides all type of image functionalities (processing, analyzing, visualization). Contains specific functions regarding DICOMs and is widely used in medical field. Version 11.2.

Other programming unrelated software used has been **X2Go client**, used to access remotely the training server.

[10]MATLAB stands to Matrix Laboratory

2.3 Hardware

All phases unrelated to training has been developed on the following computational technology:

Component	**Description**
CPU	AMD Ryzen 3500U, 2.1 GHz Base Clock
RAM	14 GB DDR4 SODIMM
Storage	512 GB SSD [11]
Integrated GPU	AMD Radeon Vega 8, 2 GB RAM
Operating System	Microsoft Windows 11 Pro 10.0.22635

Table 2.3.1: Computational technology specifications

Training computational technologies:

Component	**Specification**
CPU	Intel Core I5-7400 quad core 3GHz base clock
RAM	8 GB DDR4 SODIMM
Storage	100 GB SSD
Graphics	NVIDIA GPU, 6 GB available RAM
Operating System	Ubuntu 18.04.6

Table 2.3.2: Training Server Specifications

[11] Solid State Disk

3

Development Phases

In this chapter we will discuss about the process followed in order to implement our solution to satisfy the mentioned objectives, as well as the specific models proposed and the code solution implementation.

3.1 Data selection

Firstly, we need to find and select a whole dataset, licensed to be used in research programs. In numbers, as an initial investigation project, we are looking for approximately 1,000 to 5,000 MRI samples. Other important requirement is that the **variety** satisfy our requirements (if for example we took 1,000 T2 sagittal MRIs, our model will not be able to learn other use cases and will only correctly response an specific superresolution request). As well, we need to ensure the original images have the **highest quality possible** so that the information available is at its maximum. **Low noise**, noise can directly affect on the quality and ability to recognise certain structures, so although it can be rectified through algorithms, it is preferable to use a dataset without noise.

On this context, we can find a variety of MRI dataset as ”fastMRI” which was discarded due to its bad distribution of weighted MRIs, ”kneeMRI” dataset from University of Rijeka, but this only contained 1.5T MRIs and finally the chosen ”MRnet” from Stanford ML group. This dataset has all of the above specification and is further explained in subsection 2.1.

3.2 Artificial Neural Network Architecture Design

With the knowledge and observing the data we have available, we proceed to design the ANN that will learn the features from our data. First there was made the decision to use 2D-Convolution approach, which focuses on enhancing the resolution of individual MRI slices, treating each as a separate image although MRI is processed as voxels (3D). This is because Conv3D (equation 3.2.2) has a higher computational cost than Conv2D (equation 3.2.1) and in our specific research in the knee, an anatomical damage can be observed on a single MRI cut (as we saw in subsection 1.2.2), and first/last slices of the MRI may not have specific anatomical data to improve our training and solution. Slices containing specific lateral or medial information will be manually added to the datasets. This would be different in other anatomical evaluations as brain where specific diagnosis as Brain Cancer or Alzheimer requires of a volumetric evaluations and size calculation.

$$(f * g)(i, j) = \sum_{m} \sum_{n} f(m, n) \cdot g(i - m, j - n) \tag{3.2.1}$$

$$(f * g)(i, j, k) = \sum_{m} \sum_{n} \sum_{o} f(m, n, o) \cdot g(i - m, j - n, k - o) \tag{3.2.2}$$

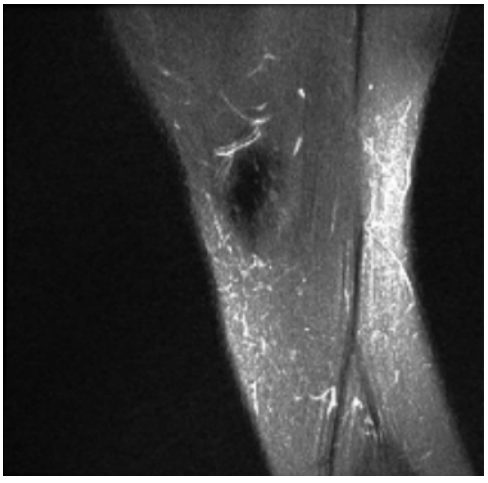


Figure 3.2.1: Example of MRI cut in absence of features information. *Source: MRNet.* [33]

3.2.1 Super-resolution Convolutional Neural Network (SRCNN) Model

First suggested model (which can be graphically seen in figure 3.2.2) will be SRCNN [34], firstly implemented by Chao Dong in 2014 [35] and widely used over medical superresolution problems. Consist on a full convolution layers on three sequential blocks for a total of 57,281 trainable parameters. Those sequential blocks are:

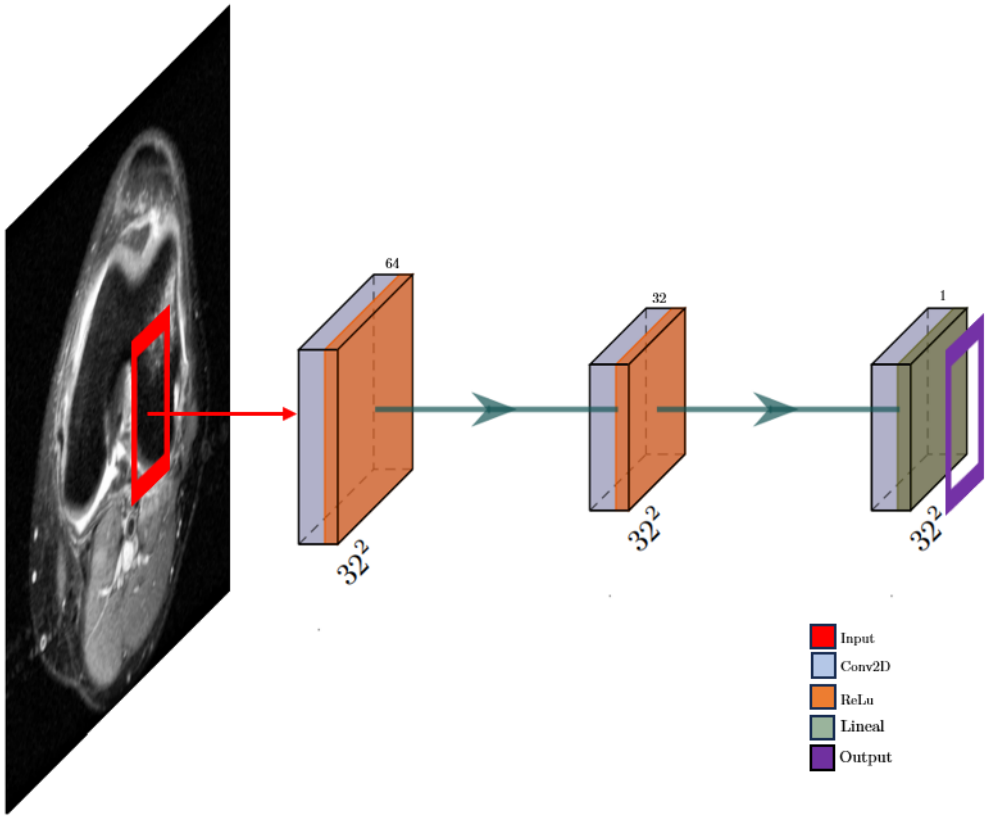


Figure 3.2.2: SRCNN model schema.*Source: Own created following,* [36]

I **Feature extraction layer**, which is the first convolutional layer, consisting on a 1 channel entry (grayscale characteristics of medical image) and 64 outputs channels (filters), 9x9 kernel, 1x1 stride (one pixel movement at a time across the image) and 4x4 as padding to ensure output size. Uses *ReLu* as activation function.

II **Non-linear Mapping Layer** With 64 input channels from the previous layer and 32 output channels as filters, with the objective to consolidate extracted features. 5x5 Kernel size and same stride/padding. *ReLu* activation function.

III **Reconstruction Layer** receives 32 inputs to create a final 1 output channel. Kernel size is 5x5 to combine detailed acquired features into a final image. No activation function/*linear* activation function.

3.2.2 Extended SRCNN model (ExSRCNN)

The second proposed model (figure 3.2.3) to super-resolute medical images. This model consist on an deep feature enhancer that tries to emphasise on deep feature extraction through multiple layers and its potential application in enhancing or transforming images based on learned features. Has a total of 557,953 trainable parameters, and as the previous model is divided in the following three blocks:

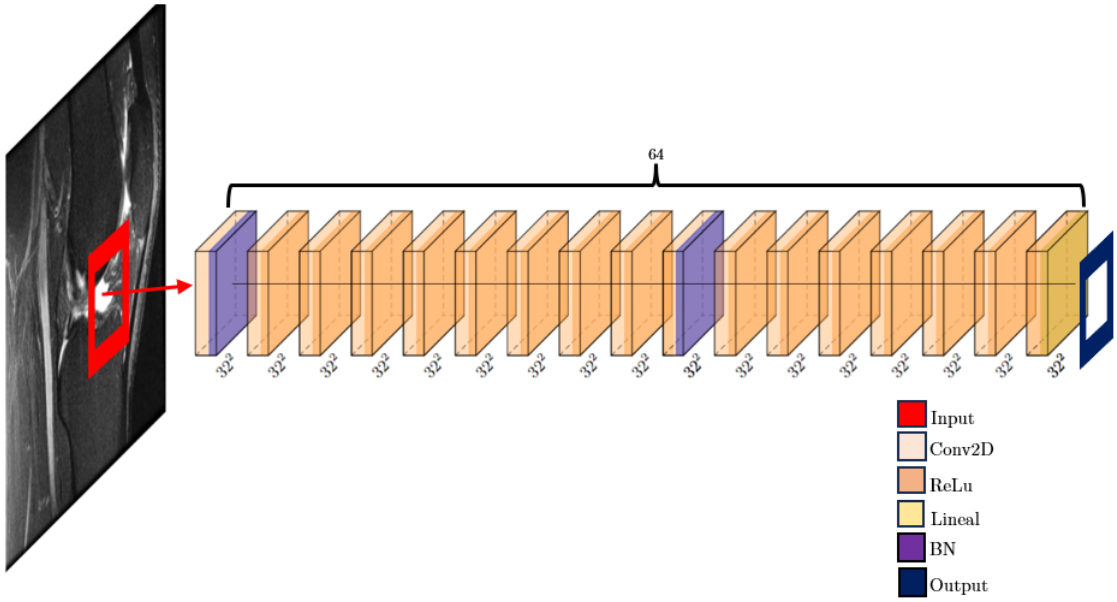


Figure 3.2.3: ExSRCNN model schema.*Source: Own created following* [36]

I **Initial Conv2D Layer**, consisting of 64, 7x7 filters followed by batch normalization, which is added with the aim of improving training stability and boost efficiency. Finally there is *ReLu* as activation function to introduce non-linearity.

II **Intermediate Convolutional Layers -Feature Extraction-**, this is the core of the network's feature extraction capability. Consist of a successive stacked 64-channels convolutional layers, with a 3x3 kernel size filters. Main purpose of these consecutive

layers is to progressively refine image features while trying to maintain computer efficiency. This blocks also adds batch normalization in the 9^{th} layer to help in further stabilization, numerical stability and reduce overfitting risk. Each layer has *ReLu* activation.

III **Output final layer**, with 64 channels and 3x3 kernel, to synthesize the previously processed features into a 1 channel image. As it is the final output, it uses *linear* activation function.

3.2.3 Residual Block SRCNN model (RBSRCNN)

This last model uses a varied structure, integrating the usage of *residual blocks* to facilitate learning and improve performance (figure 3.2.4 . There is a total of 353,729 trainable parameters, divided now in 4 differentiable blocks:

I **Initial Conv2D Layer**, quite equal to the figure 3.2.1 feature extraction layer. Usage of 64 filters with a 9x9 kernel to try capture as much as possible features (simple to complex). Activation Function is *ReLu*.

II **Residual Blocks**, consist of 4 iterations of the same residual block composed by *Conv2D* (3,3) + *Batch normalization* + *ReLu* + *Conv2D(3,3)*+ *Batch normalization*. The whole block will work on 64 input/output channels. This block is intended to refine details, allow gradient flow during back-propagation and help the network to learn identity mappings. At the very least possible situation, signal will go through unaltered.[37]

III **Non-linear mapping Layer** to map the enhanced features from 64 to 32 channels, in aim to prepare the data for a final reconstruction. Has a *Conv2D (5,5)* and *ReLu* activation function.

IV **Reconstruction Layer** is the final step, focused on reconstructing the image using the learned features. Employs *Conv2D (5,5)* with *Linear*/non- activation function for a single channel output.

All the models receive a NCHW tensor, where N stands for the batch size, C the number of image channels which in this case will be 1 for grayscale images, H, W to height and width respectively, referring to the image size. For all this models we will be using 32x32 sized patches, represented in figures 3.2.2, 3.2.3 and 3.2.4 as 32^2.

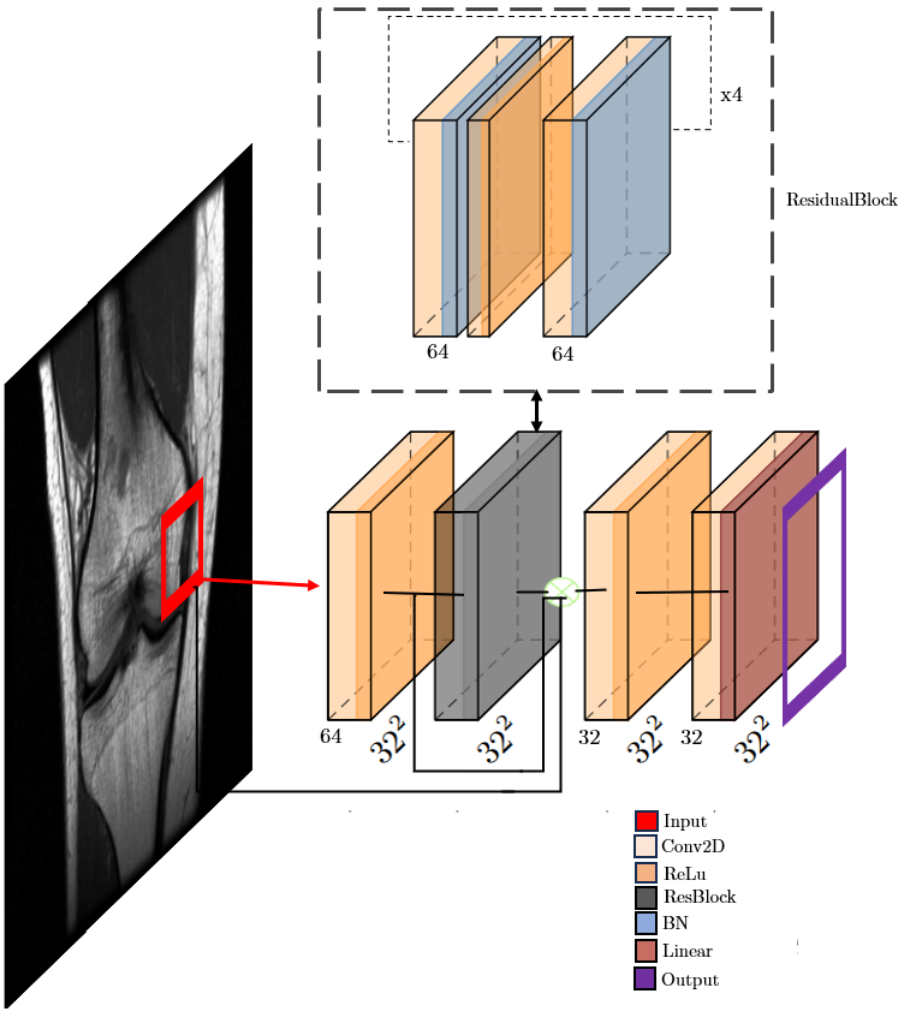


Figure 3.2.4: RBSRCNN model schema.*Source: Own created following,* [36]

3.3 Data Pre-processing

Data processing refers to any specific modification that is done or required in order to employ it in any further task it is required. In our case, MRNet consist on a processed and well distributed dataset, which facilitates our pre-processing task. This dataset is already splitted into sagittal, coronal and axial MRIs which is great for our data selection. With the aim of having a complete dataset, a random slice of each data was taken, excluding few border options for the reasons we saw in subsection 3.2.1 and manually adding useful border slices. Still, all the data was run into a normalization process where *MATLAB* was used following equation 3.3.1.

$$I_N = \frac{(I - \text{Min})}{(\text{Max} - \text{Min})} \cdot (\text{newMax} - \text{newMin}) + \text{newMin} \tag{3.3.1}$$

Obtaining the following absence of differences result:

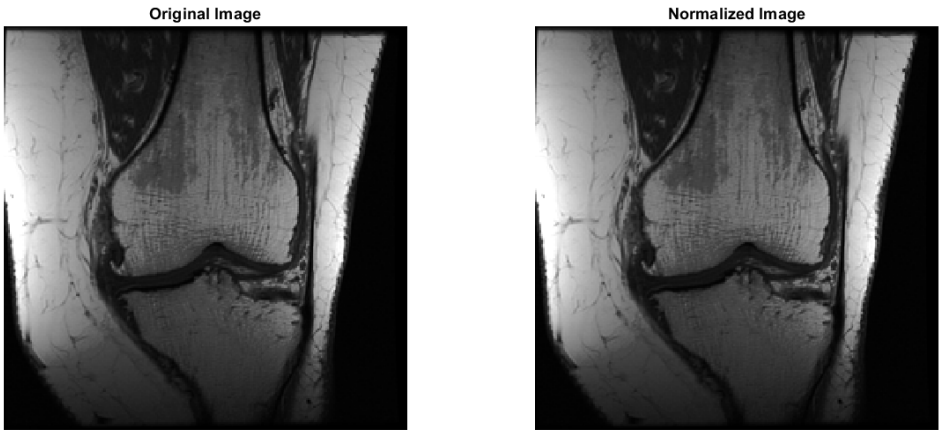


Figure 3.3.1: Normalization process done into a coronal PD MRI. [33]

Other pre-processing needed in order to perform training into our model is obtaining what has been mention as "LR". This is quite easy to do when having "HR" images. Downsampling can be done with both *MATLAB* or *Python* using libraries like *PIL* after selecting an upscale factor or downgrade factor depending on the mathematical changes that are being done. In our project it will not be used any specific method, although there can be applied

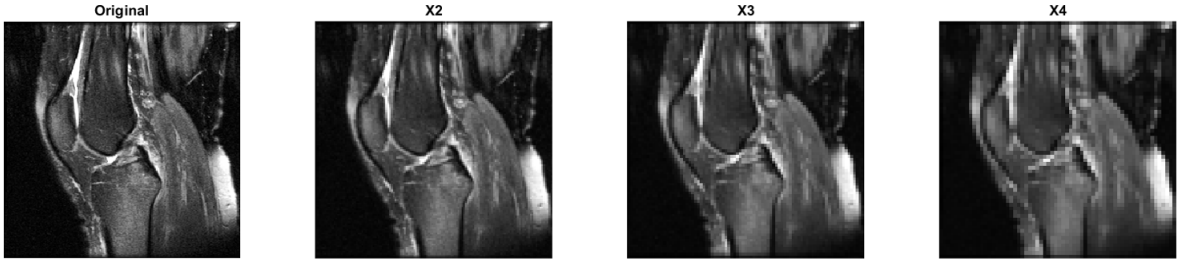


Figure 3.3.2: How resolution downgrades depending on the *Upscale Factor* [33]

some as Nearest-neighbor, Bilinear or Bicubic interpolation. *Upscale Factor/Downgrade Factor* will be establish through all this dissertation at 2/0.5.

The data distribution in **Training** and **validation**, out of the 100% of images used, 85% will be used to training process and 15% to validation. Meanwhile, **Testing** data will be taken from the "valid" independent folder. As mention before, the anatomical plane is fully represented in MRNet, so an approximate equal number of images from each anatomical plane are going to be used. Our models take 32x32 patches, with a 50% overlapping (16x16 overlapping pixels each patch) meaning a 256x256 would generate 255 patches. Final data organization will be:

- **Training**: 223 images for a total of 56,330 *patches* that will be used in LR for training.
- **Validation**: 40 images for a total of 10,240 *patches* pairs $LR - HR$, to evaluate the status of the learning.
- **Testing**: 42 images for a total of 10,752 *patches* pairs $LR - HR$, to test and asses on the best trained result got.

Once the data is ready on each folder, it will be used as *NumPy* arrays before passing it as *Pytorch* tensors through the model.

3.4 Model Training

Once the data is ready and the model design ready to use, is time to go ahead with training. As we saw in subsection 1.3.3 there are still a few parameters and selections that are needed to be made in order to proceed. Regarding *learning rate* it will be establish to $1 * 10^4$ initially, and adjust if needed if the model converges too quick or if training is too slow. Batch size are set to 16 and switched in powers of 2 if needed. This initial batch size allows to reduce computational risk (memory issues). For the weights initialization, all models contain a Kaiming Initialization/He initialization due to the high numbers of *ReLu* activation functions. Basically, it focus in initialize the weights of the Convolutional layers with values drawn from a Gaussian distribution with a mean of 0 and a standard deviation of $\sqrt{\frac{n}{2}}$, where n is the number of incoming nodes of the neurons. **Loss functions** (table 1.3.3) we focus on using are *Mean Squared Error* and *Mean Absolute Error*, to evaluate mathematically how the model is performing. Optimizer options are *Stochastic Gradient Descent (SGD)*, featuring great simplicity and straightforwardness along with memory efficiency, it updates the model weights by calculating the gradient of the loss function in respect of the parameters on a mini-batch. On the other hand, *Adaptive Moment Estimation (Adam)* which tries to combine ideas from the *SGD* optimizer and *RMSprop*. It regularly converges faster than *SGD* and has an adaptive learning rate reducing manual tuning.

3.4.1 Code implementation

The main code base of this project is done about PyTorch libraries due to their capacity and simplicity to develop and the high compatibility, as mention in subsection 2.2 with CUDA GPUs. It follows figure 3.4.1 schematic.

- **Model Implementation**: Model uses the torch library and **torch.nn**. Main elements are done with *nn.Convd2D()* which is the convolutional 2D layer with the arguments as (in, out, kernel size, stride, padding). ReLu activation function *nn.ReLu()*. Batch normalization 2D *nn.BatchNorm2d(channels)* which applies BN on a 4D input. [38]. Linear activation function can be used with *nn.Linear()* or in absence (which was the chosen method). *nn.Module* and *nn.Sequential* are containers used. Weights are initiated with *init normal()* and *zeros()*.

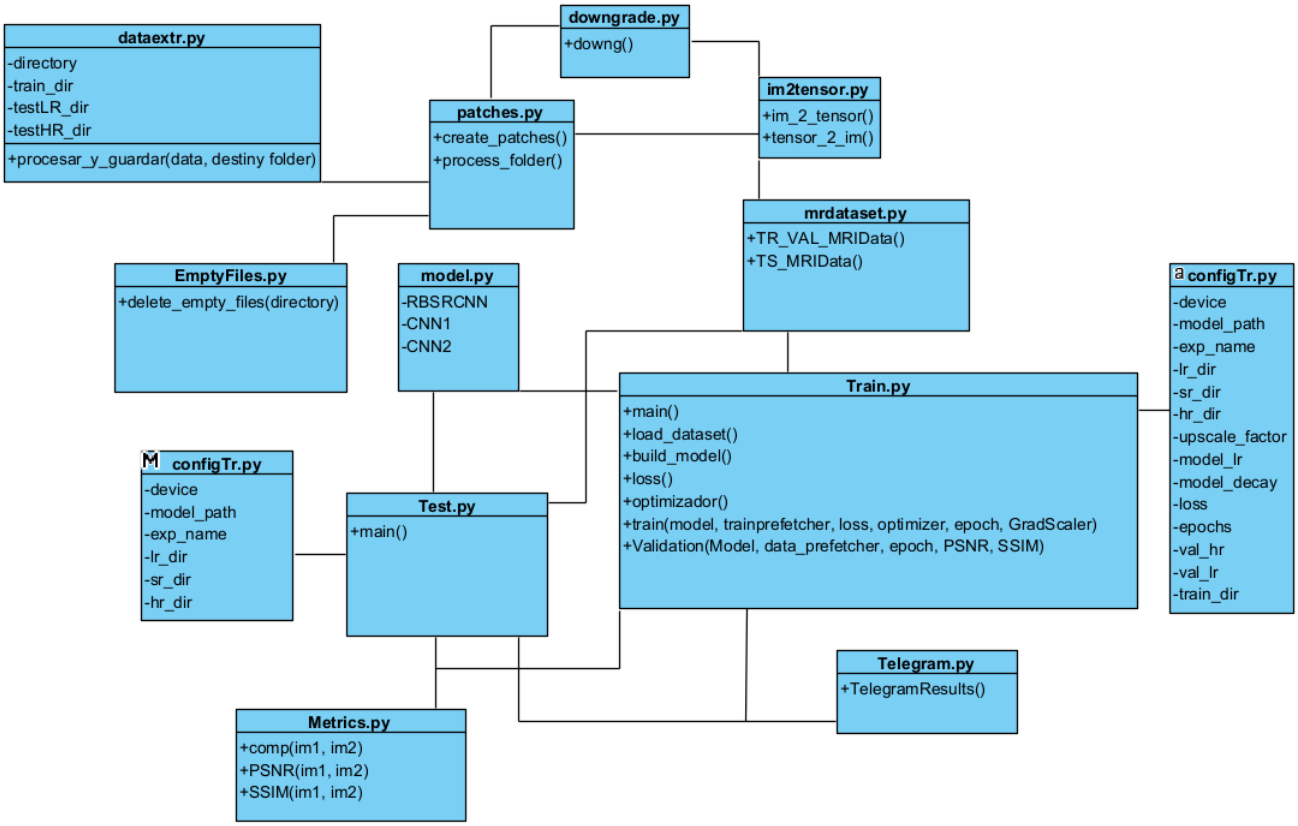


Figure 3.4.1: UML vision of the implementation

- **Images**: Images uses **torchvision functionals** functions as $to\text{-}tensor()$.

- **Dataset managing** with torch.utils.data *Dataset* and *Dataloader*.

- **GPU**, to improve the usage and efficiency torch includes functions as *torch.device* to select the GPU as the main data reader, *cuda.stream()* and so on.

- **Training, Testing**: Model will be saved with torch *save()* function, that generates a pth.rar. file with a Pickle file inside with the configuration and metrics results. Torch *to()* and channels.last as a way of ordering NCHW tensors with the aim of improving optimization. *nn.MSELoss* and *nn.L1Loss* to use the selected loss functions. *Adam/SGD* as optimizer. For testing *.eval()* will turn off Batch Normalization and all layers that are not needed for evaluation while *.train()* allows the model to be set back in a training state. Lastly *load state dict()* loads the model with the specified weights. [39]

3.4.2 Added options

Added to the model training it was developed two additional functions, first a *GPULimit* which limits GPU usage to 50% (or any specific %) of its available memory for the purpose of following the ICAI-TFG server guideline. The second function is a *TelegramBot* which will report validation metrics each 10 epochs directly to the Telegram app, as well as a final testing results. This allows to have a constant report while tracking (as the models are trained remotely in the server) the status and in absence of any message, notify indirectly a possible problem.

3.4.3 Proposed experiments

All experiments will share *Adam* as optimizer, *batch size = 16*, *learning rate* = $1 * 10^{-4}$ and *weight decay* = $1 * 10^{-4}$. These experiments are:

1. **SRCNN Model**. Configured for 300 epochs:
    A. with Mean Squared Error (MSE) as loss function.
    B. with Mean Absolute Error (MAE) as loss function.
2. **ExSRCNN Model**. Configured for 200 epochs:
    A. with Mean Squared Error (MSE) as loss function.
    B. with Mean Absolute Error (MAE) as loss function.
3. **RBSRCNN Model**. Configured for 250 epochs:
    A. with Mean Squared Error (MSE) as loss function.
    B. with Mean Absolute Error (MAE) as loss function.

3.5 Evaluation

Lastly, to evaluate the model there is the *Testing part*, where it will be used the metrics seen in subsection 1.3.4 PSNR and SSIM to get an objective evaluation by comparing the superresolution done by our model SR, with the reference high resolution HR image. In order to compare with other proposed models, our evaluator will be the metrics difference with experiments 1.A) and 1.B) (reference experiment).

As a subjective evaluations, medical field related people will be asked to evaluated different parameters and -if possible- give a final diagnose of a given SR image.

4 Results

In this chapter the results will be numerically, graphically and visually evaluated. Those results are given by the experiments ran. It will be divided in 3 sections. Firstly, a internal comparison between the internal results, checking which of the models -and experiments- proposed are more suitable for a superresolution task. Once a final conclusions about our models are done, a comparison to other proposed models related to knee MRI superresolution will take place, specially those who use the same metrics as we did. Lastly, the most important evaluation will be done by medical field professionals or students.

4.1 Internal Evaluation

Following the obtained data the validation and testing hands, there will be:

- Plotted results from validation PSNR and SSIM metrics taken with the *TelegramBot*.
- Table regarding **Testing** results $\pm$ testing standard deviation σ which is given by equation 4.1.1.
- High-resolution, low-resolution and the different superresolution images compared to obtain a visual contrast and judgement.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{4.1.1}$$

4.1.1 Validation Graphs

This first results are related to validation, which takes the data destined to this specific task and runs it through the epoch's model weights that has been calculated giving a first look on

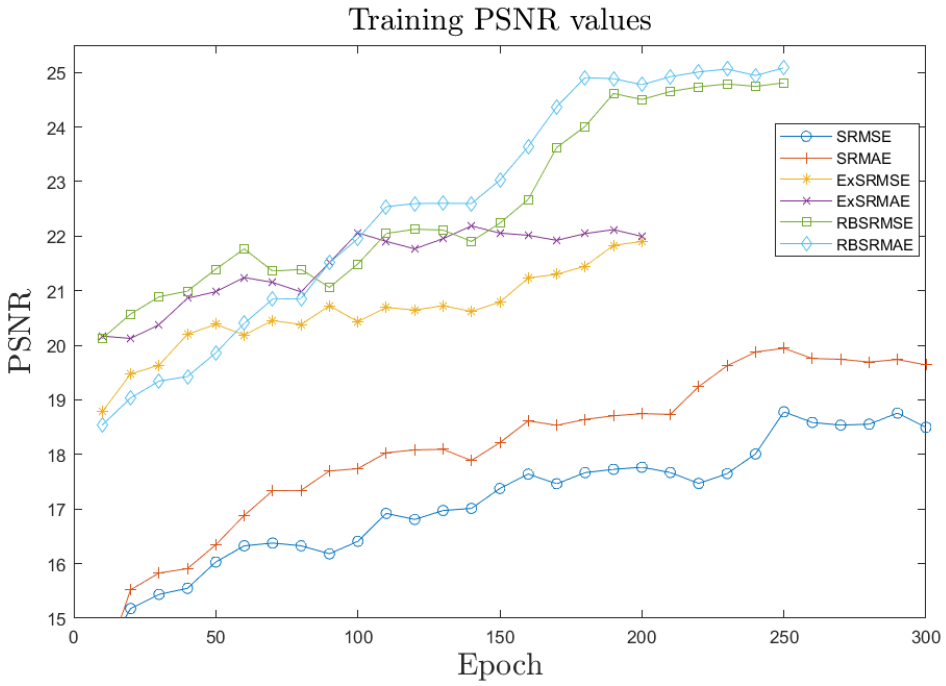


Figure 4.1.1: Training/Validation *PSNR* Values along 200/250/300 epochs

how the models parameters training is doing over the validation data. On figure 4.1.1, it can be seen that by the end of the graph, there does not seem to be a total convergence of PSNR values for the different models, indicating that they might still be learning and could possibly improve with an extended training time but figure 4.1.2 shows there might be a tendency to plateau by the end of the graph meaning that an extended training time conclusion, as a possible improvement, cannot be ensured. In both metrics it can be observed that the leading performers are experiments 3.B) and 3.A), both using Residual Blocks. The experiments using mean average error are clear to be the best-performers, based on both *PSNR* and *SSIM*. [40] Regarding other models as the Extended SRCNN (experiments 2.A), B)), they exhibit quick initial improvements followed by some flattening in later epochs. Finally SRCNN models exhibit the most moderate metrics, although their known potential in superresolution tasks, they still have the simplest architecture models.

Figure 4.1.2: Training/Validation *SSIM* Values along 200/250/300 epochs

4.1.2 Testing result

In this subsection, testing results will be presented on tables. Testing is done using the model's weights where the best validation metrics were taken. In addition, it uses completely unseen data, with the aim to obtain the most trustworthy results.

Model	**Experiment**	**PSNR**	**SSIM**
SRCNN	MSE	18.599 ± 0.337	0.634 ± 0.012
	MAE	19.714 ± 0.258	0.672 ± 0.018
ExSRCNN	MSE	21.598 ± 0.262	0.716 ± 0.008
	MAE	21.983 ± 0.2762	0.716 ± 0.009
RBSRCNN	MSE	24.10 ± 0.3078	0.733 ± 0.014
	MAE	$\mathbf{24.786} \pm 0.3111$	$\mathbf{0.745} \pm 0.012$

Table 4.1.1: Comparison of model performance in testing

Table 4.1.1 certifies first hypothesis that were done on the previous subsection 4.1.1 in which experiment 3.B) was estimated as the best model and loss function to use for superresolution

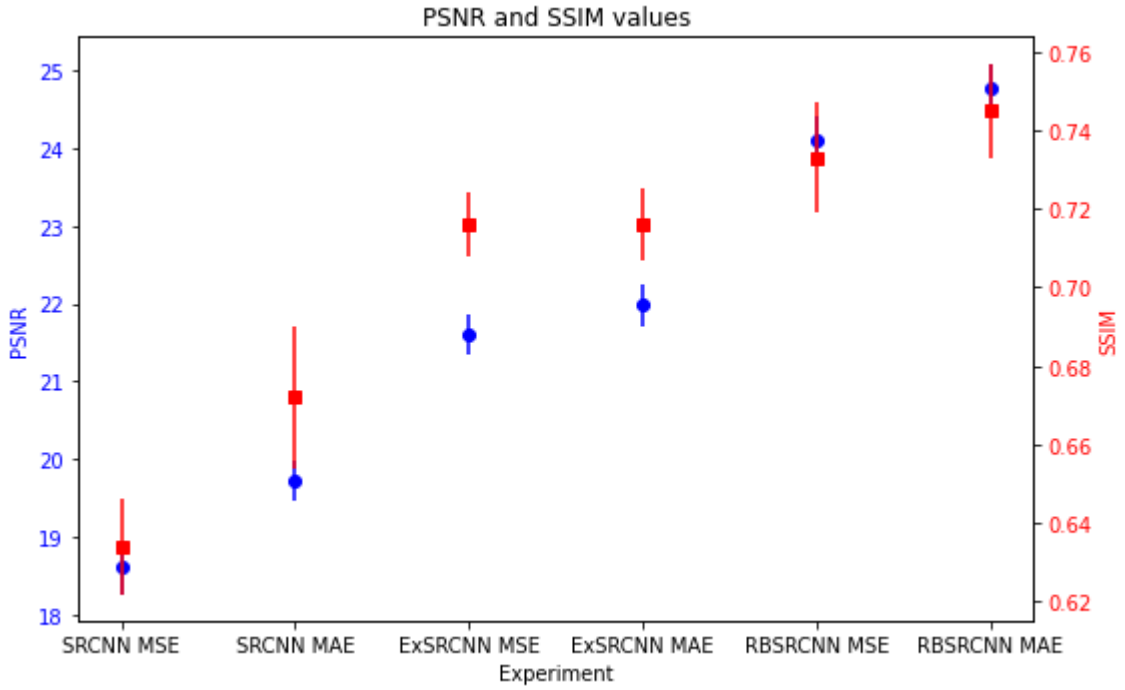


Figure 4.1.3: Graphical view of PSNR and SSIM for each model with their standard deviation

tasks. This metrics will be the one used for future comparisons between other models as it was mention. Table 4.1.1 results can be seen on graphs in figure 4.1.3

As well, experiments regarding SRCNN with both loss functions MSE and MAE, will be used as a reference to evaluate the potential improvement to the new proposed models, ExSRCNN and RBSRCNN.

4.1.3 Visual overview result

To asses and evaluate on the MRIs superresolution results, they have been placed in figure 4.1.4 with a 50x50 crop zoom in a region of interest, in order to check the true performance. All SR images are obtained through the models trained with MAE as loss function. (B-labeled experiments).

First graphical evaluation that can be done is that from HR to LR there is a quantitative loss in anatomical details, checking on the zoom crop we can observe the absence of detail and bigger sharpness. Any medical condition can be noticed although specific diagnose can be laborious.

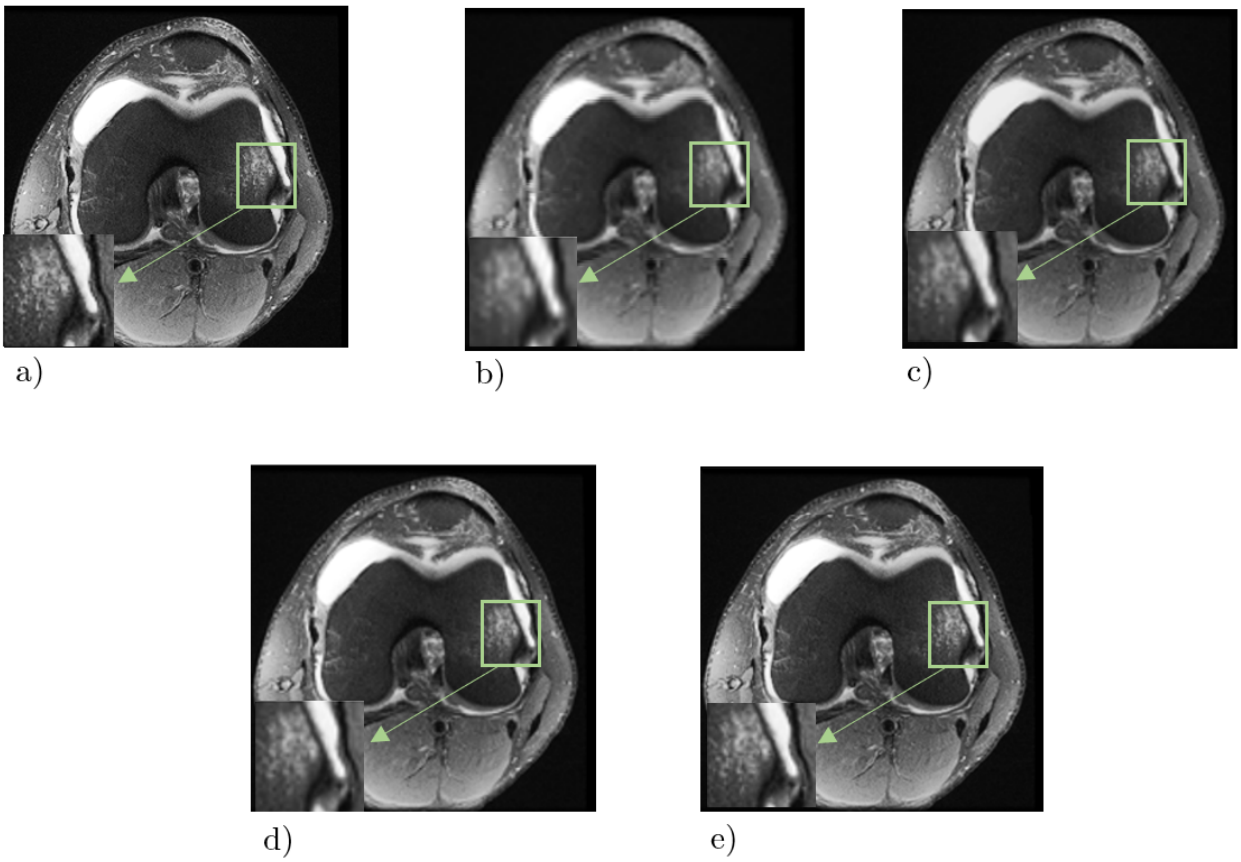


Figure 4.1.4: T2 Weighted Axial MRI. a) Original HR image. b) X2 factor LR image. c) SRCNN. d) ExSRCNN. e) RBSRCNN

Following we have **SRCNN** and **ExSRCNN** which uses those trained parameters to improve noticeable quality in the MRI. The specific anatomical details remains slightly blur to the human eye but are closer to the SR and some invisible damage are now shown. Different borders are also shown more clearly.

Lastly, as it could been expected, the best theoretically model, has the best performance in order to super-resolute a LR image. In this last MRI are now clearer and much closer to how a HR looks, giving quite a detailed crop showing possible damage. On the other hand, in the anterior view of the knee (upper part of the image) it can be seen how the superresolution could not faithfully reproduce some details as the Epidermis (commonly known as skin). In addition, some gastrocnemius (lower part of the image) muscle fibers can be observed.

4.2 External Evaluation

Following, it will be discussed the models performances and potential with other proposed models from the scientific community. As previously discuss, our comparison model will be the best-performer Residual Blocks SRCNN with Mean Absolute Error as specific configuration.

4.2.1 Other proposed CNN models

Others models proposed for a superresolution task in knee MRI are:

- **EMISR**, proposed by [41]. Consist of a *Conv(3x3, 64 output channels) with TanH* [12] + "Smart Network" of 4 different masks in 32 output channels with *PReLu*[13] + *Conv(3x3), with TanH* as hidden layers before adding a sub-pixel convolution layer. Figure 4.2.1.
- **DeepResolved**, was published back in 2018 [42]. Consist on a cascade of *Conv3D(3x3x3 filters and a map length of 64) + ReLu* + final residual block which adds to the input. Figure 4.2.2 .
- **CN-CNN** composed of 5 ResNet blocks and data consistency layer. Was proposed by [43].
- **DCCN** which is an extended model from the previous, with 5 blocks consisting of CNN and data consistency layer. Was also suggested by [43]

[12] Hyperbolic tangent activation function

[13] Parametric ReLu is an extended ReLu that adds as a learnable parameter

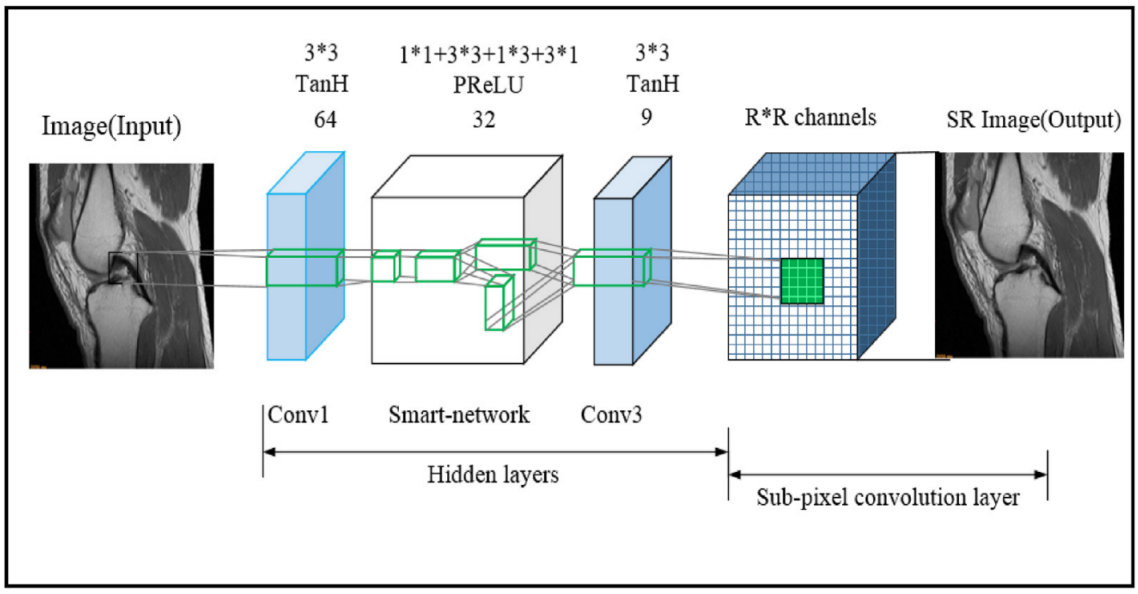


Figure 4.2.1: EMISR structure [41]

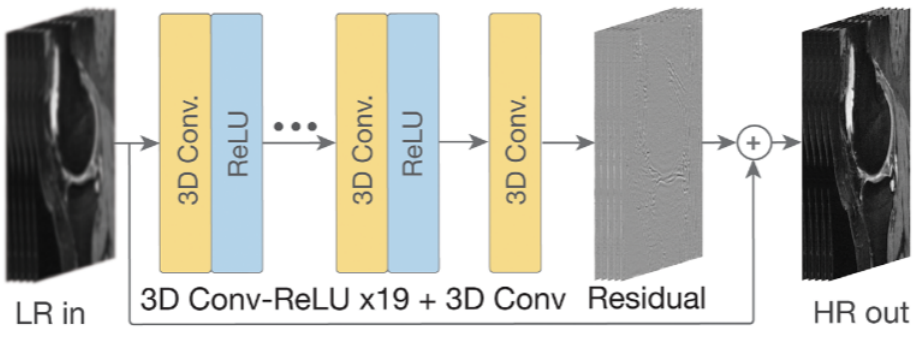


Figure 4.2.2: DeepResolved structure [42]

4.2.2 Metrics and results

The given and published metrics for common x2 downgraded images as LR are:

Model	PSNR	SSIM	Loss function
RBSRCNN	24.786	0.745	MAE
EMISR	**36.22**	-	Euclidean Loss
DeepResolved	20.8	**0.92**	MSE
CN-CNN	28.57	0.822	MSE
DCNN	28.78	0.829	MSE

Table 4.2.1: Metrics comparison of different superresolution models and experiments on Knee MRI

Which places the suggested model in an average scale. Still, we cannot presuppose very final conclusions following this data due to the results dependency in other factors as training data, optimization functions, hyperparameters or training procedures and capabilities.
Some specific mention trained data are *Osteoarthitis Initiative (OI)* data used to train *DeepResolved* which contains sagittal/coronal MRIs done with 3T acquisition equipment and focused on knee articular damage. *EMISR* was trained on non-specific medical data and applied to I Do Imaging (IDI) open source sagittal MRI dataset. For reference, just Defu Qiu [41] implemented the SRCNN model to compare experiments results. In his case, he improved by 2.308 dB while RBSRCNN model improved PSNR metrics in 5.07 dB and SSIM in 0.073 which may be an indicator of potential development to higher metrics.

4.3 Medical Field Subjective Evaluation

In this last section of the chapter, a few people will be asked to asses on a superresolute MRI, generated with the experiment 3.B) parameters. Main purpose of this task is to confirm the superresolution algorithm just betters the image quality without affecting the anatomical details or addition of artifacts. This people are:

1. Radiologist specialist with +20 years of experience. Currently working for "Sistema Andaluz de Salud (SAS) in Ronda and Antequera as a specialist.
2. Non radiologist neither trauma specialist. General internal medicine specialist. Working for "Servei Català de Salut".
3. Medical school post-graduate.
4. Last year medical school student in Universidad de Málaga.
5. Last year medical school student in Universidad CEU San Pablo.
6. 4^{th} year medical school student in Universidad de Santiago de Compostela.

And they will be asked to:

- Give a numerical valuation from 0 to 5 to both LR and SR images shown in figure 4.3.1. This evaluation has to be based on sharpness, presence of artefacts, contrast and overall satisfaction.
- Enumerate soft tissues observable on the SR image (figure 4.3.1). For this task, reviewer 1. will tell which are this tissues in the original HR image.
- Give a final diagnose on the SR image. For this task evaluators will be given the complete MRI study after downgrading each slice and applying superresolution with experiment 3.B). The study given is Train/331 from MRNet, which is labeled as an ACL tear.
- Once the previous task are fulfilled, they will be asked to give a similar numerical evaluation for the original HR. This task is done after to prevent and secure an uninfluenced answers on the other tasks.

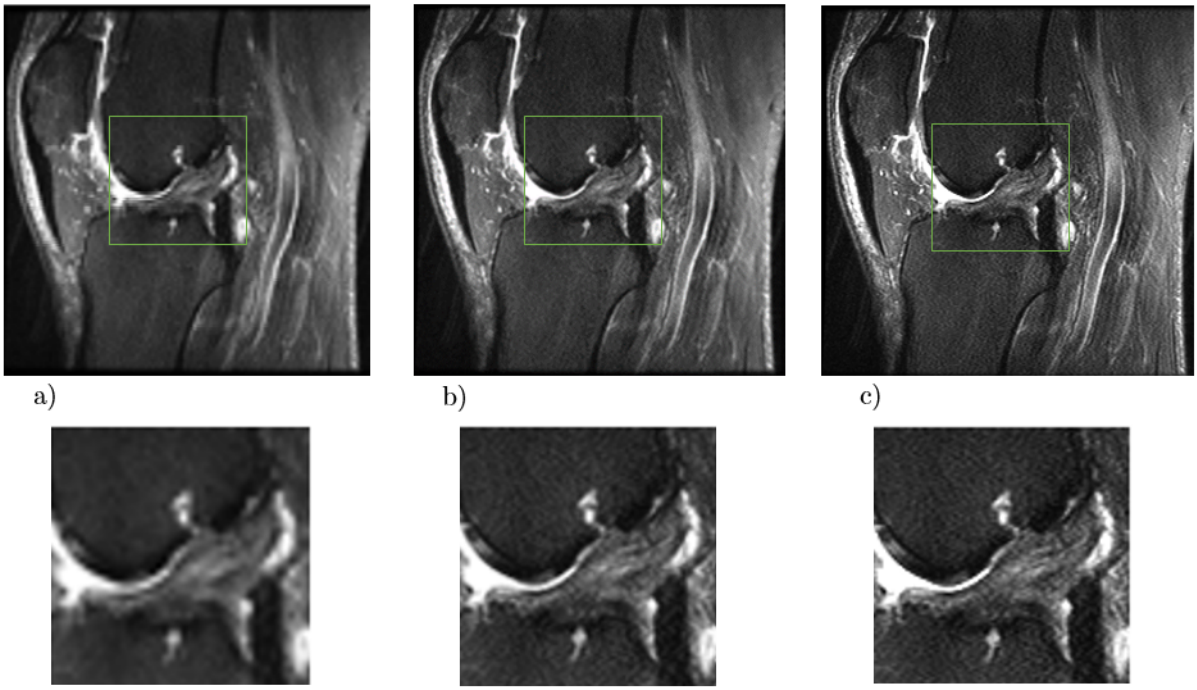


Figure 4.3.1: Sagittal T2 MRI. a) LR full and crop images. b) SR full and crop images. c) HR full and crop images. [33]

There are a total of 8 main soft tissues that might be seen on figure 4.3.1. Those are:

- Popliteal artery.
- Quadriceps/patella tendon.
- Semimembranosus and gastrocnemius muscles.
- Articular cartilage and lateral meniscus.
- ACL and PCL

Evaluator/Task	LRValue	SRValue	HRValue	AnatEnum	Diagnose
1	3.53	4.50	4.85	8	ACL Tear
2	3.58	4.62	4.87	8	ACL Tear
3	3.80	4.70	5.00	8	ACL Tear
4	3.75	4.70	4.90	6	ACL Sprain
5	3.50	4.65	5.00	6	ACL Tear
6	3.80	4.75	4.90	4	ACL [14]
Average	3.665±0.051	4.65±0.039	4.92±0.021	-	-

Table 4.3.1: Evaluation for the different tasks proposed.

Following the given answer shown in table 4.3.1, we could conclude our solution is closer to a HR MRI than to a LR MRI. Both anatomical structures and injury remained unaltered after performing the superresolution. Some evaluators still noticed, after observing the HR image, lower quality textures in muscular anatomy. While this does not affect this specific diagnose, could be an issue if looking for deep muscles injuries. However, the muscles were still noticeable.

[14]Evaluator could see how the ACL became brighter but was not able to be specific with any injury.

5

Conclusions and Futures Lines of Research

5.1 Conclusions

When this work was thought, the main objective was to be able to apply and use deep learning models to improve the imaging tests performed on patients and, in this specific case, on the knee. Therefore, if we evaluate the main objective, this is fulfilled along with the ability to have demonstrated usefulness and correct functionality in our own models, with different architectures to those published by other researchers.

In the context in which this development takes place, we could conclude positively with the results obtained, understanding the situation and the possible differences they may have had with other experiments that have been compared throughout this work. In this area, performance variability is due to more factors than just a specific model architecture or parameters. Also, the evaluations and responses obtained from medical professionals and trainees are a very positive indication for our results. It is essential that when working with elements which are basic to the health of patients, they are not modified from reality by adding or removing details.

The absence of data on the different training times for each experiment can be observed. This is due to the variability of the training environment and its saturation. Although the execution has been carried out constantly with 2GB of GPU memory as the main device, the training were also using RAM memory and CPU to try and speed up the processes. This previously mentioned, unlike the GPU memory, were not reserved for a specific use, so depending on the general load of the server, the possibility of use was higher or lower. Therefore, exact bench-

marks could not be obtained, and although it can be roughly concluded that experiments 2 and 3 requires 2 to 3 times more training time, this data cannot be guaranteed or proven.

On a personal level, this project has helped me in order to deepen my development in the programming languages used and their application combined with elements seen in some subjects of the degree. Related to a professional career, knowledge in both *MATLAB* and in *Python* is key for many specialities. Also the fact of dealing and designing deep learning elements makes my specific knowledge in them greater and, above all, clearer.
With regard to the medical part, which is an inherent fraction of this degree, the understanding and deepening of the medical imaging field for diagnosis, as well as the specific learning of the anatomy of the musculoskeletal system, provides a greater volume of knowledge. In general, MRIs and other tests can be very complex to visualise and identify, so this project has been really useful in developing an understanding of the elements of the knee and possible abnormalities, and how they can be visualised on MRIs.
Finally, being able to deal with this project and its execution in a language that is non-native to me is a challenge that I am grateful to have been able to face. Using English is probably more tedious and time-consuming, but it is essential to learn and be aware of the technical lexicon commonly used by the scientific and medical community.

5.2 Future lines of Research

5.2.1 Graphical User Interface

The first idea proposed after completing this work is to create a GUI—and generally all the necessary software—so that the pre-trained models can be used by healthcare personnel or anyone who desires to, in a simpler way. Realistically, execution through Python files and console, as well as the creation of conda environments with their corresponding libraries might be too complicated. In general, a future line would be to implement a GUI with the following objectives:

- **Improve data interpretation**, by presenting and effectively visualizing it.
- **Accessibility**: as previously mentioned, the implementation of a graphical interface significantly improves usability possibilities, especially in an environment where extensive computer knowledge cannot be assumed.
- **Direct comparative analysis**: The results could be presented more directly and alongside the image which SR is being applied.
- **Integration into hospital systems**: It could be more simply integrated with the software already present in hospitals and which is known by the professionals who use it daily.
- **Use as an educational tool**: Students related to medicine would also benefit from being able to access this type of software, helping them in their studies and training.
- **Ease of configuration**: Just like in accessibility, configuring the options that are desired is much simpler with interfaces that can guide the user.

5.2.2 Classification

As mentioned in subsection 2.1, the data used for the research is owned by the Stanford Machine Learning group. These data contain labels according to whether they present anomalies or not, having been individually evaluated by medical professionals. This presents an

ideal situation to implement deep learning classification models on images obtained by super-resolution. Another possible option would be to combine the development of this project with some of the best results of the MRNet competition, such as "Triple-MRNet (single model)" by Yash Bhalgat [44] with a result of 0.904 AUC[15].

5.2.3 Evaluating other types of models

The variety of models in deep learning but along with neural networks in super-resolution, we saw that another possibility were generative adversarial networks, better known as GANs. These are very recent but have already demonstrated different uses in the field of medicine, as well as their results (Koshino, Kazuhiro. 2021) [45] or (Kazeminia, Salome. 2020)[46]. Therefore, there would be the possibility of developing GANs to be able to evaluate and compare results with the work done.

5.2.4 Weight-transformed images

In subsection 1.1.4, we discussed the different weighting options that an MRI image may have, as well as the observable anatomical differences. Throughout this report, various illustrations with their respective weightings have been presented to favor the visualization of specific lesions. However, it is not difficult to find cases where a knee injury induces another, damaging different tissues. For this purpose, there is a current trend in development using recent technologies like artificial intelligence to create MRI images faithfully simulating other weightings based on a test with a different weighting. Recent research [47], [48], are evidence of this on-growing field. Therefore, a possible future direction would be to expand super-resolution research, along with HR images, to this practice, obtaining a wider variety of images and elements for effective diagnosis in each test.

[15](Area Under the Curve), is a widely used measure to evaluate the performance of classification models whose range is [0,1]

6

Conclusiones y Líneas Futuras

6.1 Conclusiones

Cuando se idea este trabajo, el principal objetivo era poder aplicar y utilizar modelos de aprendizaje profundo para poder mejorar las pruebas de imagen que se realizan sobre pacientes y en este caso específico, sobre la rodilla. Por tanto, si evaluamos el objetivo principal, este se encuentra cumplido junto con la capacidad de haberse demostrado utilidad y una funcionalidad correcta en modelos propios, con arquitecturas diferentes a las publicadas por otros investigadores.

En el contexto en el que se encuentra este desarrollo, podríamos concluir positivamente con los resultados obtenidos, entendiendo la situación y las posibles diferencias que hayan tenido con otros experimentos que se han comparado a lo largo de este trabajo. En este ámbito, la variabilidad de rendimiento se debe por más factores aparte de una arquitectura específica de un modelo o parámetros. Asimismo, las evaluaciones y respuestas obtenidas por parte de profesionales médicos y personas formándose para ello, es un indicativo muy positivo para nuestros resultados. Resulta esencial que cuando se trabaja con elementos, los cuales son básicos para la salud de los pacientes, estos no sufran modificaciones de la realidad, añadiendo o eliminando detalles.

Así mismo, se puede observar la ausencia de datos sobre los diferentes tiempo de entrenamientos para cada experimento. Esto se debe a la variabilidad del entorno de entrenamiento y su saturación. Aunque la ejecución se ha realizado de manera constante con 2GB de memoria GPU como dispositivo principal, los entrenamientos estaban utilizando también memoria RAM y CPU para intentar acelerar los procesos. Estas últimas mencionadas, a diferencia de la

memoria GPU, no se encontraban reservadas para un uso específico por lo que dependiendo de la carga general del servidor, la posibilidad de uso era mayor o menor. Por ende, no se han podido obtener referencias exactas, y aunque se puede concluir de manera aproximada que los experimentos 2 y 3 requieren de 2 o 3 veces más de tiempo de entrenamiento, no se pueden asegurar ni demostrar estos datos.

A título personal este proyecto me ha ayudado en gran parte a profundizar en el desarrollo en los lenguajes de programación utilizados y su aplicación combinada con elementos vistos en algunas asignaturas del grado. De cara al mundo profesional, tanto los conocimientos en *MATLAB*, como en *Python* resultan clave para muchas especialidades. También el hecho de tratar y diseñar elementos de aprendizaje profundo hace que mi conocimiento específico en ellos sea mayor y sobre todo, más claro.

Respecto a la parte médica, la cual forma parte inherente de este grado, la comprensión y profundización en el campo de las imágenes médicas para el diagnóstico, así como en el aprendizaje específico de la anatomía del sistema locomotor otorga un volumen mayor de conocimiento. En general, las resonancias magnéticas y otras pruebas pueden resultar muy complejas a la hora de visualizar e identificar, por lo que este proyecto ha sido realmente útil para desarrollar una capacidad de entendimiento sobre los elementos de la rodilla y las posibles anomalías, y como se pueden visualizar en las IRM.

Por último, el haber podido tratar con este proyecto y su ejecución en una lengua que no es nativa para mi, supone un reto del cual me siento agradecido de haberme podido enfrentar. El utilizar el inglés probablemente sea más tedioso y requiera de mayor inversión de tiempo, pero resulta fundamental para aprender y entender el léxico técnico de uso común por parte de la comunidad científica y médica.

6.2 Líneas Futuras

6.2.1 Graphical User Interface

La primera idea que se plantea tras realizar este trabajo, es crear una GUI -y en general todo el software necesario- para que los modelos preentrenados puedan se utilizados por parte del personal sanitario o todo aquel que lo desee usar de una manera más sencilla. Realistamente, la ejecución a través de ficheros de Python y consola, así como creación de entornos conda con sus correspondientes librerias puede llegar a ser complicada. En general una línea futura sería implementar una GUI con los siguientes objetivos:

- **Mejorar la interpretación de datos**, mediante su exposición y visualización de manera efectiva.
- **Accesibilidad**: Como se ha mencionado previamente, la implantación de una interfaz gráfica mejora de manera notable las posibilidades de uso, especialmente en un entorno donde no se puede presuponer conocimientos extensos de informática.
- **Análisis comparativo directo**: Se podrían presentar los resultados de forma más directa y junto con la imagen donde se está aplicando.
- **Integración en sistemas hospitalarios**: Se podría integrar de forma más simple con los softwares ya presente en los hospitales y los cuales son conocidos por los profesionales que lo usan a diario.
- **Uso como herramienta educativa**: Aquellos estudiantes relacionados con la medicina también se beneficiarian de poder acceder a este tipo de software, ayudándoles en sus estudios y formación.
- **Facilidad de configuración**: Al igual que en la accesibilidad, la configuración de las opciones que se deseen resulta mucho más sencillo con interfaces que pueden guiar al usuario.

6.2.2 Clasificación

Como se mencionaba en el segundo capítulo, la propiedad de los datos utilizados para la investigación son de Stanford Machine Learning group. Estos mismos datos contienen etiquetas

según si presentan anomalías o no, habiendo sido evaluadas individualmente por profesionales médicos. Esto plantea una situación ideal para implementar modelos de aprendizaje profundo de clasificación en imágenes obtenidas por superresolución. Otra posible opción sería juntar el desarollo de este proyecto junto con alguno de los mejores resultados de la competición MRNet, como puede ser " Triple-MRNet (single model)" de Yash Bhalgat [44] con un resultado de 0,904 de AUC[16].

6.2.3 Evaluar otro tipo de modelos

La variedad de modelos en el aprendizaje profundo pero junto a las redes neuronales en superresolución, vimos que otra posibilidad eran las redes generativas antagónicas, más conocidas como GANs. Estás son muy recientes pero ya se han demostrado diferentes usos en el ámbito de la medicina, así como sus resultados (Koshino, Kazuhiro, 2021) [45] o (Kazeminia, Salome)[46]. Por tanto, cabría la posibilidad de desarollar GANs para poder evaular y comparar resultados con el trabajo realizado.

6.2.4 Transformación de imágenes

En la subsección 1.1.4 se vió las diferentes opciones de ponderación que puede presentar una imagen de resonancia magnética, así como las diferencias anatómicas que son obvservables. Durante esta memoria se han ido mostrando diferentes ilustraciones con sus respectivas ponderaciones que favorecian la visualización de lesiones específicas. Sin embargo, no es díficil encontrar casos donde una lesión de rodilla induzca a otra, dañando diferentes tejidos. Para ello, existe una corriente de desarollo utilizando tecnologías recientes como la inteligencia artificial para crear IRM de manera fideligna simulando otras ponderaciones a raíz de una prueba con ponderación diferente. [47] o [48] son investigaciones recientes en ámbito en crecimiento. Por tanto, una posible linea de futuro sería poder expandir la investigación de superresolución, junto con imágenes HR a esta práctica, obteniendo en cada prueba una variedad mayor de imágenes y elementos para un diagnístico eficaz.

[16](Area Under the Curve), es una medida ampliamente utilizada para evaluar el rendimiento de los modelos de clasificación cuyo rango es [0,1]

References

[1] Brian M Dale, Mark A Brown, and Richard C Semelka. *MRI: basic principles and applications*. John Wiley & Sons, 2015.

[2] Johan Lilja. *Multi subject study of motor sensory function in rat brain using fMRI*. Stockholm, Sweden, 2005.

[3] Caroline Andrews, Andrew Simmons, and Steve Williams. Magnetic resonance imaging and spectroscopy. *Physics Education*, 31(2):80, 1996.

[4] Mateusz Wilczek. Larmor frequency: Radiology reference article, Mar 2009. Accessed 2023-07-21.

[5] Ehsan Samei and Donald J Peck. *Hendee's physics of medical imaging*. John Wiley & Sons, 2019.

[6] Jaume Gili. Introducción biofísica a la resonancia magnética. *Centre Diagnóstic Pedralbes*, 5, 1993.

[7] Paul Davidovits. *Physics in biology and medicine*. Academic press, 2018.

[8] Suraj D Serai. Basics of magnetic resonance imaging and quantitative parameters t1, t2, t2*, t1rho and diffusion-weighted imaging. *Pediatric radiology*, 52(2):217–227, 2022.

[9] Wilfrid Taylor Dempster and George RL Gaughran. Properties of body segments based on size and weight. *American journal of anatomy*, 120(1):33–54, 1967.

[10] 3DMedical. Complete anatomy (10.1.1), 2024 edition.

[11] Steven Paul Arnozcky. Anatomy of the anterior cruciate ligament. *Clinical Orthopaedics and Related Research*, 172(NA;), 1983.

[12] Caroline A Murphy, Atul K Garg, Joana Silva-Correia, Rui L Reis, Joaquim M Oliveira, and Maurice N Collins. The meniscus in normal and osteoarthritic tissues: facing the structure property challenges and current treatment trends. *Annual review of biomedical engineering*, 21:495–521, 2019.

[13] José Becerra, José A Andrades, Enrique Guerado, Plácido Zamora-Navas, José M López-Puertas, and A Hari Reddi. Articular cartilage: structure and regeneration. *Tissue Engineering Part B: Reviews*, 16(6):617–627, 2010.

[14] Michel D Crema, Frank W Roemer, Monica D Marra, Deborah Burstein, Garry E Gold, Felix Eckstein, Thomas Baum, Timothy J Mosher, John A Carrino, and Ali Guermazi. Articular cartilage in the knee: current mr imaging techniques and applications in clinical practice and research. *Radiographics*, 31(1):37–61, 2011.

[15] SB Thacker, DF Stroup, CM Branche, J Gilchrist, RA Goodman, and E Porter Kelling. Prevention of knee injuries in sports. *Journal of Sports Medicine and Physical Fitness*, 43(165-179), 2003.

[16] Susan Standring. *Gray's anatomy e-book: the anatomical basis of clinical practice*. Elsevier Health Sciences, 2021.

[17] Vandana Jahanvi and Abhimanyu Kelkar. Chemical shift imaging: An indispensable tool in diagnosing musculoskeletal pathologies. *SA Journal of Radiology*, 25(1), 2021.

[18] Vandana Jahanvi and Abhimanyu Kelkar. Chemical shift imaging: An indispensable tool in diagnosing musculoskeletal pathologies. *SA Journal of Radiology*, 25(1), 2021.

[19] K Markatos, MK Kaseta, SN Lallos, DS Korres, and N Efstathopoulos. The anatomy of the acl and its importance in acl reconstruction. *European Journal of Orthopaedic Surgery & Traumatology*, 23:747–752, 2013.

[20] Simon M Gianotti, Stephen W Marshall, Patria A Hume, and Lorna Bunt. Incidence of anterior cruciate ligament injury and other knee ligament injuries: a national population-based study. *Journal of science and medicine in sport*, 12(6):622–627, 2009.

[21] Dr. Ajay.C.Desay. Cases courtesy, 2010.

[22] LE Ramseier, CML Werner, and M Heinzelmann. Quadriceps and patellar tendon rupture. *Injury*, 37(6):516–519, 2006.

[23] Leon D Rybak and Martin Torriani. Magnetic resonance imaging of sports-related muscle injuries. *Topics in Magnetic Resonance Imaging*, 14(2):209–219, 2003.

[24] Jenny T Bencardino, Zehava S Rosenberg, Robert R Brown, Alvand Hassankhani, Elizabeth S Lustrin, and Javier Beltran. Traumatic musculotendinous injuries of the knee: diagnosis with mr imaging. *Radiographics*, 20(suppl_1):S103–S120, 2000.

[25] Kamal Nasrollahi and Thomas B Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25:1423–1468, 2014.

[26] Zhenzhu Meng, Yating Hu, and Christophe Ancey. Using a data driven approach to predict waves generated by gravity driven mass flows. *Water*, 12(2):600, 2020.

[27] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic programming and evolvable machines*, 19(1-2):305–307, 2018.

[28] John Murphy. An overview of convolutional neural network architectures for deep learning. *Microway Inc*, pages 1–22, 2016.

[29] Teppei Fukuya and Yasushi Shibuta. Machine learning approach to automated analysis of atomic configuration of molecular dynamics simulation. *Computational Materials Science*, 184:109880, 2020.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[31] Mingfeng Jiang, Minghao Zhi, Liying Wei, Xiaocheng Yang, Jucheng Zhang, Yongming Li, Pin Wang, Jiahao Huang, and Guang Yang. Fa-gan: Fused attentive generative adversarial networks for mri image super-resolution. *Computerized Medical Imaging and Graphics*, 92:101969, 2021.

[32] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552, 2019.

[33] Standford ML Group. Mrnet, a knee mri dataset and competition, 2022. [Accessed: (12.10.2023)].

[34] Lornatang. Srcnn pytorch, 2022.

[35] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[36] Haris Iqbal. Harisiqbal88/plotneuralnet v1. 0.0. *URL: https://doi. org/10.5281/Zenodo*, 2018.

[37] Yajun Qiu, Ruxin Wang, Dapeng Tao, and Jun Cheng. Embedded block residual network: A recursive restoration model for single-image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4180–4189, 2019.

[38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[40] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through fsim, ssim, mse and psnr—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

[41] Defu Qiu, Shengxiang Zhang, Ying Liu, Jianqing Zhu, and Lixin Zheng. Super-resolution reconstruction of knee magnetic resonance imaging based on deep learning. *Computer methods and programs in biomedicine*, 187:105059, 2020.

[42] Akshay S Chaudhari, Zhongnan Fang, Feliks Kogan, Jeff Wood, Kathryn J Stevens, Eric K Gibbons, Jin Hyung Lee, Garry E Gold, and Brian A Hargreaves. Super-resolution musculoskeletal mri using deep learning. *Magnetic resonance in medicine*, 80(5):2139–2154, 2018.

[43] Wei Zeng, Jie Peng, Shanshan Wang, and Qiegen Liu. A comparative study of cnn-based super-resolution methods in mri reconstruction and its beyond. *Signal Processing: Image Communication*, 81:115701, 2020.

[44] Yash Bhalgat. MRNet-Competition, 12, 2019.

[45] Kazuhiro Koshino, Rudolf A Werner, Martin G Pomper, Ralph A Bundschuh, Fujio Toriumi, Takahiro Higuchi, and Steven P Rowe. Narrative review of generative adversarial networks in medical and molecular imaging. *Annals of Translational Medicine*, 9(9), 2021.

[46] Salome Kazeminia, Christoph Baur, Arjan Kuijper, Bram van Ginneken, Nassir Navab, Shadi Albarqouni, and Anirban Mukhopadhyay. Gans for medical image analysis. *Artificial Intelligence in Medicine*, 109:101938, 2020.

[47] James A Grant-Jacob, Chris Everitt, Robert W Eason, Leonard J King, and Ben Mills. Exploring sequence transformation in magnetic resonance imaging via deep learning using data from a single asymptomatic patient. *Journal of Physics Communications*, 5(9):095015, 2021.

[48] Elisa Moya-Sáez, Óscar Peña-Nogales, Rodrigo de Luis-García, and Carlos Alberola-López. A deep learning approach for synthetic mri based on two routine sequences and training with synthetic data. *Computer Methods and Programs in Biomedicine*, 210:106371, 2021.

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA