Universidad Carlos III de Madrid

Biomedical Engineering

Bachelor Thesis

“Application of Super Resolution Convolutional Neural Networks (SRCNNs) to enhance medical images resolution”

Author: Andrea Lorenzo Polo

Tutor: Javier Pascau González-Garzón

External Tutor: Jose Antonio Calvo Haro

Leganés, July 2019

Author: **Andrea Lorenzo Polo**

Tutor: **Javier Pascau González-Garzón**

External Tutor: **Jose Antonio Calvo Haro**

Title: **Application of Super Resolution Convolutional Neural Networks (SRCNNs) to medical images.**

THE TRIBUNAL

President:

Secretary:

Vocal:

The dense of this Bachelor Thesis took place the 5 of July 2019 in Leganés, at the EPS (Escuela Politécnica Superior, School of Engineering) of the Universidad Carlos III de Madrid (UC3M), earning a FINAL MARK of

President: Secretary: Vocal:

Acknowledgements

I would like to express my appreciation to every person that has contributed to make this project possible.

To all the members of the Biomedical Instrumentation and Imaging Group (BIIG). To Javier Pascau, for allowing me to develop this project, and Manuel Concepción, for enormous help provided along the project.

To all my friends, especially to my roommates and Miguel and Seán, who have accompanied me through all this process, suffering and celebrating every step with me.

Finally, this wouldn’t have been possible without the infinite support of my family. Thank you for believing in me and motivating me to give my best. Thanks to my mother, whose strength and love inspires me every day.

Abstract

The importance of resolution is crucial when working with medical images. The possibility to visualize details lead to a more accurate diagnosis and makes segmentation easier. However, obtention of high-resolution medical images requires of long acquisition times. In clinical environments, lack of time leads to the acquisition of low-resolution images.

Super Resolution (SR) consist in post-processing images in order to enhance its resolution. During the last years, a branch of SR is getting promising results. This branch focuses in the application of Convolutional Neural Networks (CNNs) to the images.

This project is intended to create a network able to enhance resolution of knee MR stored in DICOM format. Different networks are proposed, and evaluation is made by computing Peak Signal-to-Noise Ratio (PSNR) and normalized Cross-Correlation. One of the networks proposed, SR-DCNN, presented better results than the conventional method, bicubic interpolation. Finally, visual comparison of the SR-DCNN and bicubic interpolation also showed that the network proposed outperforms the conventional methods.

KEYWORDS: Super Resolution, Deep Learning, Convolutional Neural Network, Medical Images.

INDEX

List of Figures

List of Tables

1 INTRODUCTION

1.1 Approach to the Problem ant State-of-the-art

Since the apparition of medical imaging, the seek for improve the resolution has been continuous. The acquisition of high-resolution medical images is determinant when it comes to providing accurate diagnosis and, consequently, a proper treatment [1]. The detection of small details can provide earlier diagnosis, which can be decisive for certain diseases.

Until recently, research has focused on increasing resolution of the images by improving the hardware. State-of-the-art acquisition machines are equipped with powerful detectors, able to obtain high resolution images. This relies on the application of advanced physics and the use of expensive, high precision detectors. Therefore, the obtention of high-resolution images is limited by cost and time [2].

During the last years, a more affordable and available alternative has become common; to post-processing the image once it has been taken. Super Resolution (SR) methods are understood as the processes able to accept a low-resolution (LR) image as input and convert it into a high-resolution (HR) image. SR methods are classified in three categories, known as interpolation-based, multi-frame based and learning based. The simplest approach for doing this was the use of interpolation methods such as nearest neighbour, bilinear and bicubic interpolation. However, these conventional methods tend to produce artifacts such as blurring or aliasing and to over-smooth the image [3]. Although multi-frame-based methods overcome these problems, their application remains a challenge in real-time medical image due to the computational time required [1].

The importance of deep learning in SR dates from 2014 with the Super Resolution Convolutional Network (SRCNN) proposed by Dong et al. [4]. This manuscript presented promising results in the application of neural networks (NN) for image enhancement. The first application of deep-learning based SR in medical images dates from 2017. This year, Dr. Umehara published "*Super-resolution convolutional neural network for the improvement of the image quality of magnified images in chest radiographs*", applying the network proposed by Dong to chest radiographs [3] . Since then, the study of deep-learning SR methods has shown a great potential and provided promising results. The implementations of the basic SRCNN first proposed are numerous as well as its application in different areas. Implementations proposed included the addition of more convolutional layers, residual networks and deconvolutional layers [5] [6] [2], and they are applied to modalities such as mammography or MRI [7].

1.2 Motivation and Objectives

This project is motivated by an internship at Hospital General Universitario Gregorio Marañón. The main task of this internship consisted in segmenting different anatomical structures in order to create 3D models that could be 3D printed. Although this worked successfully with bony structures, a problem appeared when different tissues, such as cartilage, were introduced to the design.

Particularly, this project focuses on MR knee images. Stacks of knee images from different patients were available in the hospital's database. For these volumes, it was very difficult to segment anything different from the bone since the quality was very low. In the hospital environment it is common that, due to the long waiting list of MRIs and CT scans, images present low resolution. In order to reduce acquisition time, the number of images creating a volume is reduced. The less the images forming the volume the wider the area they represent, that is, the higher the slice thickness. A high slice thickness means a low resolution.

The solution would be to increase the resolution, but due to the waiting list of the image acquisition systems of the hospital, this cannot be done. Another solution consists in applying SR, a post-processing method that seeks to improve an image's resolution once an image is acquired.

Until now, SR methods just focused on single images. This project seeks to enhance the resolution of MR images of the knee by focusing on the entire volume instead of a single image. For this purpose, it has been used a dataset including a huge collection of knee MRI images provided in DICOM[1] format by the Osteoarthritis Initiative. However, it is expected that the results obtained can be easily translated to different imaging areas and to different anatomical structures. The final goal is the process followed in the clinical environment when acquiring and image resembles the one in Figure 1.

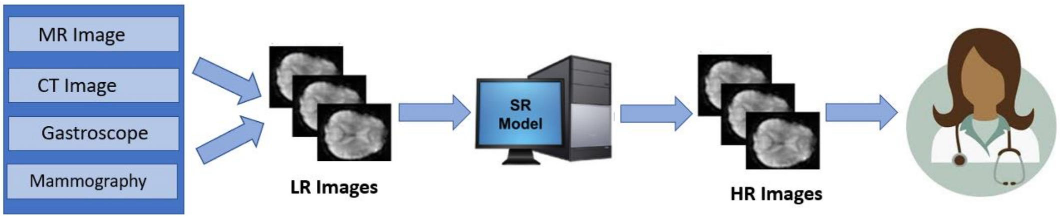


Figure 1. Medical image super resolution reconstruction process overview.

In order to familiarize with Deep Learning (DL) and Super Resolution, this project is divided in two parts. First, a simple SRCNN is design and applied to both, natural and medical images. Once this has been successfully done, the goal is to apply the network directly to DICOM volumes. The objectives are the following:

- To develop Deep-Learning based SR method able to enhance resolution of natural images in order to fully understand the functioning of the SRCNN and the processes needed.
- To identify the key differences between working with natural images and DICOM volumes in order to modify the network designed.
- To evaluate the results and analyse the impact this method could have when implemented in clinical and research areas.

[1] Digital Imaging and Communications in Medicine (DICOM) is the standard for storage and transmission of medical imaging information and data.

1.3 Regulatory Framework

1.3.1 Data Protection

For the obtention of the MRI images, the student has signed a Data Use Agreement for a Limited Data Set. This dataset is provided under the HIPAA Privacy Regulations and exclude identifiers as it can be names or telephone numbers. Besides, the agreement establishes that the data provided can only be used with research and/or educational purposes.

1.3.2 Technical Standards

Since the boom of machine learning (ML), its application for clinical purposes has become a big issue. At the moment, the FDA[2] is working in a regulatory framework to evaluate medical products that use machine learning.

Two are the main reasons this technique is generating so much controversy. One is that these algorithms keep learning from the new data when it is available, therefore they keep changing once they have been approved. The FDA has already approved machine learning software; however, the algorithms require to be "frozen" before its commercial use and need a reapproval for any implementation. The other problem resides in what is known as the "Black-box medicine", an issue almost unavoidable in DL. It consists in the lack of knowledge of what is the algorithm doing. This is due to the complexity and amount of data used, which leads to opaque computational models [8].

Therefore, right now the application of DL to enhance the resolution of medical images cannot yet be standardized as a method. However, the imminent future of DL will surely bring regulatory frameworks allowing for the use of DL [9].

1.4 Socio-Economic Environment

1.4.1 Project Budget

The table below breaks down an estimation of the budget needed to carry out this project. All the software used in this project, described in Materials section, is open-source. The data used is provided by the Osteoarthritis Initiative and do not suppose any economic impact.

Project Budget				
Materials	**Description**	**Unitary Cost**	**Months Used (Amortization)**	**Total**
	Nvidia Titan X	€ 1.310,00	9	€ 327,00
	Asus TPL300L	€ 550,00	9	€ 137,00
	Software	€ -		€ -
	OAI Knee MRI	€ -		€ -
Personnel	**Description**	**Cost (€/hour)**	**Hours Worked**	
	Professor	25	50	€ 1.250,00

[2] The Food and Drug Administration (FDA) is a federal agency of the USA department of Health and Human Services.

	Physician	25	10	€ 250,00
	Intern	17	150	€ 2.550,00
	Student	13	700	€ 9.100,00
Resources	**Description**	**Cost (€/month)**	**Months**	
	Internet	33	8	€ 264,00
Total Estimated Cost				**€ 12.628,00**

Table 1. Project Budget Breakdown.

1.4.2 Socio-Economic Impact

In Spain, the cost of a acquiring an MRI image ranges between 95 and 230€, while the cost of the acquisition device ranges from 150,000 up to 3,000,000 €. Taking as reference the data provided by Community of Madrid, the number of CT obtained per year rounds the 500 thousand and the MRI 290 thousand [10].

The acquisition time required can be obtained from the formula below [11]. Where TR is the pulse sequence, the number of phase encodes is determined by the matrix size and NEX is the number of excitations. The slice thickness, and therefore the image resolution, depends on TR value. Therefore, a larger TR will increase the acquisition time. According to Table 6, the acquisition of a high-resolution MRI image ranges from 7 to 10 minutes.

$$Aquisition\ Time\ = TR * \#phase\ encodes * NEX * \frac{1}{60{,}000}$$

The application of the network designed in this project is intended to be use in clinical environments where MRI machines are already present but there is a patient overcrowding. The long waiting lists cause images to be taken in short times, reducing image quality. The aim of the network is to optimize the acquisition process by enhancing the images with post-processing.

For example, with a scaling factor of 4, instead of obtaining 160 slices with a thickness of 0,7mm, it would be enough with obtaining 40 slices resulting in a slice thickness of 2,8mm. Passing this volume through the SRCNN, the resolution obtained would be similar to the high-resolution acquisition.

By doing this, the acquisition time is reduced around four times. This is translated in a reducing of the waiting list in the hospital and a better use of the hospital facilities.

1.5 Document Structure

After this introduction, section 2 focuses on giving a full explanation of what SR is, its importance in medical imaging and the SR methods. The most promising SR method is based on DL; section 3 provides a background of Deep Learning and explains the fundamentals concepts of a neural network so that tis functioning can be understood. After this, the lecturer is expected to have the background needed to follow up the project methods.

Section 4 includes the materials, hardware and software, used for this project. Then, the project is divided in two parts, as well as the methods, results and conclusion for each one. The first part of the project, in section 5, consists in developing a SRCNN trained with natural images. The next part consists in the optimization of that network so that it can be applied to DICOM volumes MR knee images.

Finally, section 7 covers the discussion, limitations and future work of the project. Appendixes can be found in section 8 and Bibliography in section 9.

2 SUPER RESOLUTION

The concept of Super Resolution refers to the process when one or more images' resolution is enhanced. Before introducing the concept and demonstrate why is it so important, first, the concept of resolution must be understood. Spatial resolution is the most common parameter used to measure resolution of images and it can be described as *the length of the smallest detail that can be observed* or *the closer two lines could be, being still differentiable.* An illustration of how resolution is measured can be found in Appendix A. Spatial Resolution.Pixels are the *building blocks* of an image, that is, the smallest units an image is composed of. Therefore, the minimum resolution achievable is dependent on the pixel size of an image.

Many different areas such as astronomy, video surveillance or medical imaging deal with the concept of resolutions and are immersed in the challenge of enhancing resolution. This section is intended to provide a brief background about medical imaging and explain the importance that resolution has in this area. Then, Super Resolution applied to medical imaging is introduced and all its different methods described.

2.1 Need of SR in Medical Imaging

Complications often appear during the obtention of medical images due to limitations of the acquisition machine, imaging environment and quality-limiting factors. The most common problems affecting the quality of the image are low contrast, low resolution, geometric deformations and presence of artifacts and noise. The quality of the image is directly related to the quality of the acquisition device [12]. However, as the performance of the detectors increases so does the cost of the devices and this could not be affordable. Due to the high-demand these acquisition machines have and the limited number of them in clinical environments, physicians often obtain low resolution images that, on the other hand, require less time to be taken. The resolution of the image is therefore limited by time and cost; however, the process does not finish once the image is acquired.

Digital image processing is the process by which the raw signal obtained from the sensor is converted into a digital image and posteriorly enhanced. It can be divided into three different phases that are Pre-Processing, Enhancement and Information Extraction. Digital imaging processing allows to get over abnormalities in the image and obtain the original information [12].

Improving the resolution of a medical image will results in a more detailed image containing more information, which could be crucial in order to detect a certain disease and provide the proper treatment. For example, a better resolution leads to a better classification of regions, which could help to localize a tumour more accurately. Furthermore, it also enhances the pre-operatory planning and consequently contributes to the success of the operation. Finally, increasing the resolution enhances the performance of automatic detection and image segmentation methods.

The importance of resolution in medical imaging is crucial. However, there is a trade-off between resolution and economical and time factors. Due to the high cost and difficulty of enhancing resolution of an image during its acquisition, a common solution is to accept that there will be image degradation and use image processing methods to post process the obtained image [13]. Over the last decades many methods have been proposed with the purpose of enhance resolution, these have been englobed under the term Super Resolution, and its application in the medical area have become quite important.

2.2 Super Resolution Methods

The resolution of an image is limited by the imaging acquisition device. The higher the number of sensors present, the better the resolution. This can be achieved by reducing the sensor's size, however, as this is done the amount of light incident also decreases, resulting in noise. In addition, smaller sensors increase the cost of the hardware. Super resolution has become an affordable alternative to deal with the resolution problem.

"Super Resolution refers to the task of restoring high resolution images from one or more low resolution observations" [14]. Although it can be classified in Single Image SR and Multiple Image SR, along this work only Single Image SR is going to be studied and it will be referred as SR. In fact, most SR methods designed are intended to increase resolution of single images. There are three main categories, depending on the algorithms used, that classify the existing SR methods.

- **Interpolation-based SR**. This method consists in a set of popular operations as the nearest neighbour (NN), linear, bicubic or bi-spline interpolations. The process by which these techniques operate consists in creating new pixels whose value is obtained from the values of the neighbour pixels. The simplest method, NN, simply takes the value of the nearest pixel. More complex methods are based on mathematical operation that take into account a greater number of neighbour pixels, achieving more accurate.

  Interpolating is equivalent to apply a low pass filter to our image, as it can be seen they just *join* points, trying to make these joints as smooth as possible[3]. However, there is a drawback in applying these methods. When used in high-frequency areas it results in blurred edges (bilinear or cubic) or blocking artifacts (NN). However, interpolation methods are often chosen due to their speed and simplicity.

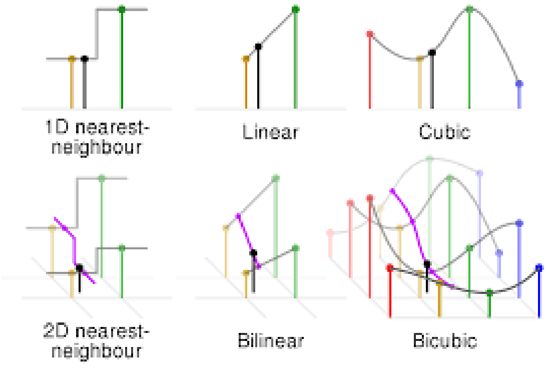


Figure 2. Graphical example of NN, linear and cubic interpolation in 1D and 2D. The colour dots correspond to the neighbour samples and the black dots to the interpolated value [15].

[3] Note that signal from the real world have a *smooth* shape, meanwhile the signals in computers are made by blocks. Interpolation tries to smooth these blocks so that they resemble more to the real word signals.

- **Multi-frame-based methods.** This technique uses multiple low resolution (LR) images to produce a single or a set of high resolution (HR) images. The common procedure consists in obtaining a prior image domain and then stabilize the space by iterative back-projecting errors between the reconstructed HR and LR images [16].

  The performance of these methods is limited since they are very sensitive to their assumed model of data and noise and the fact that, in medical imaging, it is difficult to obtain a set of complementary LR images. Furthermore, this technique requires a significant computational cost making these reconstructions very time consuming [17] [18].

- **Learning-based SR**. This relatively new approach has attracted most of the attention during the last decade. It is based in learning a relationship between HR and LR images, which results in a trained model able to translate a LR input into a HR output.

  Early learning-based methods were based on large dictionaries that related LR patches from the input image to HR ones and, after this, the output image was obtained by reconstruction using the HR patches. These methods have been improved with introduction of (1) Neighbour Embedding, which divided the dictionary into multiple neighbours, (2) Sparse Coding methods, which learned a dictionary and then apply a sparse vector to associate the LR to HR patches and, finally, (3) the introduction of Regression Models.

  For the last years, the state-of-the-art of learning-based SR has been Deep Learning. The first Super Resolution Convolutional Neural Network (SRCNN) was proposed in 2015 by Dong et al. [4] and many implementations have been proposed afterwards. With Deep Learning, Learning-based has outperformed interpolations and multi-frame methods without increasing the computational charge [2].

Learning-based method are able to *create* new details, something that is completely impossible doing interpolation. For example, imagine a photography of an Indian person with a Bindi (red dot) between their eyebrows. This photo has lost resolution and the Bindi appears as a small red blur. If we train a model with photos of persons, some wearing the Bindi, the model will know that a red blur between the eyebrows means that the person is wearing a Bindi. Furthermore, the model will be able to reconstruct it. However, interpolating methods will just give new values to those pixels so that they look like the surrounding pixels representing the skin. This will result in the red blur dying down until it become unnoticeable. The fact that the model is creating new pixels from its own *knowledge* could be both an advantage but also a disadvantage. However, for a model that has been trained properly and results have shown to work properly, the model will outperform interpolation methods.

Summing up, the introduction of Deep Learning (DL) in SR brought several benefits. First, neural networks achieve better simplicity and accuracy than state-the art-method. Second, even in practical on-line methods they operate fast, since are fully feed forward and do not need to be optimized. That is, once the model is trained, it can be used directly, and results are obtained in seconds. The third advantage is that restoration quality can be improved by using larger and more diverse datasets for training [4].

Along this work, DL-based SR methods are applied to medical images. In order to completely understand how a CNN is able to enhance an image's resolutions, it is necessary to first understand what DL is and how CNNs work. The next section is intended to provide this information.

3 DEEP LEARNING

3.1 Brief History

The term *'Intelligence'* derives from the Latin 'intelligere', to comprehend or to perceive. Throughout the course of the history a variety of definitions have been given to this term, including the capacity for logic, learning or problem solving. A general approach commonly accepted nowadays is the ability to acquire and apply knowledge and skills.

Not just its definition but everything related to the term intelligence has always had attached important questions such as '*Can we create intelligence?'*. People have long imagined machines with human abilities. The ancient Greek philosopher Aristotle could already think of an automated world in his work '*Politics*' as an unreachable fantasy:

> *"For suppose that every tool we had could perform its task, either at out bidding or either itself perceiving the need [...] Then master craftsmen would have no need of servants nor master slaves*" [19].

However, Aristotle was not so wrong and, in fact, nowadays Artificial Intelligence (AI) does exist. This term accounts for every machine that has the ability to act somehow as a human, either using logic, perceiving information, solving problems, etc. The ability to process information, learn from it and make decision is called Machine Learning and consists of an important branch of AI. Arthur Samuel was one the pioneers that proposed this new technique. He proved that a machine programmed to learn how to play the game of checkers plays it better than the person who wrote the programme. The machine just needed to know the rules of the game and to be given a sense of direction and it could learn to play in a short period of time [20]. The principles of ML can be applied in many different situations, simple tools such as e-mail filtering or face recognition that are present on a daily-basis are examples of applications of ML [21].

As it has been explained, the aim of this field is to provide the machine with the ability to perform as a human. While some tasks are hard for a human, tasks such as computing long mathematical equations do not pose any difficulty for a computer. However, other tasks such as classifying whether an image shows a cat, or a dog used to be impossible for a computer. An approach that promotes analysing the information as it was processed by the human brain has recently appeared to solve these challenges.

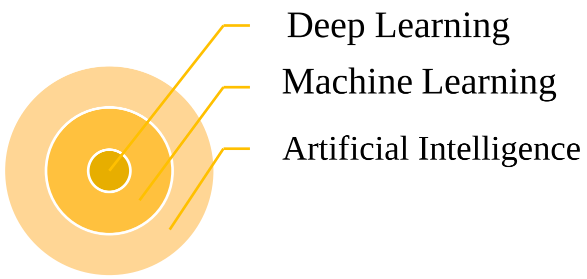


Figure 3. Venn Diagram showing how deep learning is a subset of machine learning, which is a subset of artificial intelligence.

Deep Learning is a branch of ML, which itself is a branch of AI. While ML focuses on learning, DL bases the learning on the use of artificial neural networks. These models consist of 'perceptrons' which are inspired by neurons and try to imitate them: perceptrons receive information and according to that information they may get activated, in which case the activated neuron, which is connected to many other neurons, will send an ‘impulse’ and so on [22]. A schematic of how a simple perceptron network is structured is shown in Figure 4.

The first neural network was proposed in 1943 by McCulloch and Pitts and consists of a binary classifier capable of recognizing two different categories given an input. However, this approach required manually actualizing the parameters [23]. Along the years different names have been given to the use of these networks, and implementations of them with methods such as backpropagation or loss function have resulted in what today we know as Deep Learning. The recent emergence of graphics processing units (GPUs) and the availability of large dataset to train networks, along with the implementation of the network with optimization methods have made it possible to train bigger and more efficient models [22].

3.2 Fundamentals of a Neural Network

In order to understand the processes behind a neural network, which will be later used in this project, the basic concepts are going to be explained in this section. As the name suggests, an artificial neural network is a net of artificially created neurons, so called *perceptrons*, which are connected to each other and communicate by transferring information. But how does the network perform the actual learning? How does a perceptron interpret the data and know when to be activated? All these processes are supported by different mathematical methods that are explained below.

1. Activation Functions

Figure 4 shows the basic architecture of a neural network. This network is very simple but is good enough to understand how they work for the purpose of explanation.

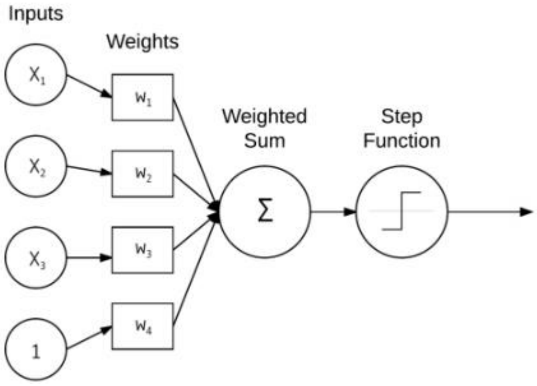


Figure 4. Schematic of a perceptron that receives 4 inputs, ($x_1 x_2, x_3, x_4$) that are multiplied respectively by ($w_1, w_2 w_3 w_4$). Weighted sum is computed prior applications of the step function [22].

Every input, called *x*, is connected to the perceptron via a specific weight *w*. The addition of every input multiplied by its weight gives the weighted sum, which is basically the function $\sum_{i=1}^{n} x_i w_i$. The output node consists of the activation function applied to the weighted inputs sum, in this case the NN is using the step function:

- $f(\sum_{i=1}^{n} x_i w_i) = \begin{cases} 1, & weighted\ sum > 0 \\ 0, & otherwise \end{cases}$

The concept of weights has been introduced, these are trainable parameters that will change throughout the learning process. But what if the weighted sum is far from zero, either too big or too small? This occurs often, and another trainable parameter is used to solve it. The bias, *b*, acts as a threshold that will determine if the perceptron is activated.

- $f(\sum_{i=1}^{n} x_i w_i) = \begin{cases} 1, & weighted\ sum > b \\ 0, & otherwise \end{cases}$

The aim of the activation function is clearly shown with this example, however there are several activation functions used nowadays. In fact, although the step function establishes clear rules for activation, newer functions have replaced it since the step function is not differentiable and therefore cannot be backpropagated [24].

The sigmoid activation function is usually a better choice since its saturation values approaches asymptotically and is differentiable everywhere, moving in the range (0,1) for every given input set. Calling the weighted sum x, the sigmoid function can be defined as:

- $f(x) = {}^{1}/_{(1+e^{-x})}$

Another very commonly used function is the Rectified Linear Unit (ReLU), also known as the ramp function, which follows the expression:

- $f(x) = max(0, x)$

Although the implementation of new activation functions has increased considerably since DL gained importance, the classic functions such as the ones explained above perform quite satisfactorily and are good enough for beginners. Some of the most common functions can be observed below.

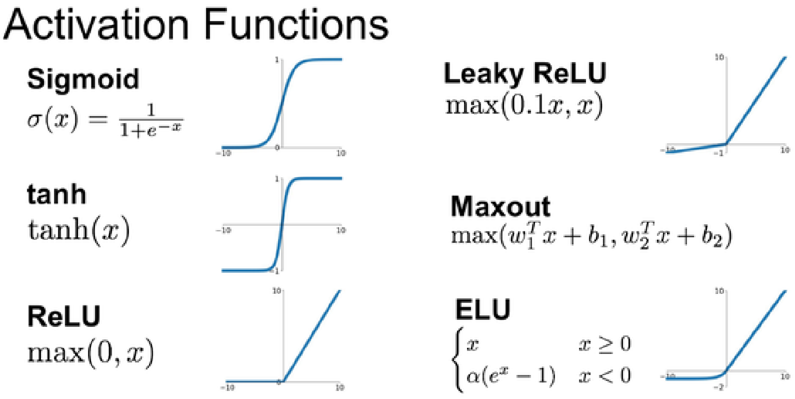


Figure 5. A selection of the most used functions in neural networks. From top to bottom and right to left; Sigmoid Activation Function; Leaky ReLU, a variant of ReLU that allows negative values; Hyperbolic Tanget; Maxout, that instead of sum the weighted values Hyperbolic Tanget; Maxout, that instead of sum the weighted values selects the maximum one; ReLU; and finally ELU, another variant of ReLU that performs better than Leaky ReLU [25].

The NN architecture consists of a number of different layers, each layer containing a determined number of perceptrons connected to both the previous and subsequent layer. For example, a network that receives 3 inputs could be made up of a first layer of 3 perceptrons, a second layer of 3 perceptrons and an output layer with one perceptron. This NN architecture is described as 3-3-1.

2. Neural Learning

The output given by the NN depends on the trainable parameters (weights and biases), which during the learning phase are changed according to the **Delta Rule**. The **epoch number** is the one determining the number of times the learning algorithm updates the trainable parameters.

Given a vector containing every data point *X* $(x_1, x_2 \ldots x_j)$, the learning algorithm would try to approximate the output to the true classification *D*. The first step is initializing the trainable parameters by randomly assigning small values. Then, the inputs vector is passed through the network, which is done by computing the dot product of the inputs and the weights, with the product being the input given to the activation function.

- $f\left(x_j \cdot w(t) + b\right) = y_j$

The results are then compared to the correct classification and finally, the delta rule is applied. This rule is basically the actual perceptron training procedure. It updates the trainable parameters according to the value of the comparison $(d_j - y_j)$, following the above expression and α being the **learning rate** [22].

- $w_i(t+1) = w_i(t) + \alpha(d_j - y_j)x_{ji}$

This learning algorithm is repeated during the training phase while the parameters change. Different methods exist to determine when to stop, the most common is to set a number of epochs that varies according to the network and dataset. Another method also used is to stop when the misclassification number has not changed in a large number of epochs [22].

3. Backpropagation

Among the training algorithms, *backpropagation* is the most widely used. It is composed by two phases, known as the *forward pass* and the *backward pass*. Note that data managed when training a NN is often too big to handle and can become inconvenient. Python libraries such as TensorFlow or Theano take charge of the numerical computation making machine learning easier and faster.

Although these computations are not required to be known and they are not shown during an actual learning process they are fundamental to understand the mathematics behind the NN. An example using a 2-2-2 neural network will be used for simplicity. The network architecture is shown in the figure below, composed by an input layer, a hidden layer and an output layer.

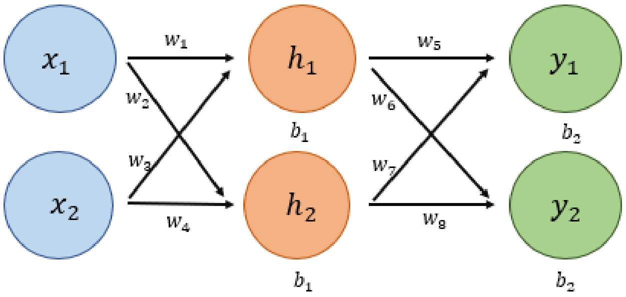


Figure 6. Diagram of the NN architecture. The first layer receives 2 inputs and communicates to the hidden layer, finally the output layer gives the outputs. This architecture contains three layers and 2 perceptrons per layer, defined as 2-2-2.

Forward Pass

As the name suggests, during the forward pass the operations occur from the input layer towards the output layer. As explained in the previous section the weighted sum for each perceptron is obtained and then the activation function is applied. Assuming input values are $x_1 = 0.05$, $x_2 = 0.10$, random biases $b_1 = 0.35$ $b_2 = 0.60$ and the initialized weights $w_1 = 0.15$ $w_2 = 0.25$ $w_3 = 0.20$ $w_4 = 0.30$ $w_5 = 0.40$ $w_6 = 0.50$ $w_7 = 0.45$ $w_8 = 0.55$.

Results for the first perceptron of the hidden layer:

$$h_1 = x_1 \cdot w_1 + x_2 \cdot w_3 + b_1 = 0.05 \times 0.15 + 0.10 \times 0.20 + 0.35 = 0.3775$$

$$outh_1 = f(h_1) = {}^{1}\!/_{1 + e^{-h_1}} = {}^{1}\!/_{1 + e^{-0.3775}} = 0.593269992$$

And, in the same way:

$$outh_2 = 0.596884378$$

Then the second layer perceptrons *communicate* to the output perceptrons, giving:

$$\begin{aligned} y_1 &= outh_1 \cdot w_5 + outh_2 \cdot w_7 + b_2 \\ &= 0.593269992 \times 0.40 + 0.596884378 \times 0.45 + 0.60 \\ &= 1.10590596 \end{aligned}$$

$$outy_1 = f(y_1) = {}^{1}\!/_{1 + e^{-h_1}} = {}^{1}\!/_{1 + e^{-1.10590596}} = 0.75136507$$

And again:

$$outy_2 = f(y_2) = 0.772928465$$

Now the total error is calculated. This is the difference between the output obtained and the correct classification *D*.

$$\begin{aligned} E_{total} &= \sum_{i=1}^{2} \frac{1}{2}(d_i - outy_i)^2 \\ &= \frac{1}{2}(0.01 - 0.75136507)^2 + \frac{1}{2}(0.99 - 0.772928465)^2 \\ &= 0.274811083 + 0.023560026 = 0.298371109 \end{aligned}$$

Backward Pass

This time the calculations go from the total error towards each trainable parameter. The algorithm updates every trainable parameter (weights and biases) so that the error is minimized. This is done by differentiating the total error by every parameter.

For example, to study how much w_5 affects the total error, $\frac{dE_{total}}{dw_5}$ should be calculated. Applying the chain rule this can be easily obtained:

$$\frac{dE_{total}}{dw_5} = \frac{dE_{total}}{douty_1}\frac{douty_1}{dy_1}\frac{dy_1}{dw_5}$$

Now, every difference is calculated separately:

$$E_{total} = \frac{1}{2}(d_1 - outy_1)^2 + \frac{1}{2}(d_2 - outy_2)^2$$

$$\frac{dE_{total}}{douty_1} = 2 \cdot \frac{1}{2}(d_1 - \ outy_1)^{2-1} \cdot (-1) + 0 = -(d_1 - outy_1)$$
$$= -(0.01 - 0.75136507) = 0.74136507$$

$$outy_1 = {}^{1}\!/_{1 + e^{-y_1}}$$

$$\frac{douty_1}{dy_1} = outy_1(1 - outy_1) = 0.75136507(1 - 0.75136507) = 0.186815602$$

$$y_1 = outh_1 \cdot w_5 + outh_2 \cdot w_6 + b_2$$

$$\frac{dy_1}{dw_5} = 1 \cdot outh_1 \cdot {w_5}^{(1-1)} + 0 + 0 = outh_1 = 0.593269992$$

Putting everything together:

$$\frac{dE_{total}}{dw_5} = \frac{dE_{total}}{douty_1}\frac{douty_1}{dy_1}\frac{dy_1}{dw_5} = 0.74136507 \times 0.186815602 \times 0.593269992$$
$$= 0.082167041$$

This number indicates how much w_5 affects the total error and is used in the updating phase to indicate how much w_5 should change. This is a gradient descent method since the system in moving downward the gradient. If the update moves too much sometimes the system ‘oscillates’ around the desired value and never reaches it. For this reason, another parameter, the learning rate μ, is included. By adding it, the updates are smoother; the lower the learning rate the slower the system moves down the gradient. This makes sure that the desired values are not missed out but on the other hand negatively affects the number of epochs needed and thus the training time.

$$w_5 = w_5 - \mu \cdot \frac{dE_{total}}{dw_5} = 0.40 - 0.5 \cdot 0.082167041 = 0.35891648$$

Finally, the same process is performed on every trainable parameter until the training phase is completed.

Note that although this is an extremely simple network it requires several computations just to update one weight. As previously mentioned, there are libraries that take charge of this. However, when working with deeper networks or big datasets, it can be too demanding even for them. A ‘trick’ often used is the use of mini-batches, instead of computing the gradient over the entire dataset, the data is separated in batches and the gradient is evaluated on every batch. Typically, batch sizes present values of 32, 64, 128 and 256.

3.2.1 Real Applications of a NN

Now the fundamental aspects of a NN have been explained it is time to show how it is applied to solve real problems. There are many websites such as Kaggle or Tunedit that provide datasets and challenges to analyse these datasets with DL.

Using an example, this section explains the workflow followed when solving a problem using a NN. The example used tries to determine whether a patient would attend their medical appointment or not. Around 30% of patients do not attend their medical appointment, what if Deep Learning could predict someone to no-show an appointment? The dataset has been obtained from a Kaggle’s challenge and contains 300,000 medical appointments with 15 variables (characteristics) of each patient. A simple Linear Regression is used to determine the ‘weight’ that each on these parameters has for the outcome, show-up or not.

Data Pre-processing

First step consists in processing the information so that the NN can properly read it. This is simply done by characterizing the variables with numbers, for example, set Monday to Sunday as 0 to 6 and Yes or No as 0 or 1. Then, the data must be separated into the input columns and output columns. The input are 10 selected feature variables and output is the Show-Up variable. Outliers that can affect the process should be removed in this first phase.

Then, data is split into train/test datasets, the conventional division is 67/33, 75/25 or 90/10 for training and testing datasets respectively. It is very important to establish that these sets are *independent.* Additionally, a third validation set can be included allowing to tune hyperparameters without using the testing set. The validation set usually takes around a 10-20% of the training dataset.

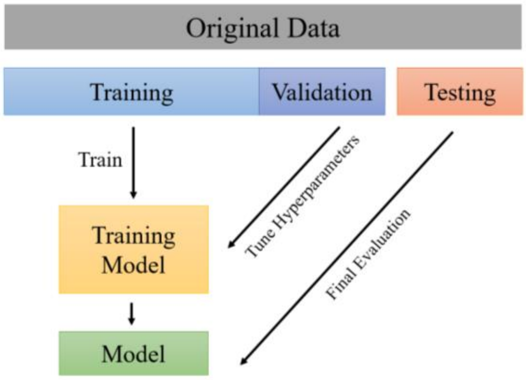


Figure 7. Scheme of the common distribution of the dataset into training, validation and testing sets and how each of them is applied during the training procedure.

Design, train and evaluate the network

After original data is divided and prepared to be introduced, the step consists in designing the architecture of the model. That is, how many layers and what kind of layers the model is made of. This is very dependent on the problem to be solved and models can range from the simplest model of 3 layers (input, hidden and output) to the deepest NN that has been design, made of a total of 152 layers. Before training the model, parameters such as the epoch number, batch-size and learning rate must be set.

Loss functions are used to evaluate how well the network is classifying, quantifying the difference between the network´s prediction and the ground-truth label. Trainable parameters are always adjusted in order to minimize this loss. When dataset classified has two possible labels, binary cross-entropy is used, and when there are more than two categories, categorical cross-entropy is used. Usually, loss is computed and plotted for every epoch along with accuracy, which is the percentage of right guesses obtained.

Methods used to tweak weights and biases are called **optimization methods.** The most classic is SGD (Stochastic Gradient Descent). So far, the most important implementations of SGD consist in very elaborated methods and include decay, momentum and Nesterov's Acceleration. Although explaining these in detail is out of the scope of this project, the basic idea of how SGD works could be summarized in minimizing the loss function down gradient, in order to find a minimum.

For the example being used, the loss-function selected is the binary cross-entropy, since there are just two possible outcomes, Show-Up and No Show-Up. A simple NN made of three layers is able to achieve an accuracy of the almost 70%.[4]

3.3 Convolutional Neural Networks

3.3.1 Introduction to CNN

The power of Deep Learning reached its maximum potential with the apparition of Convolutional Neural Networks, that is, NNs learning from images. During the last decade the amount of visual data produced is massively growing as well as the number of pixels contained [26]. Nonetheless, the query of a way to recognize and represent visual data dates from the late 50s when Hubel and Wiesel found that the visual cortex of a cat is structured by different types of cells able to respond to oriented edges and from there build a complex visual stimulus that represent an image [27].

Not so far from this discover, during the 70s computer scientist started to look for a way to represent objects. The answer was to reduce the complexity of objects by representing them with basic geometric models. This method was used similarly in Generalized Cylinder and Pictoral Structure, the first one was able to represent an object by using cylindrical shapes while the second one stablished main parts and then join them with lines [28] [29].

[4] This network architecture and its correspondent results are obtained from a public Kerner published in Kaggle [50].

It wasn't until 1982 when David Marr published 'Vision' and image processing started to be design in a similar way as it is our brain. That is, from small features, as edges or curves, a full 3D image is obtained [30]. Other challenges appeared as recognition of an object photographed from different angles and at different distance. This problem was firstly confronted by David Lowe, who proposed to identify the Local Scale-Invariant Features and match these patterns [31].

Although these new methods were promising, the lack of fast computers and access to big dataset were a remnant. In 2000 this problem was disappearing and big datasets as PASCAL, ImageNet or CIFAR appeared, challenging computer scientist to develop the best algorithm to classify datasets thousands of classes. It's here where in 2012 Alex Krizhevsky won the ImageNet classification challenge using the deep CNN shown in Figure 8. At this moment the boom of DL applied to image classification started [32]. Since then, the use of CNN has become widespread, an example can be face recognition which thanks to CNN has developed very quickly [33]. At first sight the architecture of AlexNet, or any CNN, could be seen confusing. During the next section an overview of the basic concepts of a CNN is be done.

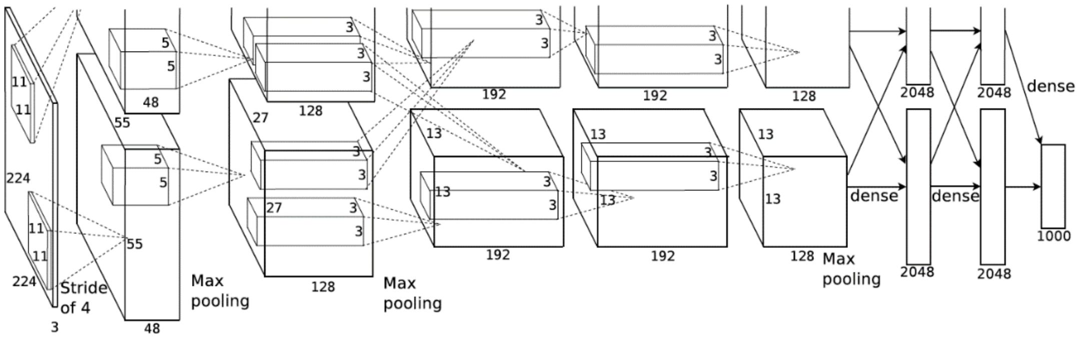


Figure 8. Illustration of the architecture of the AlexNet CNN that in 2012 won the ImageNet Classification Challenge. Model structure is made up by convolution layers, max pooling layers and fully connected layers [32].

3.3.2 Fundamentals of CNN

> *"Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers"* [34].

Let's first establish the difference between a fully connected (FC) and a convolution layer. The first ones are what has been used until now, in order to train a FC layer to recognize images, pixels should be stacked in a same row as if it was raw data. The problems that appear are that spatial information is lost, so same objects wouldn't be associated if they are at different positions and finally the computational cost increases.

To understand convolution, it's useful to think about a *big matrix*, which is the input image, and a *tiny matrix* that is called kernel or filter and it's a matrix containing weights. Convolution refers to the tiny matrix passing through the big one and performing a simple multiplication of the overlapped pixels. Figure 9 represents, quite accurately, the procedure followed at the first convolution layer of a CNN. In this case the big matrix is an image of

5x5 pixels[5]. It has to be taken into account that every image has **deep**, so they consist of three layers for the Red, Green and Blue Channels (RGB) that determine the final colour at each position. Therefore, the image is a (5x5x3) matrix while the kernel's size is (3x3x1)[6]. Convolution value is calculated by taking the dot product of the corresponding values every time the kernel and the image overlaps.

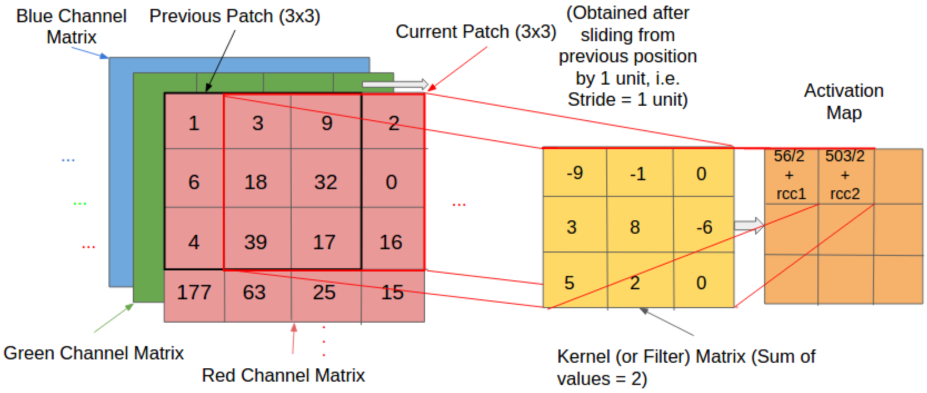


Figure 9. Illustration of how a convolution is realised. Every value of the (3x3) kernel is multiplied by its correspond pixel values in the matrix and then values are summed up. Rcc refers to the contribution from the green and blue channels [35].

Number of times this overlapping occurs is known and depends on the **stride**, this number stablishes how many pixels move, horizontal and vertically, every time a convolution is done. For example, in the latter example the stride was 1. The final size of the output matrix would depend also in this parameter, being S the stride, W the width of the image and F the width of the kernel, width of the output image follows the equation $W_{output} = \frac{W-F}{S} + 1$.

Problems appear when, for example, a stride of 2 is used with the image and kernel of figure X, since the pixels obtained at the final matrix are not an exact value. To overcome this common problem a simple solution is used: **padding**. This method consists in increasing the input image size by adding pixels, usually with 0 value and then it is called **zero-padding.** As it happened with NN, CNN does also use activation layers. Therefore, after the kernel has passed through all the image pixels, activation function maintains just the 'significant' values.

The results of all this calculation is that after a total of K different kernels are applied to the image, different features can be detected at different zones of the image. The final output matrix is then $WxWxK$. Following, a second set of kernels is applied to the output matrix and so on for every convolutional layer. The process performed by all the convolution layer is not another thing than the processed followed by cat's cortex, explained at the beginning of this section; the first set of layers detects edges, next layers use these edges to detect shapes and finally these shapes are used to detect high-level features such as animals, face expressions, vehicles, etc., in the highest layers [22].

[5] Although the entire image is not shown, in order to obtain an activation map (3x3x1) the actual size has to be (5x5x1).

[6] It is common to present more than one kernel at the same convolution layer.

As it can be observed in AlexNet architecture in Figure 8, a CNN is not just made up of convolutional layers. In fact, the last layers decompose the convolutional matrix into its pixels, as it was normal raw data, to obtain fully connected layer that would interpret the results. commonly one or two FC placed prior to the **softmax classifier**, which compute the final output probabilities for each class.

Although FC are always placed at the end of the network, there are some other layers placed between convolutions. In the AlexNet network **maxpooling layers** are applied between convolution layers. These layers are a class inside the Pooling Layers, a class used to reduce the size of an input volume by operating independently on each depth slice using the maximum or average function. However, the use of pooling layers is decreasing since, as it was shown before, convolutional layers do also reduce the size. At the moment, both layers are widely use, however, some library do already abrogate for eliminating its use. Well-known network architectures as ResNet just use convolution for down sampling [36]. Doing this, computational cost is reduced, and the important features highlighted.

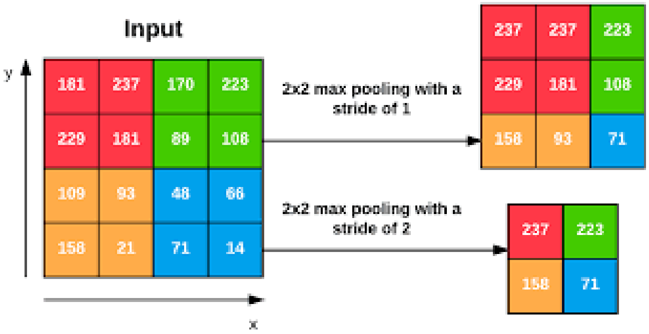


Figure 10. Illustration of down sampling with max pooling using a stride of 1 and 2 [13].

On the other hand, at some cases the objective is not to decrease but augment the size of the layer. **Deconvolution layers,** also known as transposed convolution, are designed to do up-sample the input. Figure 11 shows an example of up-sampling in a NN, filter refers to the kernel for the deconvolution. The parameters of the filter change for every epoch. The output contains copies of the filter weighted by input.

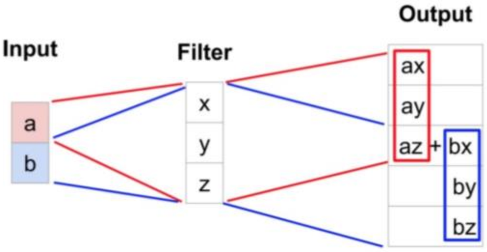


Figure 11. Scheme of Deconvolution layer in a NN [37].

Overfitting is the more common problem that appears when training a NN, it can be localized when the loss of the training dataset tends to descend while validation loss does not. As it exists overfitting also does underfitting, both concepts can easily be understood by looking at Figure X. To reduce overfitting exist **regularization** methods, that can be defined as *any method that increases testing accuracy perhaps at expense of training accuracy*. Some are applied to the loss function, implementing the model architecture or modifying the training data.

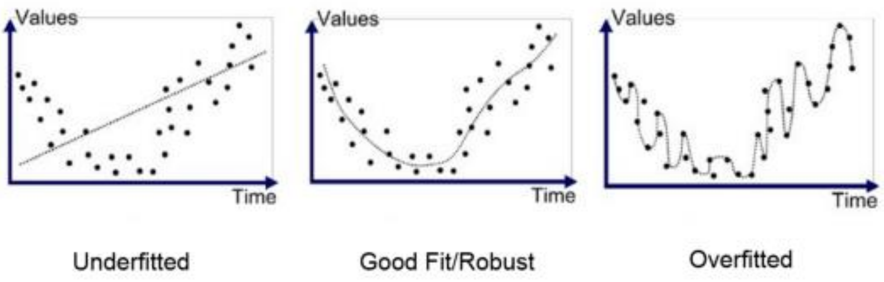


Figure 12. Illustration of underfitting, robust fitting and overfitting [38].

Dropout and **batch normalization** layer do both help to reduce overfitting. The firs type randomly disconnects inputs from the preceding layer to the next one while the latter basically consists in normalizing the output after every batch [39].

3.3.3 Real Applications of a CNN

The difficulty to understand visual data by computers has speed the use of CNN to classify images. The introduction of well-known benchmark data was a key point for CNN to develop all its potential since it allowed to evaluate the progress of object recognition. One of the most influential dataset is the PASCAL Visual Object Challenge, composed of 20 object classes and around ten thousand of images per category. This 'boom' drove to the creation of more ambitious datasets as the well-known ImageNet, which included 22 thousand of categories and more than 14 million images. Another significant benchmark dataset is the CIFAR-10, which includes 60 thousand of images belonging to 10 different classes. During the last year the benchmark data that can be found has increased greatly.

As more datasets appeared, so did established architectures of CNN. Important networks as GoogLeNet, LeNet, VGGNet have provided good results and its use has expanded. However, the study of this networks is out the scope of this work.

The use of CNN has spread to an important number of fields, including clinal applications. The first use is dated in 1990, where a CNN was used to classify tissue of mammographs to be either mass or normal tissue [40]. Classification of medical images went further and in 2016 breast tumours could be classified between benign and malignant [41].

In the next example, a dataset called HAM10000, '*Human Against Machine',* that contains 10 thousand images of 10 different skin cancers is used. As in section 3.2.1, data is divided into train, test and validation data, being the train/test division 80/20 and the validation set the 10% of the training images. All the images were reshaped into dimensions (70,100,2) and data augmentation was performed to prevent overfitting. The features, that is, the input data given, is the image itself, while the target, that is, the desired output, is the cancer type.

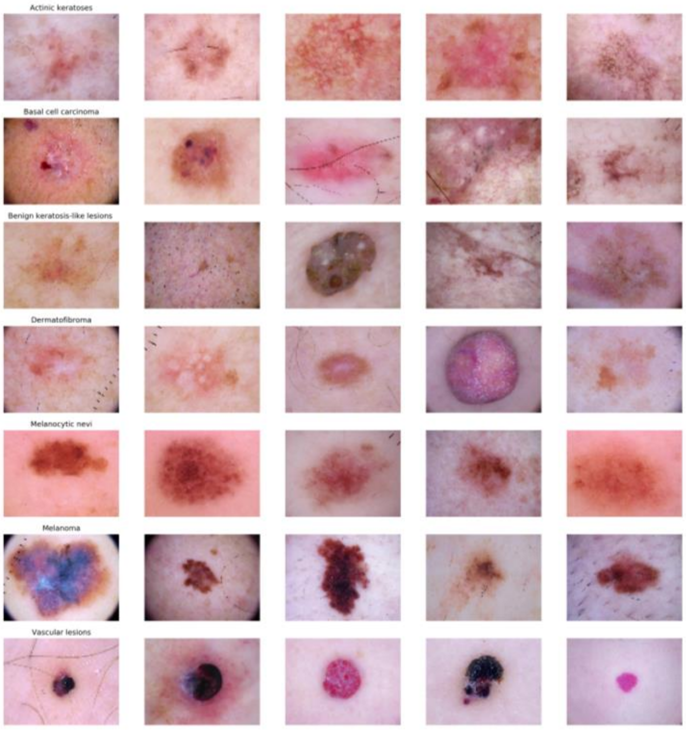


Figure 13. A selection of five images for each of the different cancer skin of the HAM10000 dataset. By order: Actinic keratoses, Basal cell carcinoma, Benign keratinosis-like lesions, Dermatofibroma and Melanocytic nevi.

Model architecture follow the structure showed below. Two convolution layers with ReLU activation are first applied, then data is down sampled with max polling and dropout is applied to reduce overfitting. This procedure is done a second time and next, data is flattened so that it's applied to a fully connected dense layer. Finally, after a last dropout layer a softmax classifier gives the final output.

[[Conv2D-> ReLU]*2 -> MaxPool2D -> Dropout]*2 -> Flatten -> Dense -> Dropout -> Out

The optimizer chosen in this case is Adam (Adaptative Momentum Estimation), a more elaborated manner to update the weights than SGD designed for stochastic optimization. It uses parameters as the mean of the gradient weights and its variance to update the weights [42]. This optimization also provides **momentum** to the learning process, which basically consists in increase the strength of updates for the weights whose gradients point in the same direction and decrease it for those who switch direction [43].

The network is then trained for 50 epochs using a batch size of 10. Batch size is used to implement the optimization method, and data will be presented by batches to the network at a time to perform backpropagation. In this case, for around 7.200 training images the steps per epoch are 7,200/10=720 [22].

Finally, the model is evaluated along the training process. The optimal results are obtained at the end of the process, resulting in a validation accuracy of 78.55% and validation loss of 0.64. Accuracy and los values along all the training process are shown below.

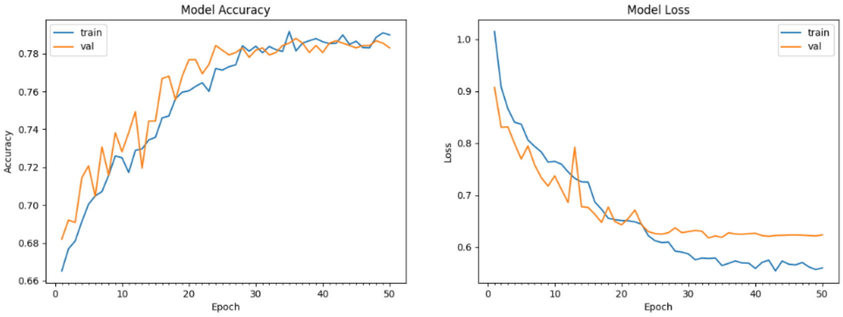


Figure 14. First figure shows the accuracy along every epoch during the training process and the second one for the loss. The blue lines represent values for the training set and the orange for the validation set.

Although the use of CNN in medicine is wining importance, there're still challenges to overcome. First, medical images are no so easy to gather so that a benchmark dataset can be created. In addition, classification of the different images is very subjective and only specialists can determine the diagnostic of every image. As for a human eye it is easier to differentiate between a panda and a cat than between cancerous or benign tumour, it will be harder to train a network to perform correctly this classification.

Another factor that makes it harder to achieve images is the high cost, economic and temporal, of obtaining CT or MR images. Furthermore, these images usually have low quality and it is hard for a CNN to detect features.

Convolutional networks are also applied to medical images in the field of segmentation and, although until now it has been restricted to 2D segmentation new approaches as the V-Net are currently performing volumetric segmentations [43].

4 MATERIALS

4.1 Data

The data used in this work was provided by The Osteoarthritis Initiative (OAI), an eleven-year longitudinal cohort-study sponsored by the National Institutes of Health. The goal of this study is to provide resources to better understand, prevent and treat knee osteoarthritis, one of the most common causes of disability in adults.

4.1.1 OAI Background

The object of the study consists in documenting the natural evolution of knee osteoarthritis. The spectrum studied includes subjects with established osteoarthritis (OA), with end-stage disease, with early/preclinical disease and risk subjects.

Almost five thousand women and men with ages between 45 and 79 were recruited and enrolled at five clinical centres in four U.S. cities. The OAI participants are split into three cohorts. The progression cohort consists of participants with symptomatic knee OA at the beginning of the study. The incidence cohort consists of participants who do not have symptomatic knee OA at the beginning of the study, but who have a high risk of developing knee OA. The normal control cohort consists of participants who do not have knee symptoms or radiographic evidence of tibiofemoral knee OA at the beginning of the study.

To study the progression, it was stablished a total of four annual planned follow-up visits. Joint imaging biomarkers, magnetic resonance imaging and radiography, and biochemical and genetic markers from blood and urine are collected at baseline and all follow-up visits. It was obtained data from 431,000 clinical and imaging visits, and almost 26,626,000 images.

This repository has been used in more than 400 published research manuscripts. In order to obtain access to this repository, researches have to request for access and once access in conceded, create an account.

4.1.2 Imaging Sequences for the Knee MRI

The core knee imaging acquisition methods are described briefly in this section. A large variety of measurements of structural OA pathology can be derived from these images. The OAI Steering Committee defined a core set of measurements that will be obtained from the images and made available as part of the public data release. Raw images are also be available to the research community for additional measurements.

Due to the large extension of studies, Study 538 Osteoarthritis Initiative (OAI) 18-Month Follow-Up Visit (V02) was selected. This study contains MRI images from a total of two hundred patients, each patient presenting 8 different MRI sequences. Relevant information about the image stacks contained by each sequence is presented in the table below, for more detailed information about each series and the acquisition protocol refer to Appendix B. OAI MRI Acquisition.

No.	Scan	# slices	Size (pixels)	Slice Thickness (mm)

1	**Localizer (3-plane)**	23	(512,512)	5
2	**COR IW TSE**	~ 35	(384,384)	3
3	**SAG 3D DESS WE**	160	(384,384)	0.7
4	COR MPR SAG 3D DESS WE	76	(384,384)	0.7
5	AXIAL MPR SAG 3D DESS WE	76	(384,384)	0.7
6	**SAG IW TSE FS**	37	(444,444)	3
7	**COR T1W 3D FLASH WE**	80	(512,512)	1.2
8	**SAG T2 MAP 120mm FOV**	~ 200	(384,384)	3

Table 2. Different series taken at each acquisition, along with the number of slices, the size of the slices and the pixel spacing for each series.

To guarantee homogeneity and consistency between the images and due to the large amount of data, only one sequence, SAG 3D DESS WE, was selected. The reasons this sequence was considered the most convenient are the next; it present the highest number of slices, which is very important since the project will reduce by a scaling factor of 2, 4 or 8; it present the lowest slice thickness, that is, the better resolution; finally, is one of the more consistent, as the table shows some of the sequences do not present the same amount of slices depending on the patient.

4.2 Tools

4.2.1 Software

All of the scripts for this thesis were written in Python, a programming language developed under open-source license. The codes were written and saved in Jupyter Notebook, an open-source web application designed to share documents containing live code, equations, visualizations and text.

Because Python is an open-source language, there exists many libraries that do not come with the Python but can be downloaded and implemented in Python. There is a huge variety of libraries designed for very different purposes. The reason Python was chosen is a specific library, called Keras, that allows to easily design neural networks. All the libraries used are described in the table below.

Keras	A high-level neural network Application Portal Interface (API) that runs on top of other libraries like Theano or Tensor Flow. These ones contain the algorithms needed in Deep Learning.
Pydicom	This package is very powerful when working with medical images stored in dicom format. It allows to read, open and modify these images.
OpenCv	Open Source Computer Vision Library collects state-of-the-art computer vision and machine learning algorithms.
ScikitImage	As OpenCV, this package provides with algorithms used in image processing.
Matplotlib	This library allows to create plots and show images, in addition it also reads, saves and modifies these figures.
NumPy	This is one of the most fundamental libraries for Python. Its more important applications are a powerful N-dimensional array object.
H5py	This package allows the user to store huge amounts of numerical data and easily manipulate them. Data is stored in H5 file extension.

Glob	Glob is used to find patterns that match, that is, when the program needs to find names with matching pattern spread in different directories.
Math	Math module provides with complex mathematical function.
os	This module allows to read and write files, manipulate paths and all the lines contained in a file. It also allows to check existence, create and eliminate files and directories.

Table 3 . Collection of all the python modules used in the project.

4.2.2 Hardware

The hardware used during this thesis are an AsusTP300L Intel Core i3 4GB, connected to a GPU NVIDIA TITAN X (Pascal) with 12 GB of memory.

5 SRCNN TO ENHANCE SINGLE IMAGES

The final objective of this project is to obtain HR MR images starting with LR images stored in DICOM format. In order to do so and to guarantee the correct functioning of the CNN, the first part of this project consist in the obtention of a SRCNN able to enhance the resolution of single images. Once the structure, parameters and procedure are completely optimized, the neural network is adapted so instead of receiving an image format it receives the DICOM volume and enhances it. This last part is discussed in section 6.

5.1 Methods

The process followed can be split in three parts, as three is the number of the main codes used: *build_dataset.py*, where the training and validation data is prepared; *train.py*, where the architecture, parameters and hyperparameters of the network are modified; finally, *evaluate.py*, where the network is tested, and results obtained.

5.1.1 Preparation of the training data

Given a dataset of images, already divided in train and test sets, the code process them image by image.

Both datasets are composed of LR images, which are the input to the SRCNN, and HR images, which are the ones the network will use as ground-truth. To obtain LR images from HR ones, every image is down-sampled by a scaling factor and then recovered to its original size using bicubic interpolation. The following process is applied image by image, to every pair of LR and HR images.

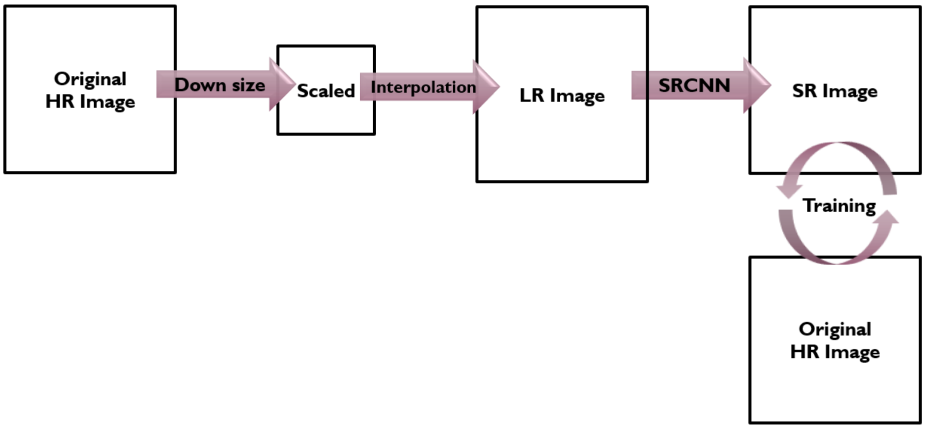


Figure 15. Workflow followed along the process. The original HR image is down-sampled and recovered to its original size, resulting in a LR image. This image is given as input to the SRCNN that will enhance it, producing a Super Resolution (SR) image. The objective is that the SR image is equal to the HR image, so during the training process the original image is used as ground-truth.

The input of the network is a one-channel image; therefore, the first step consists in moving from an RGB[7] image to a single channel. This is done by transforming every image from RGB to YCrCb, a colour model that separates intensity and colour information. Y is responsible for luminance information while Cb and Cr store, respectively, blue-yellow

[7] RGB is the standard format in which colour images are defined, where R stands for Red, G for Green and B for blue.

and red-yellow chrominance. The input for the network, that is, the channel to be enhanced, is the Y channel. This channel, with values between 0-255 is normalized to 1. After the output Y is enhanced and denormalized, it is combined with the remaining channels to recover the RGB format.

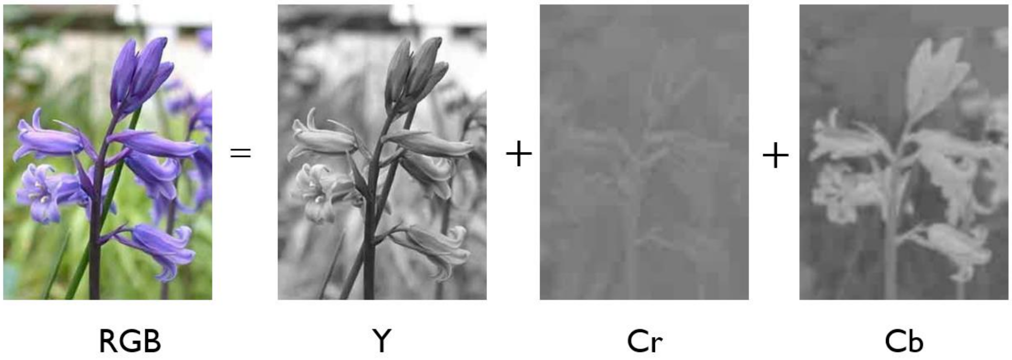


Figure 16. YCrCb coulour model separates the image in luminance (Y), chormaticity red-yellow (Cr) and chromaticity blue-yellow (Cb).

Next step consists in obtaining small patches of 32x32 pixels from the entire image. This technique speeds up the training process and provides diversity [44]. For images of the training set the patches consists in a mesh covering almost all the image, while the patches taken from validation images are positions randomly selected. Figure 15. Workflow followed along the process. The original HR image is down-sampled and recovered to its original size, resulting in a LR image. This image is given as input to the SRCNN that will enhance it, producing a Super Resolution (SR) image. The objective is that the SR image is equal to the HR image, so during the training process the original image is used as ground-truth. provides an example where the training patches are obtained by creating a mesh and 30 validation patches are obtained randomly.

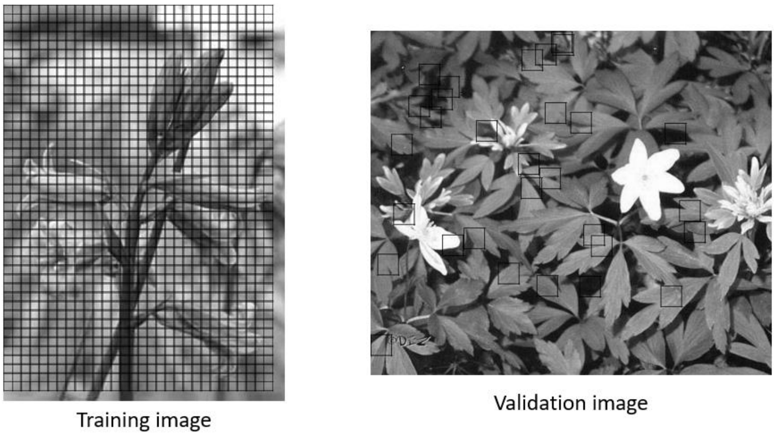


Figure 17. Patches from training and validation image.

The pair of LR/HR patches from the training and validation sets are stored in two different files, both in H5 format, called *test.h5* and *train.h5*.

5.1.2 Train the network

The network used is the SRCNN proposed by Dong. et al in 2014 [4]. Although many implementations have been proposed since this one appeared, the results obtained using the SRCNN are good enough. Therefore, due to its simplicity and good performance, SRCNN was chosen for this project. It consists in an end-to-end mapping between LR and HR images. Being Y the LR image and X the HR image, the goal is to recover an image F(Y) that is as similar as possible to the ground-truth X. F(Y) is the image enhanced by the network, that is, the SR image.

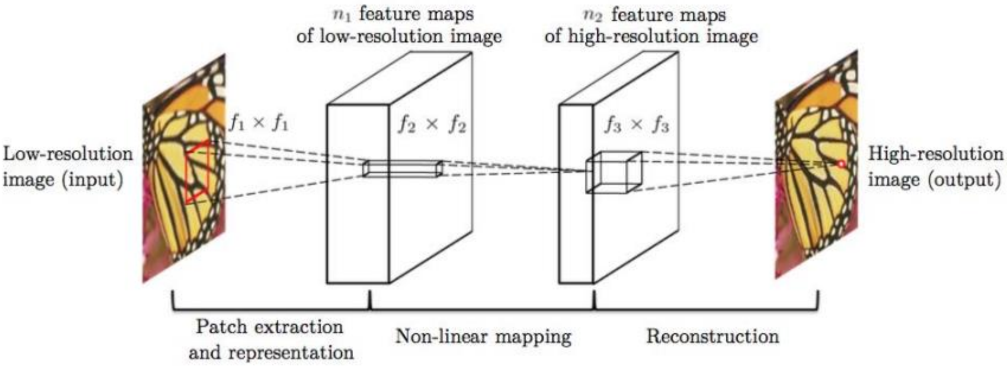

n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
Low-resolution image (input)
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

Figure 18. SRCNN proposed by Dong et al. [4]

As the figure above shows, the SRCNN consists of three layers. Each layer performs a different operation:

1) **Patch extraction and representation.** The first layer consists in a set of $n_1$kernels with size f_1xf_1, that will convolve through the image taking patches with size f_1xf_1. This results in $n_1$feature vectors that are representations of every patch[8]. Being W_1 the weights of the first layer and B_1 the biases, the activation function used is ReLU[9].

$$F_1(Y) = max\,(0, Y \times W_1 + B_1)$$

2) **Non-linear mapping.** The second layer convolves n_2 kernels with size f_2xf_2, mapping each of the n_1 feature vectors into a n_2-dimensional one. This process maps the low-resolution feature vectors to a representation of a high-resolution patch. Again, ReLU activation function is used.

$$F_2(Y) = max\,(0, F_1(Y) \times W_2 + B_2)$$

3) Reconstruction. The third convolutional layer produces a high-resolution image, by aggregating the high-resolution feature vectors. The result must be as similar as possible to the ground-truth image. It follows the formula:

$$F_3(Y) = F_2(Y) \times W_3 + B_3$$

The number of kernels for every layer is $n_1 = 128$, $n_2 = 64$ and $n_3 = 1$, with size $f_1 = 9, f_2 = 3$ and $f_3 = 5$.

Mean Square Error (MSE) is the loss function selected to train the network. That is, the MSE between the output SR image and the original HR image dictates how good the performance is. This loss is minimized employing Adam optimizer, which takes steps to

[8] Note that every patch created during the preparation of the training set is now treated as a entire image. Patch extraction in the first layer refers to taking smaller patches from the input patch through convolutions.

[9] More information about activation functions can be found in section 3.2.

the negative of the gradient in the direction of local minima. These steps modify the weights, W, and biases, B, for every epoch. The learning rate for the optimization is 0,0003.

Different trainings are realized for scaling factors of 2, 4 and 8. Also, the network is trained with *17 Category Flowers Dataset,* a public dataset, and with images from OAI stored in png format. The latter dataset does not require to change values to YCrCb colour format since it is already an intensity image, with just one channel. The set of natural images contained 180 train images and 20 validation images, while the OAI knees dataset contained 45 and 15.

5.1.3 Evaluation Methods

Once the training is done, the weights and biases for each layer are stored in a H5 file. Although the network trained just receives as input patches 32x32 sized, the weights and biases can be load into a new network which is exact to the one of the training, but able to accept any input size.

In order to evaluate the performance of the SRCNN, three different methods have been used. Two of them are quantitative methods, peak signal-to-noise ratio (PSRN) and Structural Similarity Index Measure (SSIM), and one qualitative, based on Visual Comparison. In the three cases the protocol was the same; to compare both the input and the output of the network with the original image, being the input the interpolated image and the output the SR image supposedly enhanced by the SRCNN. Interpolation has always been done using bicubic interpolation in python. Figure 19 represents the evaluation workflow followed. Notice that interpolation is done one single time and gives one single measure for PSNR and SSIM, however, the SRCNN is trained usually along 70 or 100 epochs, giving one set of results each time.

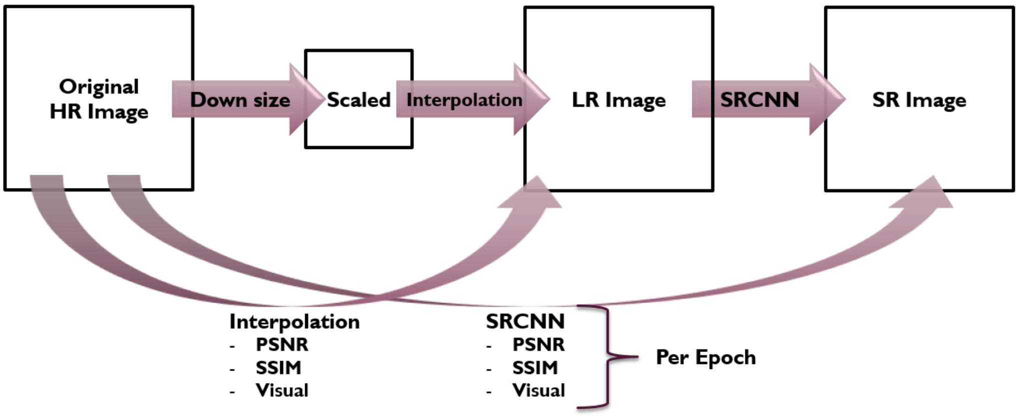


Figure 19. Worflow followed for evaluation. The image interpolated with bicubic interpolation is compared to the original one. Then, the interpolate image pass through the network and the output is evaluated. Same input (interpolated image) will result in 100 different outputs for a training process with 100 epochs.

PSNR

The first evaluation method, peak signal-to-noise ratio, is commonly used as a measure of the quality of images. It measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representations [45]. It is commonly expressed in terms of the logarithmic decibel scale and measured in dB, and it is defined as:

$$PSNR = 20 \times \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) = 20 \times \log_{10}\left(\frac{2^n - 1}{\sqrt{MSE}}\right)$$

where MSE is the mean square error[10], that is, the loss function, and MAX_I is the maximum value of each pixel. Normal images usually have bit-depth of 8, so its maximum pixel value is 255. However, when working with medical images maximum value is $2^{16} - 1$.

Since PSNR is inversely proportional to MSE, as the network is trained the loss function will work to minimize the MSE and the PSRN therefore will increase. When computing PSRN of the same image, MSE is zero and PSNR value is set to 100 dB.

SSIM

The second method also uses quantitative metrics. It is called Structure Similarity Index Measurement (SSIM) and it quantifies image quality degradation by assessing similarity between two images. It was originally defined to improve traditional methods as PSRN and MSE, however, they are normally use together. SSIM is defined as a weighted combination of three image parameters, the luminance (l), the contrast (c) and the structure (s).

$$SSIM(x, y) = \left[l(x, y)^{\propto} \cdot c(x, y)^{\beta} \cdot s(x, y)^{\gamma}\right]$$

The individual comparison functions are:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c\,(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

where μ corresponds to the mean of the signal intensity and σ to the standard deviation. $C_1 = (k_1L)^2$, $C_2 = (k_2L)^2$ and $C_3 = \frac{C_2}{2}$ where $k_1 = 0.01,\ k_2 = 0.003$ and L is the dynamic range of the pixel values [46]. This parameter can be easily obtained in Python since it is implemented in the library Scikit-Image.

Note that comparing the same image using this methods, PSNR value obtained is 100dB and SSIM would be 1. Therefore, PSNR values range from 0 to 100dB and SSIM from 0 to 1. Also, note that these terms are not related since they evaluate different aspects of the image. In fact, during the training process it happens that interpolated images present bigger PSNR than the SR image while SSIM was higher for the SR image.

Visual Comparison

[10] Notice that MSE is the loss function used, therefore the PSNR will be enhanced along the training, as the network looks for the minimum MSE.

The last method is used to evaluate qualitative the difference between an image and its ground truth by Visual Comparison. It consists in a simple subtraction of the images where each pixel value of the image that is being compared is subtracted from the pixel value that occupies the same position in the ground-truth image. In this case, the optimal output would be a black image[11] and structures in the original image lost during the reconstruction process are easily distinguished. Visual Comparison works well when using big scaling factors since it is when more information is lost and edges and structures that have lost resolution can be detected.

5.2 Results

A total number of 6 different SRCNN are trained. There are two datasets, one of natural images (*17 Category Flowers)* and one with knee images extracted from OAI's MRI. Foe each dataset, networks are trained with scaling factor of x2, x4 and x8.

5.2.1 SRCNN trained with natural images

The figure below shows the values for SSIM and PSNR along a training process of 100 epochs. The network is trained with 180 images and validated with 20 images. Note that the value for the interpolation is kept constant, since the interpolation image is the input to the network and does not change during the training process. As the SRCNN is learning, the results for the output SR image increases.

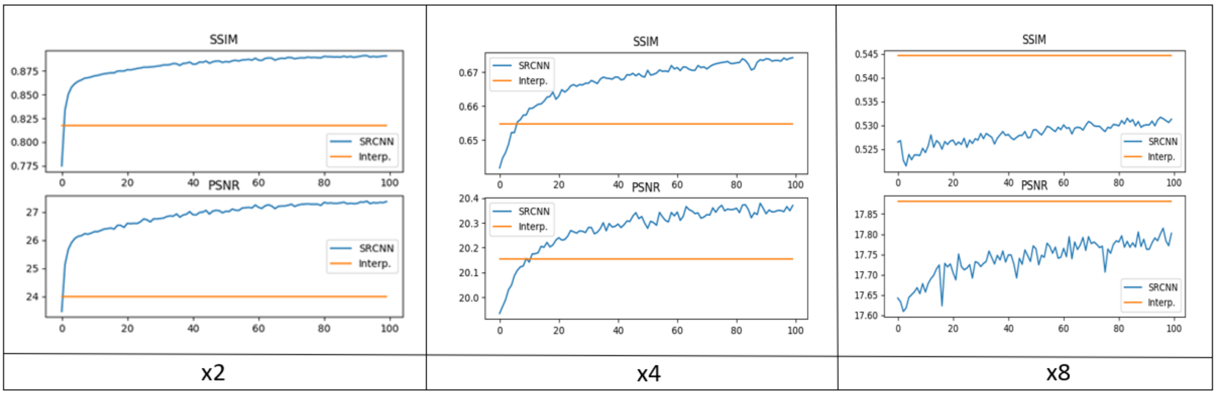


Figure 20. PSNR (dB) and SSIM for scaling factors of 2, 4 and 8 for natural images. The orange line represents the input image, obtained by bicubic interpolation, which is the same for all the training process. The blue line is the output of the network, that changes as the SRCNN learns.

In addition to values along the training process, the comparison of an image once the network is trained results in the following results. The network compared corresponds to a scaling factor of x4. For each row, the first image is the original one, the second one the enhanced one (interpolated or SR) and the third one the subtraction of both. More results of these networks are shown in Appendix C. Results of natural images enhanced with SRCNN.

[11] Note that subtracting the same image, all the pixels in the Visual Comparison would have zero value, which is nothing but a black image.

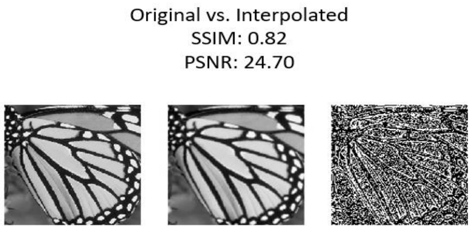


Figure 21. Visual comparison by subtracting the original image and the interpolated image.

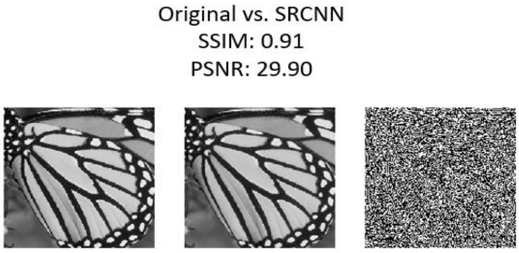


Figure 22. Visual comparison by subtracting the original image and the SR image.

5.2.2 SRCNN trained with knee images

The same process in done with the SRCNN trained with 45 knee images and 15 validation images. Epoch number for scaling factor 2 is 100 while for 4 and 8 is 50. The results are shown in the graphs below.

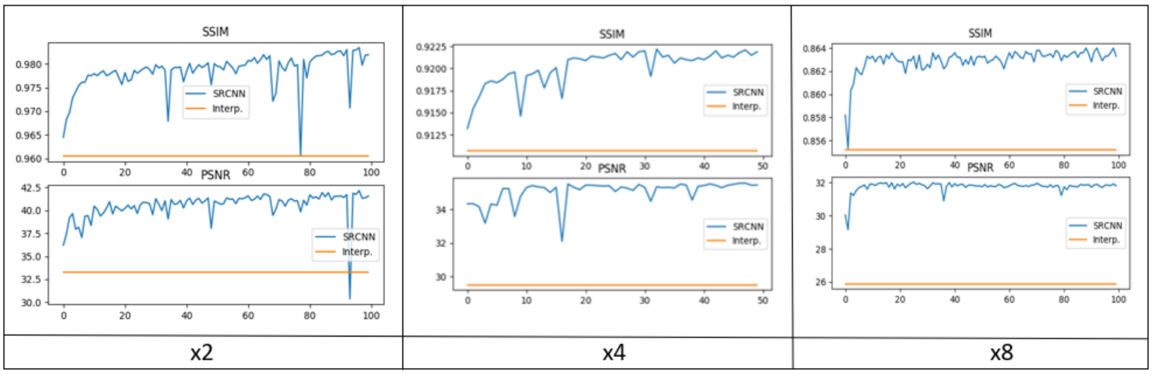


Figure 23. PSNR (dB) and SSIM for scaling factor of 2, 4 and 8 for knee images. The orange line represents the input image, obtained by bicubic interpolation, which is the same for all the training process. The blue line is the output of the network, that changes as the SRCNN learns.

Visual comparison by subtracting knee images doesn't not provide relevant information for any scaling factor. Same happens by just visualizing the knee images.

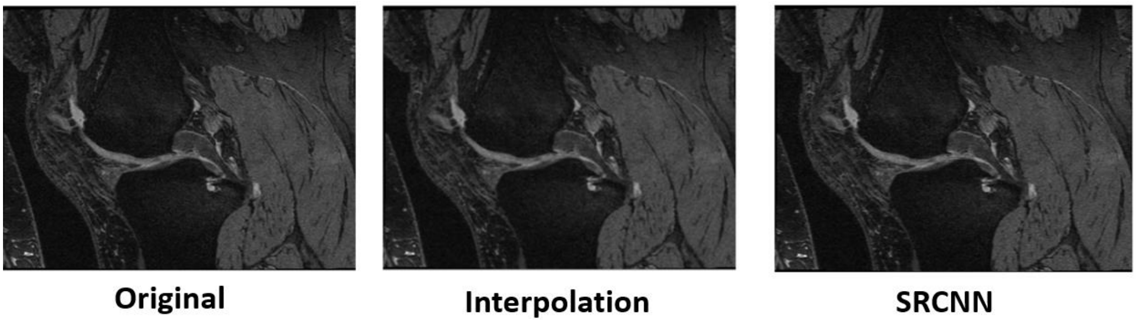


Figure 24. Original, interpolated and SR image of a knee. Scaling factor is 4.

5.3 Conclusion

The objective of this first approach is to build a solid workflow, verify the network outperforms conventional methods and finally extract conclusions and useful information to apply in section 6 and. It has been proved that resolution can be enhanced with SRCNN, outperforming conventional methods such as bicubic interpolation. Although the project presents good results, some appreciations have to be taken into account for the next part of the project.

Analysing the behaviour of every network as it is training (Figure 20 and Figure 22), it can be observed that scaling factors of 2 and 4 can be reliable and provide good outcomes. However, for a scale of 8 the SRCNN tested with natural images is outperformed by bicubic interpolation. The reason this does not happen in the SRCNN tested with knee images is that the size of these images is significantly bigger and structures are better conserved. Since they are more conserved it is easier for the network to recover the details and a scaling factor of 8 does not modify the image so much.

Because with a scaling factor of 2 the image is barely modified, and with 8 the SRCNN is not always able to overcome interpolation, the scaling factor in section 6 is set to 4.

The curves describing the improvement for the SR image for natural images is smoother than for knee images and it does not present peaks. This is due to two reasons. First, the number of images in the training and validation sets is bigger for the natural images. To enhance the performance of the network is important to provide it with enough information. Second, the range of pixel values in the actual MRI images is very wide. However, in the process of converting the DICOM image into a PNG image format, the range of values is reduced to 0-255. These modifications alter the network performance.

In order not to loss information, the MRI volume should be introduced to the network in its original DICOM format and the output be also stored as a DICOM volume.

In conclusion, the SRCNN designed outperforms the method and improves as it is trained. When applied to medical images stored in DICOM volumes, the different range of pixel values must be taken into account.

6 SRCNN APPLIED TO DICOM VOLUMES

Once the SRCNN is set up and it is understood how it works, implementations have been made in order that it can be used with stacks of medical images stored in DICOM format. This section explains all the modifications and improvements made in order to enhance the resolution of the knee MRI obtained.

6.1 Methods

6.1.1 Working with DICOM format

Some things to consider when working with DICOM volumes are:

-**Intensity Channel.** Medical Images do not have RGB channels, just a single channel tells the intensity of each pixel. This means all the information is contained in one single channel. Since all the information goes through the network, it is more precise. The step of changing from RGB to YCrCb colour format is supressed.

-**Pixel Values in an MRI**. The first thing to have into account when working with the MRI volumes is that the number of white pixels is significantly lower than the number of dark pixels. The background and most of the structures correspond to dark pixels. In addition, the range of pixel values for every volume is very different; instead of going from 0 to 255, the minimum value is always zero, but the maximum can change considerably.

The image below shows the logarithm histogram for three different MRI volumes. Note that there are so few white pixels that they couldn't be appreciated in the no-logarithm histogram. Also, note that just for three volumes the maximum vales are very separated, with values of 1500, 1800 and 2175.

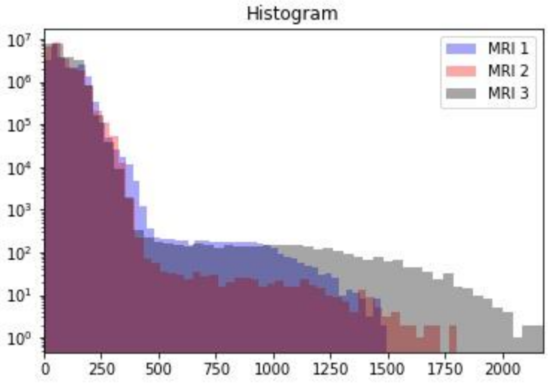


Figure 25. Logarithmic histogram of three MRI SAG 3D DESS WE series of three different patients. Data provided by OAI.

In the same way that in section 5 the pixel values were normalized to 1, for every MRI volume the pixel values are normalized from 0 to 500. Note that in the previous section values ranged between 0 and 255, but this time the range is wider, and information would be lost if values were normalized to a small range.

-**Dependency.** The stacks of images creating a volume are not independent of each other. The volume is a stack of images where consecutive images have a close relationship, since they belong to the same structure in the body. Therefore, the design of the network could be implemented in such a way that it also learns the relationship between the images. Different network architectures are discussed in section 6.2.2.

Recapitulating information about the MRI volumes provided by Osteoarthritis Initiative, they consist in stacks of 160 images taking in the sagittal direction. Every image has a size of 384x384 pixels, with slice thickness of 0,7mm. These images present high-resolution since they are the ground-truth. In order to obtain the LR images as in Section 5, the objective is to increase the slice thickness of every sagittal image, which means reducing the number of images. Starting with volumes with size 384 x 384 x 160, slice thickness can be reduced by a scaling factor of 4. The volume is modified to a size of 384 x 384 x 40, computing the average of every 4 sagittal images and then applying bicubic interpolation to recover the original shape.

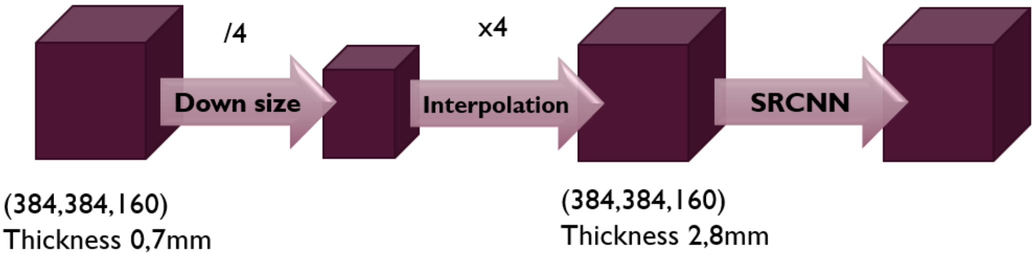


Figure 26. Workflow followed along the process. The original number of sagittal slices is reduced by 4 and then the original size is recovered with bicubic interpolation. This results in the slice thickness increasing with a factor of 4.

6.1.2 Training the network

As the main workflow followed is the one shown in Figure 19, it is modified according to the network architecture. Four different models are trained and tested:

1) **SRCNN**. The basic network shown in Figure 18 is used. Number of kernels per layer are, again, $n_1 = 128$, $n_2 = 64$ and $n_3 = 1$. And the kernels size, which is the same for every model, are $f_1 = 9$, $f_2 = 3$ and $f_3 = 5$. ReLU is the activation function and Adam the optimizer, with learning rate of 0,0001. Patches used for training have size of 32x32 pixels.

$$INPUT \;=>\; [CONV\,2D \;=>\; ReLU] * 2 \;=>\; CONV\,2D \;=>\; OUTPUT$$

Since SRCNN is 2D, it is trained with the images of the volumes. Sagittal slices present good quality after the volume its down-sampled. Looking to the images below, it is easier to notice the loss of resolution by looking at the coronal or axial slices than to the sagittal. For this reason, the images given as input to the SRCNN are the set of 384 sagittal or coronal images, with size of 384 x 160 or 160 x 384, respectively.

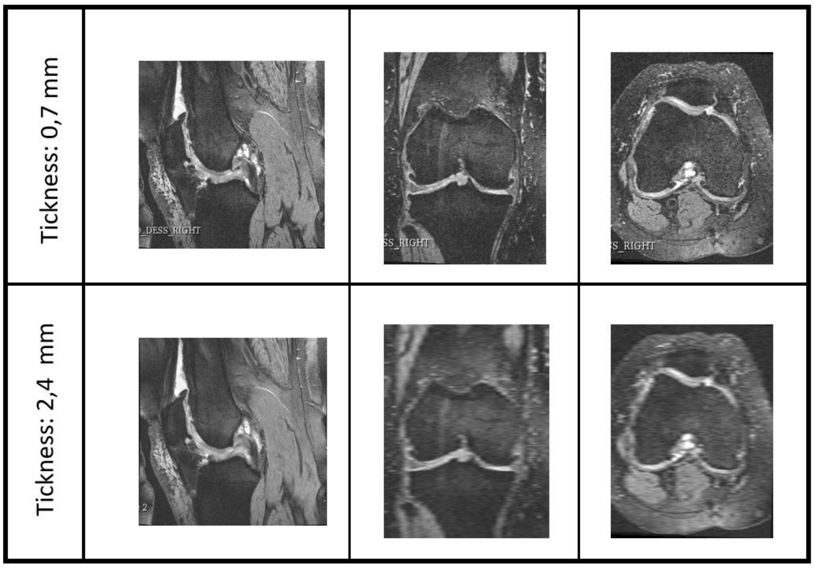


Figure 27. Sample of sagittal, coronal and axial images. Above for the original volume and below for the LR volume.

2) **3D SRCNN**. It has been mentioned that, one of the characteristics of working with DICOM volumes is that consecutive images are related between them. The information these relations could provide is lost if using a 2D network. Therefore, a new 3D model is designed. The architecture is very similar; every 2D Convolutional layer is changed to a 3D Convolutional layer, and a fourth layer is added.

A 3D input is therefore needed, this time the patch size used is of 64x64x64 pixels, because it contains more information. The trade-off between information processed and computational efficiency results in that the number of kernels per layer has to be reduced to is $n_1 = 32$, $n_2 = 16$, $n_3 = 8$ and $n_4 = 1$. Activation function is changed to PReLU, which is a form of Leaky ReLU[12]. Adam optimizer is used with learning rate of 0,00005.

$$INPUT\ => [CONV\,3D => PReLU] * 3 => CONV\,3D\ => OUTPUT$$

3) **SR-DCNN**. As the volume is interpolated to its original size and then passed through the network, there are errors performed by the interpolating method that the network is receiving. A good option is to let the network to perform the resize process, taking the volume back to its original shape. This is achieved by the Super Resolution Deconvolution Neural Network, proposed by Liu et al. in 2018 [47].

For example, the network can receive a volume with 384 coronal slices with size 384x40 and, with the first layer being a deconvolutional layer, increase the image to 384x160 and then it follows as the normal SRCNN. Therefore, the size of the patches is 64x16, and after the deconvolution layer they become 64X64. Number of kernels per layer are $n_1 = 1$, $n_2 = 64$, $n_3 = 64, n_4 = 32$ and $n_5 = 1$. First layer just presents one kernel, which corresponds to the deconvolution layer. That is because it is not extracting a set of feature vectors from an image but just increasing the size.

[12] More details about Leaky ReLU and other activation functions can be found in section 3.2.

$$INPUT \ => \ DEC\ 2D \ => \ [CONV\ 2D \ => \ PReLU] * 3 \ => CONV\ 2D \ => \ OUTPUT$$

4) 3D SR-DCNN. Finally, by combining both previous implementations the a 3D Super Resolution Deconvolutional Neural Network is used. The patches used for training the network are of size 64x64x16. Because of computational efficiency, the number of kernels is decreased again $n_1 = 1$, $n_2 = 16$, $n_3 = 8, n_4 = 4$ and $n_5 = 1$.

$$INPUT \ => \ DEC\ 3D \ => \ [CONV\ 3D \ => \ PReLU] * 3 \ => CONV\ 3D \ => \ OUTPUT$$

All the networks are trained with 8 different patients, that is, 8 different volumes, and validated with other 4 patients. However, since the size and shape of patches change for every network the final train and validation sets are different. Every network is trained during 50 epochs. Appendix D. Structures of the CNNs. provides a deeper illustration of each network and its layers.

6.1.3 Evaluation

Because of the different nature of the data tested, the evaluation methods have to be modified.

PSNR

The SSIM is discarded, since it is measuring how well the structure is maintained. Because all the structures are a knee, the SSIM does not change significantly and it does not provide any relevant information. Visual comparison by subtracting the images is also dismissed because it is just useful when important losses of information in the image happen. However, PSNR values are still used since they provide relevant information when comparing the volumes.

Normalized Cross-Correlation

Until now, the methods used measured pixel-by-pixel similarity between pixels in the same position. When the MRI volume passes through the network the intensity values of the pixels can vary significantly. Although the relationship between them and the information provided is improved, the similarly to the values of the original pixels can be affected. Cross-Correlation measures similarity not between pixels but between the full volumes by allowing the range of values for each image to be different. It focuses more in the hole volumes and similarity of the information contained by them. In order to so, normalized cross-correlation normalized with respect to the mean and standard deviation by following the next formula:

$$NCC = \frac{1}{NM}\sum_{x=0}^{N}\sum_{y=0}^{M}\frac{(A(x,y) - \bar{A})(B(x,y) - \bar{B})}{\sqrt{Var(A)\, Var\,(B)}}$$

Where A and B are the two images with size MxN, mean $\bar{A}$ and $\bar{B}$ and variances $Var(A)$ and $Var\,(B)$. Normalized cross correlation ranges from 0 to 1, with value of 1 obtained with images that are the same.

Clinical Evaluation and Questionnaire

Besides the quantitative metrics used, when working with medical images, and images in general, the most reliable source is the human eye. In order to assess the clinical importance of the image enhancement, a qualified physician from the traumatology department of the Hospital General Universitario Gregorio Marañón, determined the most important anatomical structures of the knee. The visual evaluation is made between the interpolation method and the network presenting the better quantitative metrics.

Then, a survey among personnel working in the Biomedical Imaging and Instrumentation Group (BIIG) of Universidad Carlos III of Madrid was made. A group of 5 subjects answered the survey, this group was composed of Biomedical Engineering students and graduates. The objective of this survey is to determine if important anatomical structures can be better visualized in the SR volume. The first question consisted in, at first sight, decide which volume presents better resolution among the SR-DCNN and the interpolated one. Then, subjects are asked to evaluate how well different anatomical structures can be visualized in each volume. The structures evaluated are the ones selected by the physician, and include: Anterior Cruciate Ligament (ACL), Posterior Cruciate Ligament (PCL), Internal Lateral Ligament (ILL), External Lateral Ligament (ELL), Patellar Cartilage (PC), Tibial Cartilage (TC), Femoral Cartilage (FC) and Meniscus. An example of a survey form and the guide provided to localize these anatomical structures can be found in Appendix E. Questionnaire Form. Appendix F. Guide to Localize Knee Anatomical Structures., respectively.

6.2 Results

PSNR and Normalized Cross-Correlation

The values for normalized Cross-Correlation and PSNR are obtained for every image along each one the volume axis for a total of 5 patients. The mean values obtained are plotted in the three graphs below. That is, 384 values for each coronal image, 384 for each axial image and 160 for each sagittal image. Values for the first and last images for each axis are not computed, since they only contain background and do not provide relevant information. While in the previous section the graphs (Figure 20. PSNR (dB) and SSIM for scaling factors of 2, 4 and 8 for natural images. The orange line represents the input image, obtained by bicubic interpolation, which is the same for all the training process. The blue line is the output of the network, that changes as the SRCNN learns. and Figure 22. Visual comparison by subtracting the original image and the SR image.) showed how the PSNR and SSIM increase as the network was trained, in these graphs the network is already trained. The values compared are the outputs of the different trained networks versus the input to the networks, the interpolated slice. This is compared for every image along every axis.

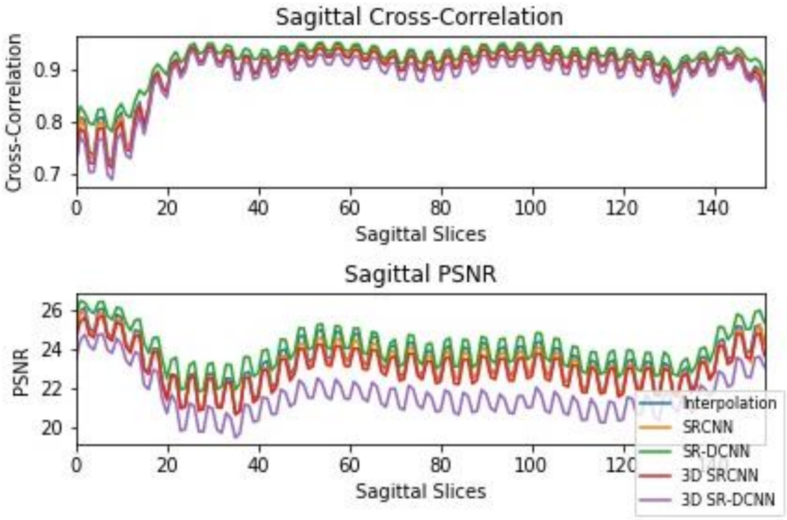


Figure 28. Normalized Cross-Correlation and PSNR (dB) for along sagittal axis for interpolated (blue), SRCNN (orange), SR-DCNN(green), 3D SRCNN (red) and 3D SR-DCNN (purple).

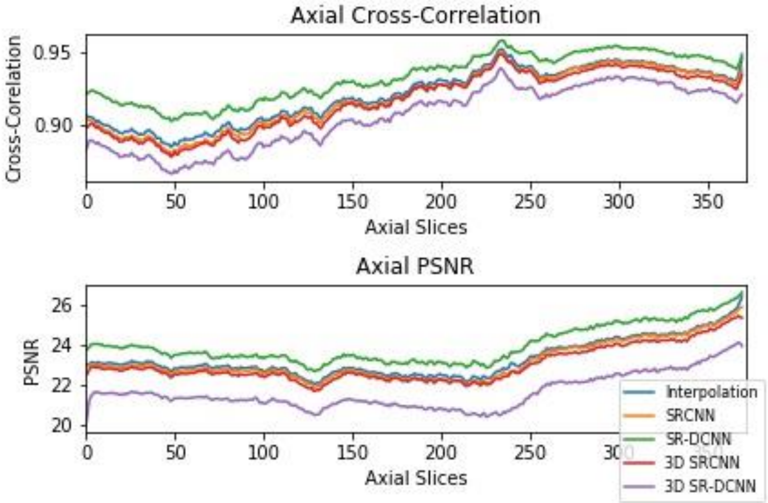


Figure 29. Normalized Cross-Correlation and PSNR (dB) for along axial axis for interpolated (blue), SRCNN (orange), SR-DCNN (green), 3D SRCNN (red) and 3D SR-DCNN (purple).

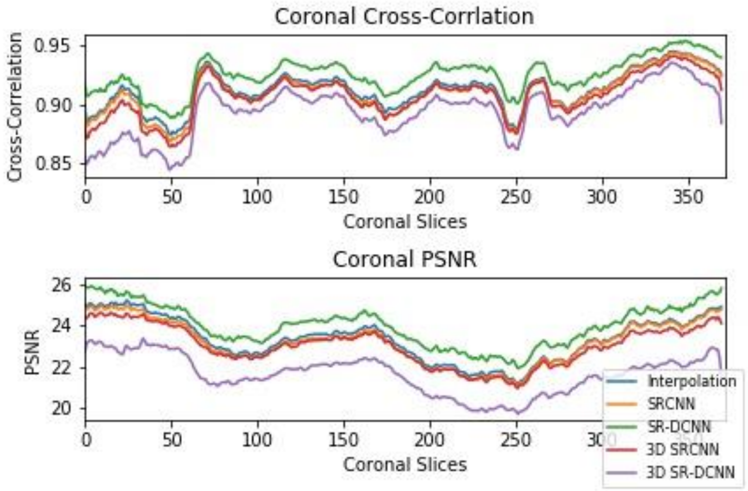


Figure 30. Normalized Cross-Correlation and PSNR (dB) for along coronal axis for interpolated (blue), SRCNN (orange), SR-DCNN (green), 3D SRCNN (red) and 3D SR-DCNN (purple).

The next table contains he means and standard deviations of the plotted values.

Method	PSNR (dB)	Normalized Cross-Correlation
Interpolated	23,33 ± 1,06	0,9130 ± 0,0280
SRCNN	23,18 ± 1,12	0,9103 ± 0,8001
SR-DCNN	**24,03 ± 1,04**	**0,9261 ± 0,0239**
3D SRCNN	23,00 ± 0,97	0,9146 ± 7,0738
3D SR-DCNN	21,63 ± 0,98	0,8959 ± 0,0316

Table 4. Mean and standard deviation of PSNR and Normalized Cross-Correlation.

Visual Evaluation

In addition to the quantitative metrics obtained, the volumes are also visualized in 3D Slicer. An example of a single coronal slice for each different network is shown below, together with the original slice.

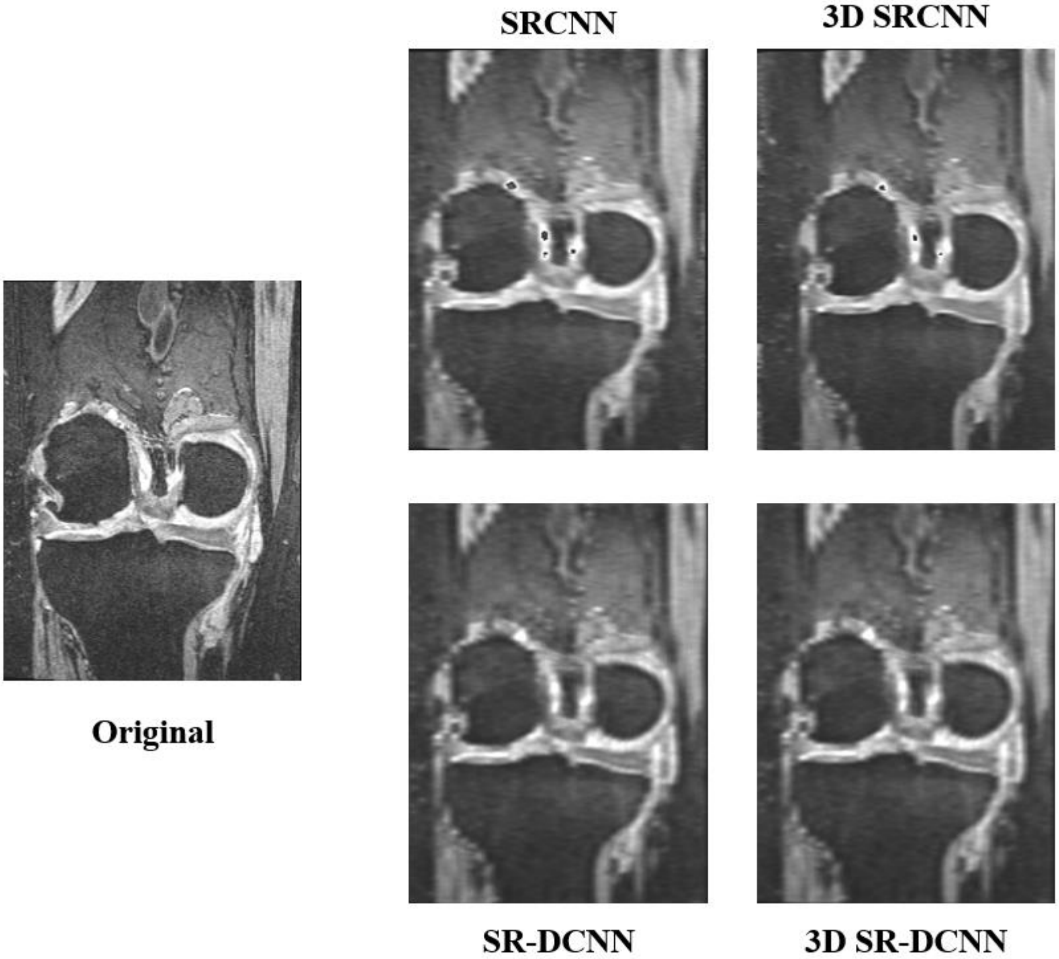


Figure 31. Coronal slice for original, SRCNN, 3D SRCNN, SR-DCNN and 3D SR-DCNN volumes.

As explained previously, the following visual is just performed to compare the conventional method, bicubic interpolation, and the network with the best results, SR-DCNN.

A physician specialized in traumatology from Hospital General Universitario Gregorio Marañón defined and evaluated the most relevant anatomical structures in the knee. For the Patellar Cartilage (PC), Tibial Cartilage (TC), Femoral Cartilage (FC) and Meniscus, it was established that the structures are better defined in the SR Volume, with significant differences. For the Anterior Cruciate Ligament (ACL) and Posterior Cruciate Ligament (PCL) the visualization also improved, but less significantly. Finally, Internal Lateral Ligament (ILL) and External Lateral Ligament (ELL) could not be visualized in any volume due to the loss of resolution. Because of this, ELL and ILL are disregarded for visual evaluation. Images of the different structures for every volume are shown below[13].

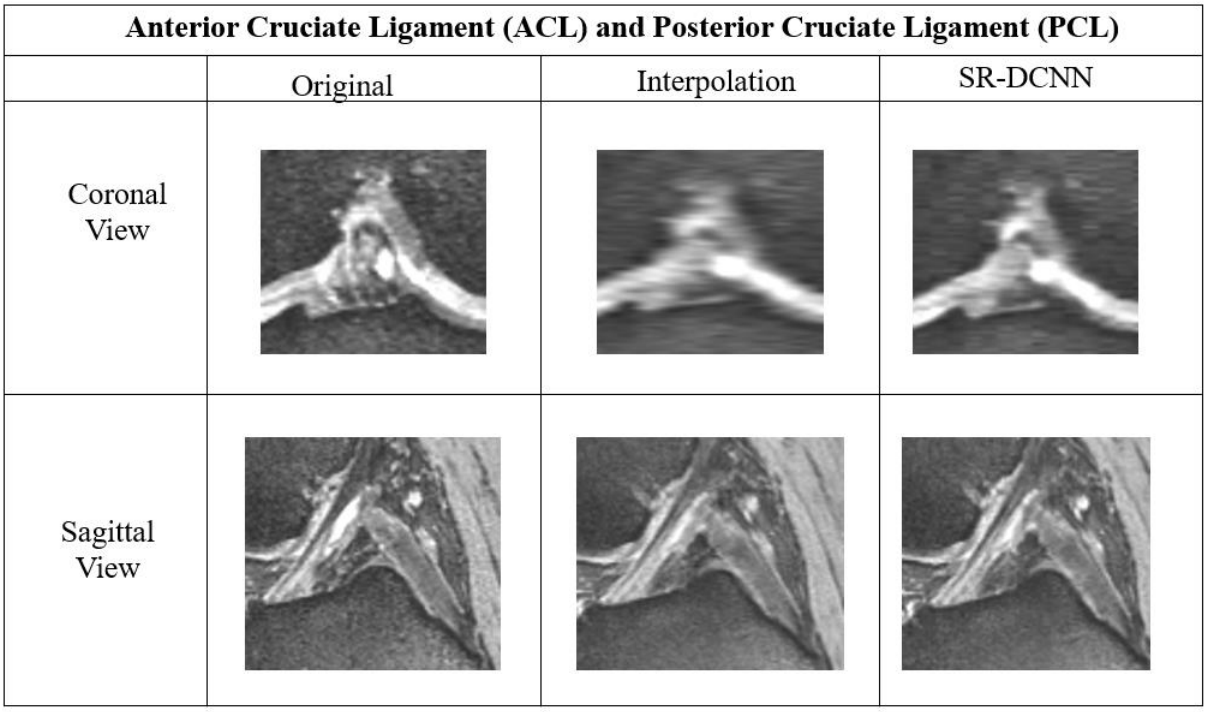

Anterior Cruciate Ligament (ACL) and Posterior Cruciate Ligament (PCL)
Original
Interpolation
SR-DCNN
Coronal View
Sagittal View

Figure 32. ACL and PCL for original, interpolated and SR volume.

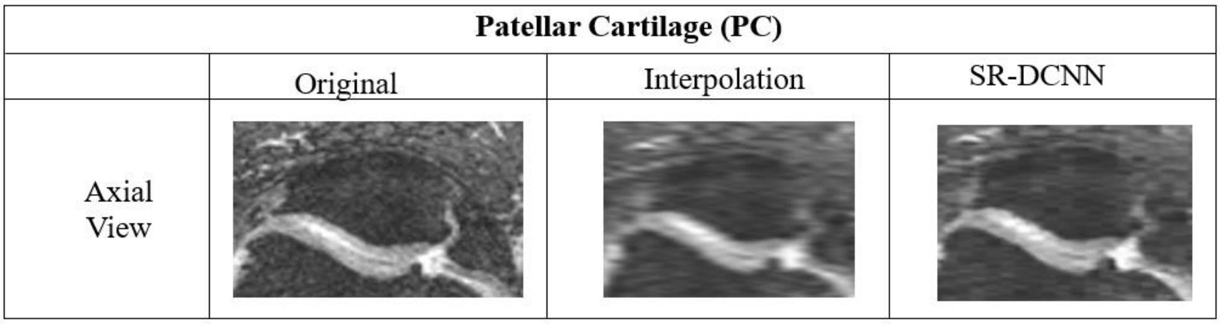


Figure 33. PC for original, interpolated and SR volume.

[13] Refer to Appendix F. Guide to Localize Knee Anatomical Structures. for a Guide to Localize Knee Anatomical Structures.

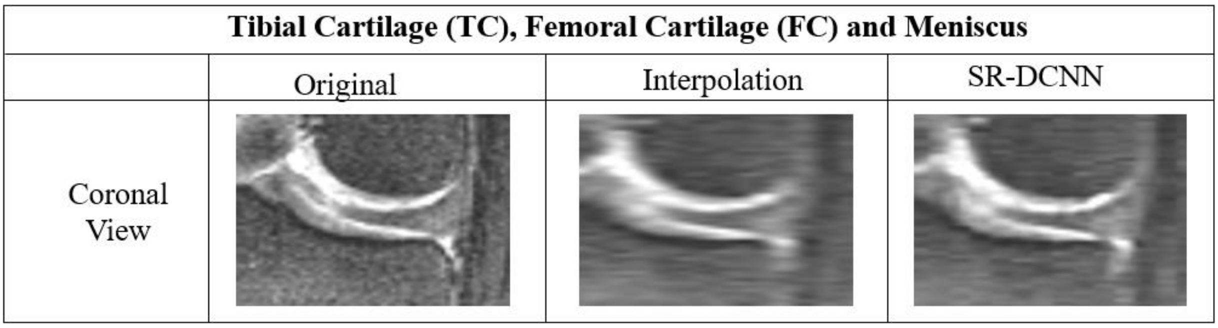


Figure 34. TC, FC and Meniscus for original, interpolated and SR volume.

The results of the questionnaire shown that 100% of the subjects agreed that, at first sight, the SR volume presents better resolution than the interpolated one.

The following table shows the results for visualization quality of the different anatomical structures. A value of 1 means that the structure cannot be observed and 5 that the structure is totally defined.

	SR -DCNN Volume	Bicubic Interpolation
APC and PCL	4,72 ± 0,44	4,36 ± 0,55
PC	3,93 ± 0,61	3,04 ± 0,99
TC and FC	4,28 ± 0,75	3,32 ± 0,90
Meniscus	4,16 ± 0,80	3,32 ± 0,93

Table 5. Evaluation of the visualization of different structures (1-5).

6.3 Conclusions

The values for PSNR and Normalized Cross-Correlation resulting when comparing the original volume versus the interpolated, SRCNN, SR-DCNN, 3D SRCNN and 3D SR-DCNN are shown in Figure 28, Figure 29 and Figure 30 and summarized in Table 4. The network using deconvolution clearly outperforms the rest of methods, while the one combining 3D and deconvolution is significantly worse. The remaining networks result in values similar to bi-cubic interpolation values.

Number of kernels

The values for the SR-DCNN network show that Deep Learning can be a good alternative versus interpolation, however it does not occur when 3D is introduced. For the SR-DCNN there are 1-64-64-32-1 kernels per layer. For the 3D SR-DCNN kernels are reduced and reaches the lowest values, with 1-16-8-4-1 kernels per layer[14].

There is a trade-off between the data the network is learning, since deconvolution and dimensionality (3D networks) are increasing the data the network learns, the kernels of the network are reduced. That is, there is more data to learn from, but less information can be extracted. A 3D SR-DCNN trained with enough kernels is likely to outperform bi-cubic interpolation and SR-DCNN.

[14] For every network, the hardware used does not allow to train the network with a bigger number of kernels per layer.

The same trade-off occurs for 3D SRCNN, since kernels are reduced to 32-16-8-1, the performance decreases. However, the reduction is not so significant, and the performance does not decrease so much.

Finally, the SRCNN performs worse than the SR-DCNN while the kernels are no reduced (128-64-1). In this case, the number of layers is reduced and here information is lost. However, the most important factor is that the input to the network has been interpolated. The SR-DCNN's first layer performs deconvolution, where the volume goes to its original shape inside the network. Introducing the volume already interpolated as input, the network is receiving an interpolation error.

Deconvolution vs. Interpolation

It can be slightly appreciated that, the results for the networks whose input is the interpolated volume there is a *blocky* artifact at the edges. This artifact disappear when the up-sampling is performed by the network with deconvolution, as it can be seen the edge is much smoother for SR-DCNN. Otherwise, deconvolution implies another artifact that was not before. It can be observed that, since the network generates 4 slices for each 1 given, there is noticeable step, every 4 slices, for the deconvolution volume.

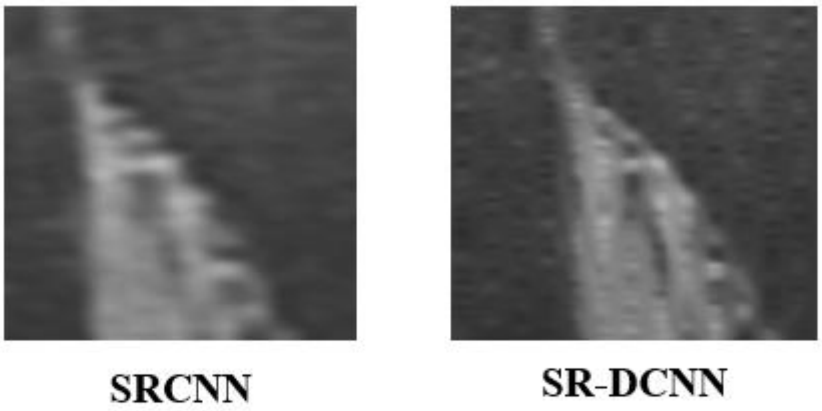


Figure 35. Zoomed area of the SRCNN and SR-DCNN slices shown in Figure 31.

There is a trade-off between interpolation and deconvolution since both generate some kind of artifact. However, PSNR and normalized Cross-Correlation show that the error is reduced when deconvolution is used.

Presence of Black Areas

In Figure 31 it can be observed that some networks created black areas in structures that tend to present very high values, that is, very white areas. The reason of this can be understand by looking at the histogram of Figure 25, that shows the number of high intensity values is very low. Due to this reason, the network does not have enough information to learn how to proceed these pixels.

This problem is solved by adding deconvolution. Although it could be seem not related, results reveal that, when the network up-samples the input, it also learns to maintain the white values in the areas where they are present. This also explains the best values obtained by the SR-DCNN. Although the 3D SR-DCNN has also learned to maintain the high pixel values, its small number of kernels per layer does not allow it to reconstruct properly the volume.

SR-DCNN vs. Bicubic Interpolation

Since SR-DCNN is the network with the best quantitative results, the volumes studied visually are the ones obtained with this one. Both, a physician and subjects participating in the survey, agreed that the SR volume resolution was better. Also, important anatomical structures can be better distinguished. While the different between structures rounds 1 point in Table X, it can be observed that the difference in APC and PCL decreases. As it has been mentioned, this is because the resolution in the acquisition axis does not decrease as much as in the other directions[15].

[15] This is shown in Figure 24. Original, interpolated and SR image of a knee. Scaling factor is 4..

7 DISCUSSION

SR-DCNN outperforms bicubic interpolation when comparting PSNR and normalized Cross-Correlation values. In addition, qualitative evaluation also supports that the volumes enhanced with the network present the better quality. Once the network is trained, the introduction of the network in the hospital workflow does not modify the acquisition time, since the enhancement performed by the network only takes 8.45 seconds.

The use of SR-DCNN enhances MRI resolution and facilitates anatomical structure recognition. However, because of security reasons the results of this network cannot be used yet for clinical disease evaluation and diagnose. It still needs of more trials and an approval to guarantee that structures generated by the network are reliable and that introduction of different patients won’t change the result.

Nevertheless, it’s use can be already implemented in clinical processes that do not imply a clinical assessment. This project proposes the utilization of the SR-DCNN in order to make easier segmentation tasks. This can help and speed up the design of 3D printed surgical guides. Additionally, this first implementation will help to assess the actual performance of SR when it is implemented in a clinical workflow.

7.1 Limitations

An important issue when training super resolution networks is that they only learn for a certain scaling factor, a certain part of the body and a certain acquisition modality. Even for MRI, different networks have to be trained for different series. Although the scaling factor is just a matter of time, the need of a certain dataset for every part of the body and modality is an important drawback. The use of the network it is limited to the availability of big datasets. For example, it will be easy to collect many chest CT and train the network with them, however, it will be harder if the network needs to learn from MRI of a hand.

Another limitation is the Black-Box Medicine, explained in Technical Standards section. The use of Deep Learning for medical purposes is still a controversial issue until the reliability of *all* the results is guaranteed.

7.2 Future Work

In order to enhance performance and provide more accuracy to the reconstruction process, elaborated, state-of-the-art SR networks should be trained. The use of more complex networks will probably eliminate the pattern that appears every four slices, explained in section 6.3 Also, a bigger dataset will improve the training process.

There is a solution to the scaling factor limitation, and a single network can be trained for different scaling factors. The idea was proposed by Lium et al. and it consists in, instead of training a different network for every scaling a factor, progressively up-scaling the input until it reaches the desired size [44]. Therefore, it only learns a x2 scaling and this one is performed several times.

Finally, this project encourages the application of SR to different body parts and image modalities and, once it is possible, the assessment of its performance once it is introduced to the clinical workflow.

8 ACRONYMS

SR: Super Resolution. Post-processing method by which

LR: Low Resolution.

HR: High Resolution.

ML: Machine Learning.

DL: Deep Learning.

AI: Artificial Learning.

NN: Neural Network.

CNN: Convolutional Neural Network.

SRCNN: Super Resolution Convolutional Neural Network.

SR-DCNN: Super Resolution Deconvolutional Neural Network.

PSNR: Peak Signal-to-Noise Ratio.

SSIM: Structure Similarity Index Measure.

ACL: Anterior Cruciate Ligament.

PCL: Posterior Cruciate Ligament.

ILL: Internal Lateral Ligament.

ELL: External Lateral Ligament.

PC: Patellar Cartilage.

TC: Tibial Cartilage.

FC: Femoral Cartilage.

9 APPENDIX

9.1 Appendix A. Spatial Resolution.

In order to assess the spatial resolution achievable by imaging sensors and imaging systems, it is common to use tests as “1951 USAF Resolution Test Target”. It consists in the acquisition of an image as the one shown below with the acquisition system to evaluate. The spatial resolution is determined by the biggest group of lines the system has not been able to detect as separated lines.

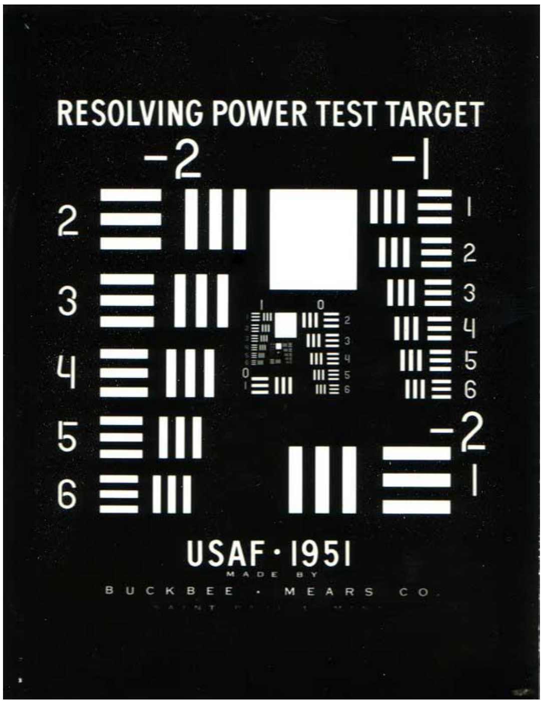


Figure 36. Scanned image of a glass USAF 1951 Resolution test chart [48].

9.2 Appendix B. OAI MRI Acquisition.

There are two stablished protocols for the MRI acquisition, one for each knee. The Right Knee Standard Exam consist in the following acquisitions, centre location and oblique angulation should be identical for similar acquisition planes. Left Knee Standard Exam is identical, expect that the Cor 3D T1 FLASH WE and SAG T2 MAP sequences are omitted. However, all participants with metallic implants in an eligible right knee should switch protocols.

MRI scan of both knees and thighs using a Siemens Trio 3.0 Tesla scanner and extremity RF coil positioned at R60.

No.	Scan	Duration in Minutes		
		R knee	**L knee**	**Total**
1	**Localizer (3-plane)**	0.5	0.5	**1**
2	**COR IW TSE**	3.4	3.4	**6.8**
3	**SAG 3D DESS WE**	10.6	10.6	**6.8**
4	COR MPR SAG 3D DESS WE	0.0	0.0	0.0
5	AXIAL MPR SAG 3D DESS WE	0.0	0.0	0.0
6	**SAG IW TSE FS**	4.7	4.7	**9.9**
7	**COR T1W 3D FLASH WE**	8.6	**-**	**8.6**
8	**SAG T2 MAP 120mm FOV**	10.6	**-**	**10.6**
	Total	38.4	19.2	**57.6**

Table 6. Breakdown of the time taken for each series, indicating the time each knee needed and the total acquisition time.

The following description of each of the series is an adaptation from "*MRI Procedure Manual for Examinations of the Knee and Thigh"* [49], a protocol for the obtention and processing of the sequences contained in the OAI dataset.

- Localizer 3-plane is a conventional three-plane localizer, where offsets for coil positions are pre-programmed into the corresponding exam sequences.
- COR IW TSE exam is intended for evaluation of joint alignment, cartilage morphology, osteophytes, the body of menisci, collateral ligaments and for the presence/Extent of subchondral bone cyst and attrition. This Coronal scan is a double oblique acquisition.
- SAG 3D DESS WE with water excitation is a single oblique acquisition, oriented such that the sagittal slices are perpendicular to a line tangential to the posterior surfaces of the femoral condyles. This acquisition will provide information for total joint cartilage thickness and volume. In addition, information about osteophytes, subarticular bone cysts and bone attrition, and possibly collateral ligaments will be available.
- COR MPR SAG 3D DESS WE is a Coronal multi-planar reconstruction of the Sagittal 3D DESS with Water Excitation. It does not require further acquisition.
- AXIAL MPR SAG 3D DESS WE is an Axial multi-planar reconstruction of the Sagittal 3D DESS WE.
- The Sagittal IW with Fat Suppression exam is for evaluation of the effusion volume, the anterior and posterior femoral and tibial osteophytes and for the presence / extent of subchondral bone cysts and attrition. This Sagittal scan is a single oblique acquisition,

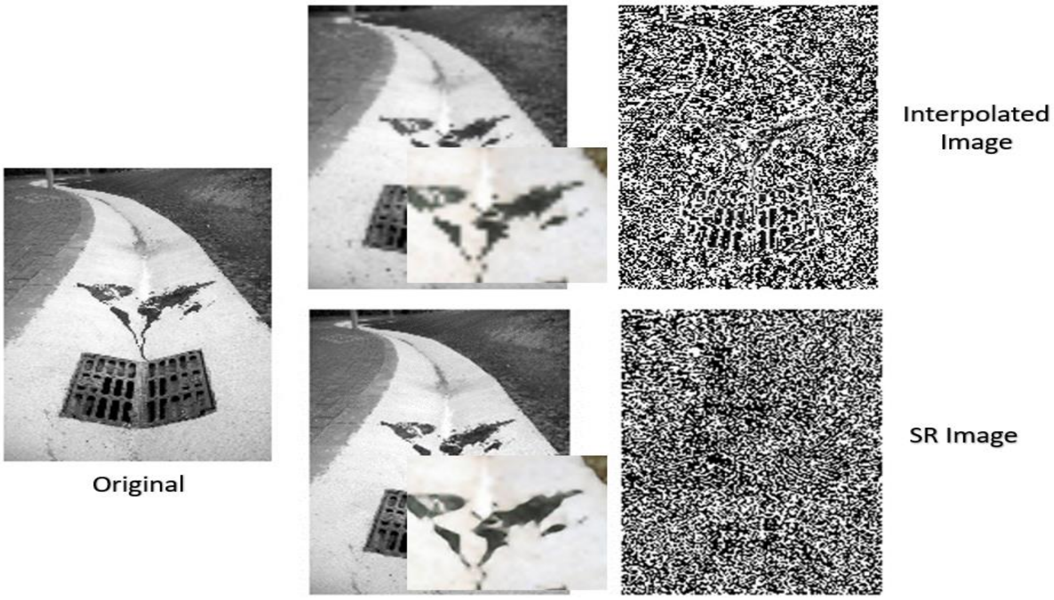
oriented such that the sagittal slices are perpendicular to a line tangential to the posterior surfaces of the femoral condyles

- The Coronal 3D FLASH with Water Excitation is a sequence commonly used for cartilage thickness measurements and volume segmentation. This 160mm FOV acquisition should fully cover the femoral condyles and tibial plateau. This Coronal scan is a double oblique acquisition.
- The Sagittal T2 mapping sequence is a small FOV exam intended to cover both the patellar and femoral / tibial cartilage only.

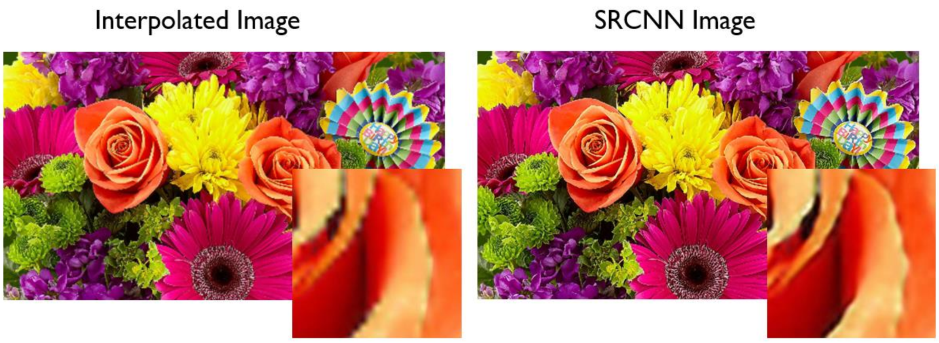
9.3 Appendix C. Results of natural images enhanced with SRCNN.

Images interpolated and processed with SRCNN are shown in this appendix, in order to provide examples for visual comparison and understand the performance of the super resolution network.

In this example, the image is reduced by a scaling factor of 8. After it is passed through the network the edges and structures look smoother. Also, in the resulting image of subtracting the interpolated image and the original image some structures can be appreciated. For example, the lines of the path or the holes of the sewer. This means that the edges describing them are distorted during the bicubic interpolation. However, since they do not appear for the SR image it means the SRCNN is able to recover these details.



For the next case, the scaling factor is 4. Although at first sight it would be harder to observe improvements in the SR image, by zooming the edges it can be shown that they are smoother and closer to reality in the SR image.



9.4 Appendix D. Structures of the CNNs.

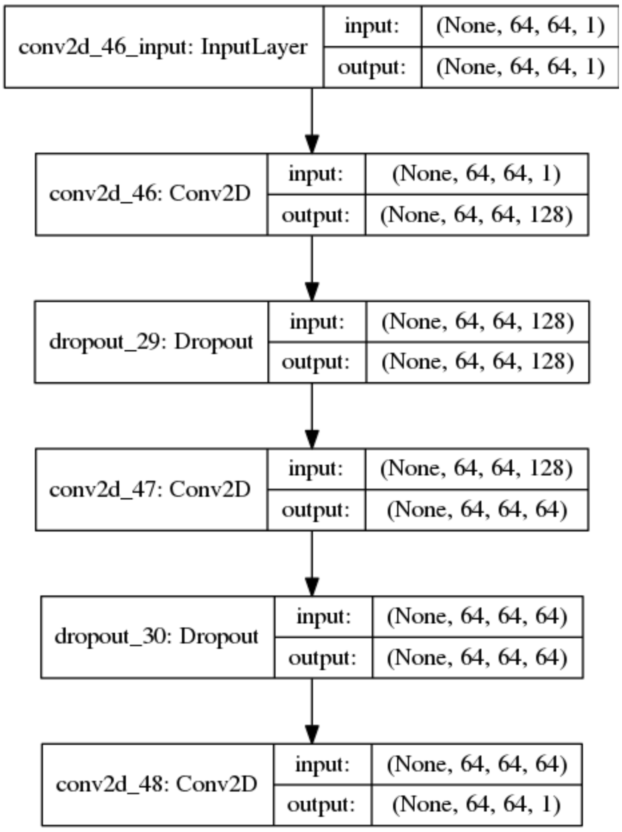

conv2d_46_input: InputLayer
input: (None, 64, 64, 1)
output: (None, 64, 64, 1)
conv2d_46: Conv2D
input: (None, 64, 64, 1)
output: (None, 64, 64, 128)
dropout_29: Dropout
input: (None, 64, 64, 128)
output: (None, 64, 64, 128)
conv2d_47: Conv2D
input: (None, 64, 64, 128)
output: (None, 64, 64, 64)
dropout_30: Dropout
input: (None, 64, 64, 64)
output: (None, 64, 64, 64)
conv2d_48: Conv2D
input: (None, 64, 64, 64)
output: (None, 64, 64, 1)

Figure 37. SRCNN.

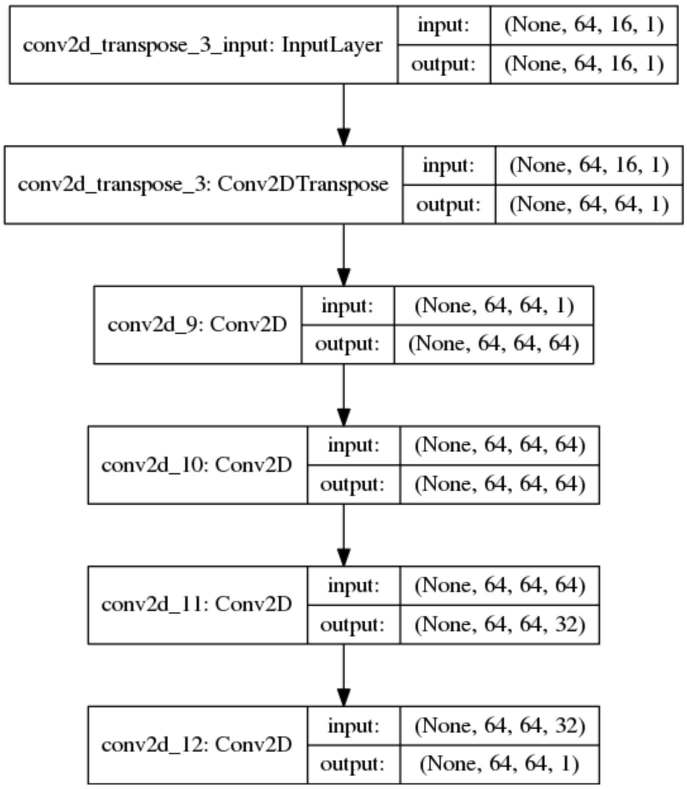

conv2d_transpose_3_input: InputLayer
input: (None, 64, 16, 1)
output: (None, 64, 16, 1)
conv2d_transpose_3: Conv2DTranspose
input: (None, 64, 16, 1)
output: (None, 64, 64, 1)
conv2d_9: Conv2D
input: (None, 64, 64, 1)
output: (None, 64, 64, 64)
conv2d_10: Conv2D
input: (None, 64, 64, 64)
output: (None, 64, 64, 64)
conv2d_11: Conv2D
input: (None, 64, 64, 64)
output: (None, 64, 64, 32)
conv2d_12: Conv2D
input: (None, 64, 64, 32)
output: (None, 64, 64, 1)

Figure 38. SR-DCNN.

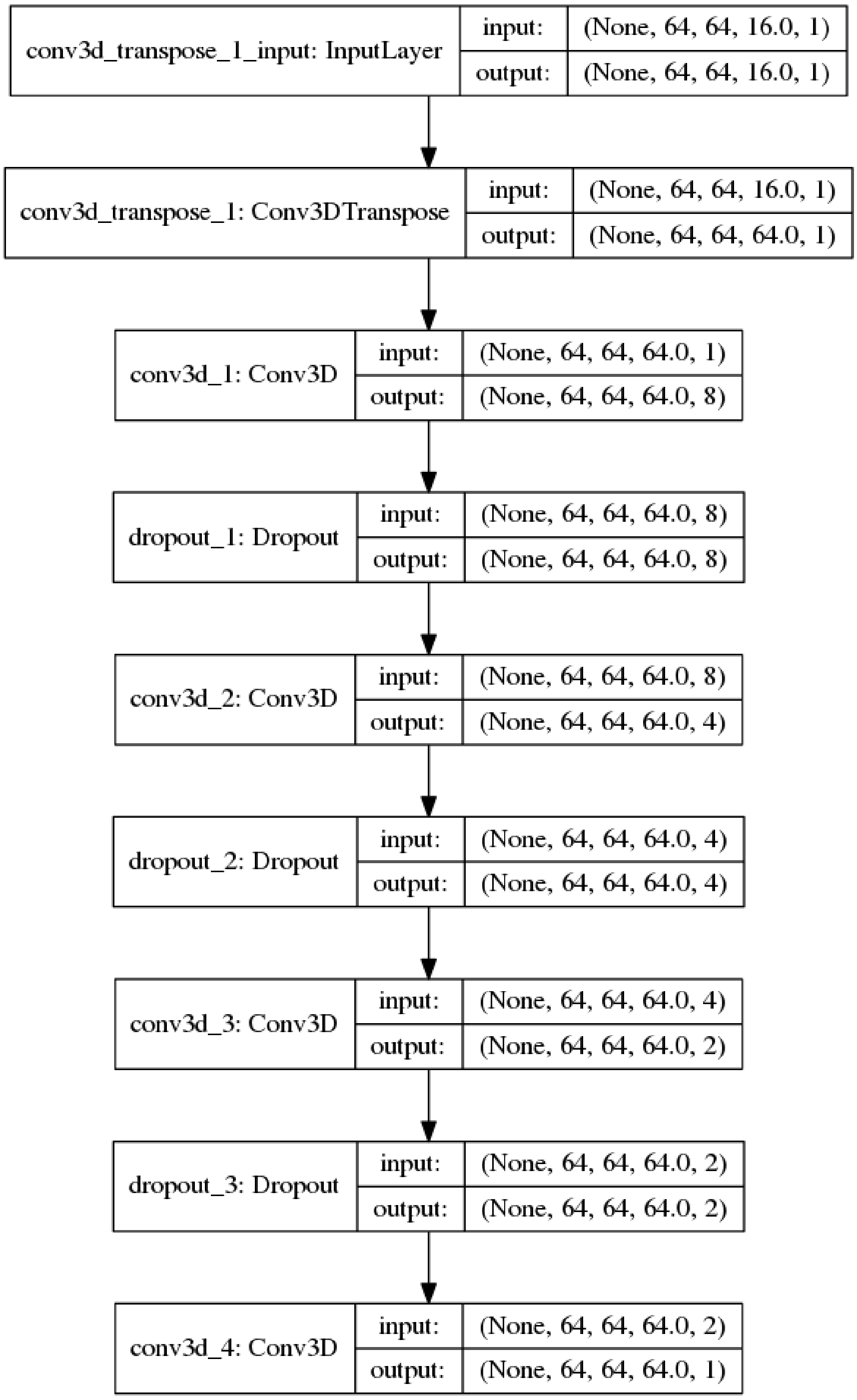

conv3d_transpose_1_input: InputLayer
input: (None, 64, 64, 16.0, 1)
output: (None, 64, 64, 16.0, 1)
conv3d_transpose_1: Conv3DTranspose
input: (None, 64, 64, 16.0, 1)
output: (None, 64, 64, 64.0, 1)
conv3d_1: Conv3D
input: (None, 64, 64, 64.0, 1)
output: (None, 64, 64, 64.0, 8)
dropout_1: Dropout
input: (None, 64, 64, 64.0, 8)
output: (None, 64, 64, 64.0, 8)
conv3d_2: Conv3D
input: (None, 64, 64, 64.0, 8)
output: (None, 64, 64, 64.0, 4)
dropout_2: Dropout
input: (None, 64, 64, 64.0, 4)
output: (None, 64, 64, 64.0, 4)
conv3d_3: Conv3D
input: (None, 64, 64, 64.0, 4)
output: (None, 64, 64, 64.0, 2)
dropout_3: Dropout
input: (None, 64, 64, 64.0, 2)
output: (None, 64, 64, 64.0, 2)
conv3d_4: Conv3D
input: (None, 64, 64, 64.0, 2)
output: (None, 64, 64, 64.0, 1)

Figure 39. 3D SR-DCNN.

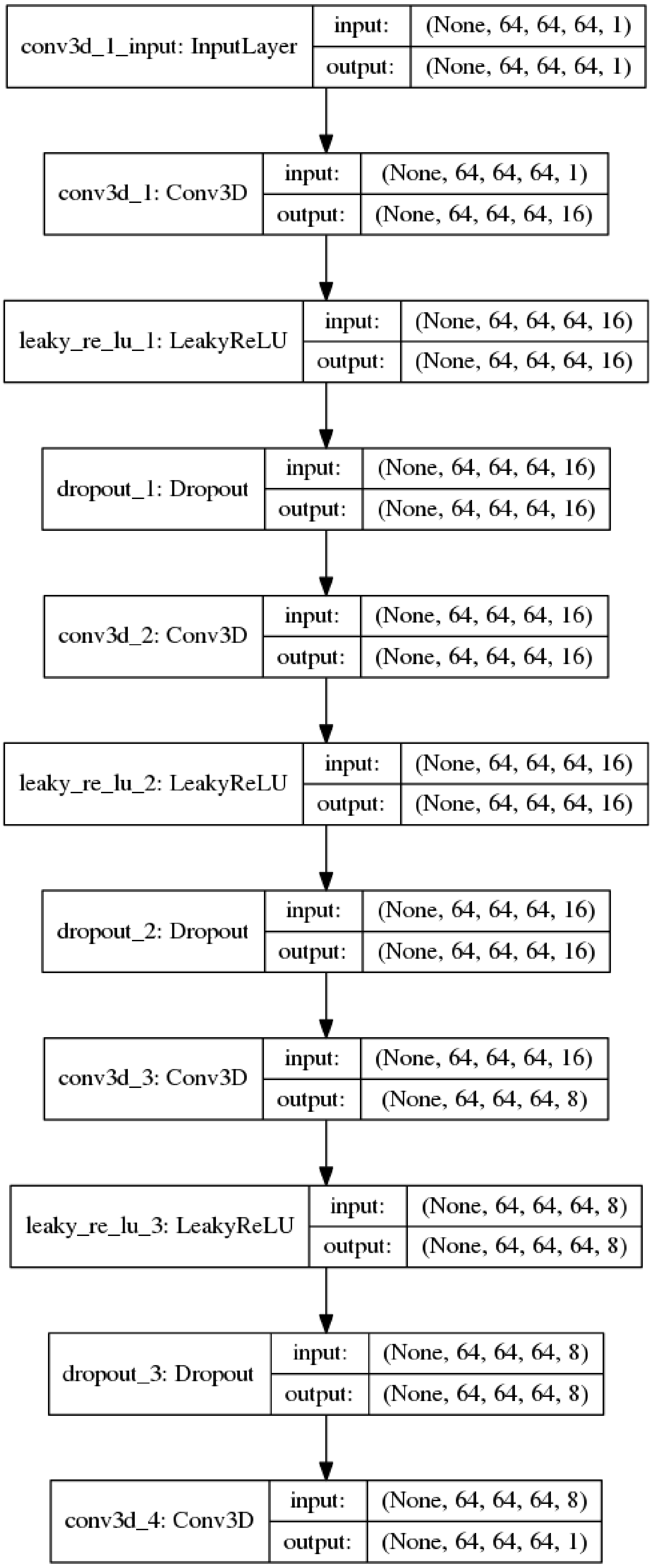

conv3d_1_input: InputLayer | input: (None, 64, 64, 64, 1) | output: (None, 64, 64, 64, 1)
conv3d_1: Conv3D | input: (None, 64, 64, 64, 1) | output: (None, 64, 64, 64, 16)
leaky_re_lu_1: LeakyReLU | input: (None, 64, 64, 64, 16) | output: (None, 64, 64, 64, 16)
dropout_1: Dropout | input: (None, 64, 64, 64, 16) | output: (None, 64, 64, 64, 16)
conv3d_2: Conv3D | input: (None, 64, 64, 64, 16) | output: (None, 64, 64, 64, 16)
leaky_re_lu_2: LeakyReLU | input: (None, 64, 64, 64, 16) | output: (None, 64, 64, 64, 16)
dropout_2: Dropout | input: (None, 64, 64, 64, 16) | output: (None, 64, 64, 64, 16)
conv3d_3: Conv3D | input: (None, 64, 64, 64, 16) | output: (None, 64, 64, 64, 8)
leaky_re_lu_3: LeakyReLU | input: (None, 64, 64, 64, 8) | output: (None, 64, 64, 64, 8)
dropout_3: Dropout | input: (None, 64, 64, 64, 8) | output: (None, 64, 64, 64, 8)
conv3d_4: Conv3D | input: (None, 64, 64, 64, 8) | output: (None, 64, 64, 64, 1)

Figure 40. 3D SRCNN.

9.5 Appendix E. Questionnaire Form.

DICOM ENHACEMENT EVALUATION

Name:

____

Position: _______________________ Date: __ - __ - ____

For the given set of 5 patients, each one with two MRI volumes (volume 1 and volume 2), mark the one that, at first sight, present better resolution. Then, for the given anatomical structures determine how good can be differentiated. Being 1=The structure could not be appreciated and 5= The structure was clearly defined.

The structures to be evaluated are: Anterior Cruciate Ligament (ACL), Posterior Cruciate Ligament (PCL), Patellar Cartilage (PC), Tibial Cartilage (TC), Femoral Cartilage (FC) and Meniscus.

Patient A (No ACL)					
	Volume 1			**Volume 2**	
Best Resolution					
	1	**2**	**3**	**4**	**5**
PCL Volume 1					
PCL Volume 2					
PC Volume 1					
PC Volume 2					
TC and FC Volume 1					
TC and FC Volume 2					
Meniscus Volume 1					
MeniscusVolume 2					

Patient B					
	Volume 1			**Volume 2**	
Best Resolution					
	1	**2**	**3**	**4**	**5**
ACL and PCL Volume 1					
ACL and PCL Volume 2					
PC Volume 1					
PC Volume 2					
TC and FC Volume 1					
TC and FC Volume 2					
Meniscus Volume 1					
MeniscusVolume 2					

Patient C					
	Volume 1			**Volume 2**	
Best Resolution					
	1	**2**	**3**	**4**	**5**
ACL and PCL Volume 1					
ACL and PCL Volume 2					
PC Volume 1					
PC Volume 2					
TC and FC Volume 1					
TC and FC Volume 2					
Meniscus Volume 1					
MeniscusVolume 2					

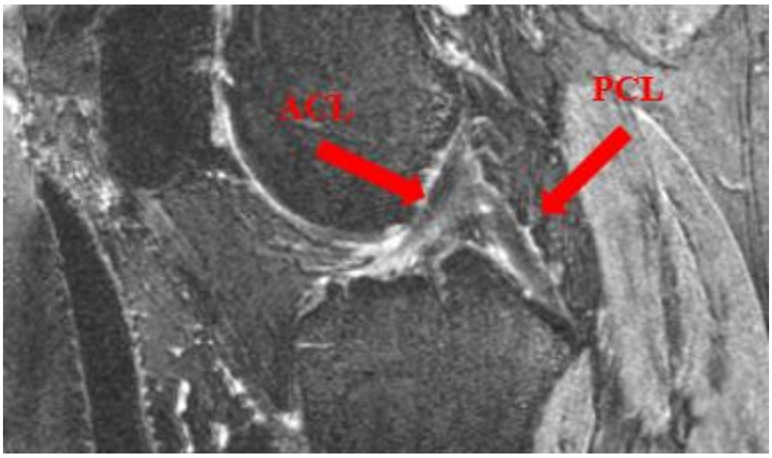
Patient D					
	Volume 1			**Volume 2**	
Best Resolution					
	1	**2**	**3**	**4**	**5**
ACL and PCL Volume 1					
ACL and PCL Volume 2					
PC Volume 1					
PC Volume 2					
TC and FC Volume 1					
TC and FC Volume 2					
Meniscus Volume 1					
MeniscusVolume 2					

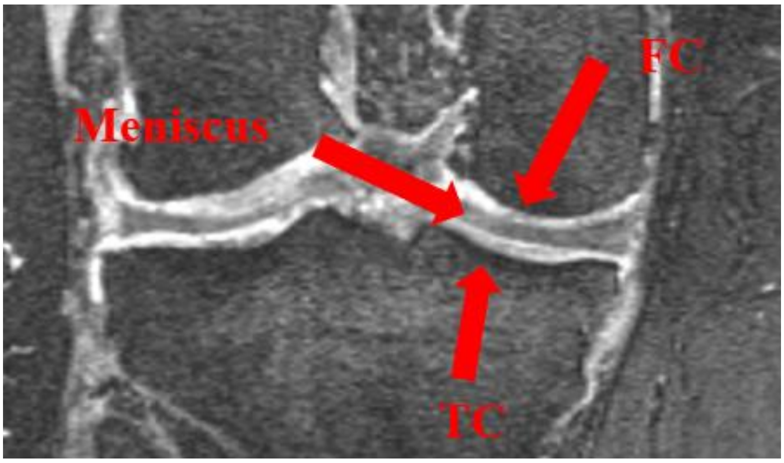
Patient E					
	Volume 1			**Volume 2**	
Best Resolution					
	1	**2**	**3**	**4**	**5**
ACL and PCL Volume 1					
ACL and PCL Volume 2					
PC Volume 1					
PC Volume 2					
TC and FC Volume 1					
TC and FC Volume 2					
Meniscus Volume 1					
MeniscusVolume 2					

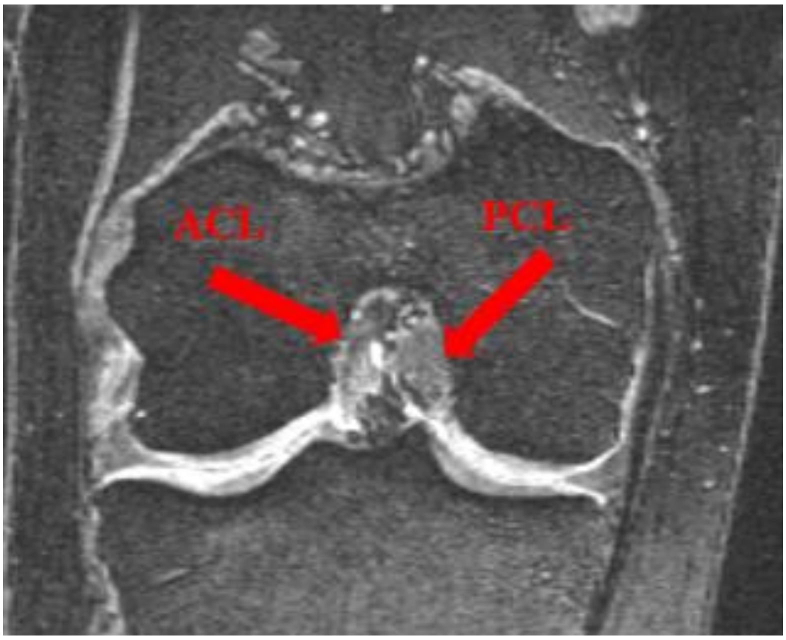
Additional comments:

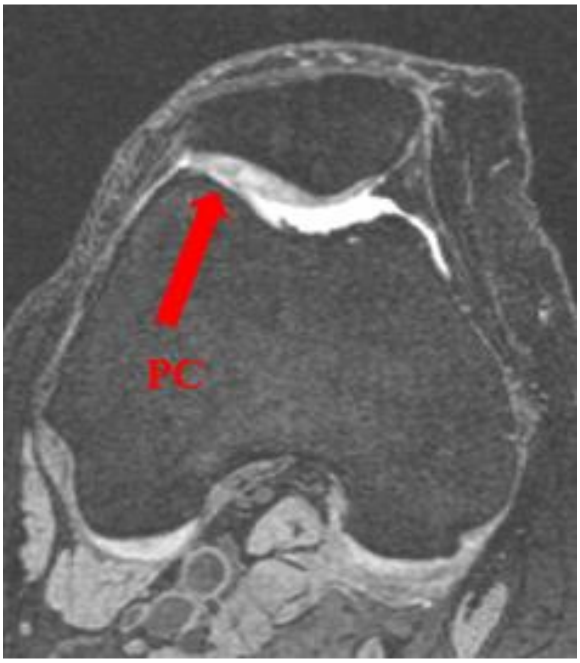
9.6 Appendix F. Guide to Localize Knee Anatomical Structures.

This guide is provided together with the Questionnaire Form.








PC

10 BIBLIOGRAPHY

[1] S. C. a. P. M. K. a. K. M. G. Park, “Super-resolution image reconstruction: a technical overview,” *IEEE signal processing magazine,* vol. 3, no. 3, pp. 21-36, 2003.

[2] S. Wei, “Improving resolution of medical images with deep dense convolutional neural network,” *Concurrency and Computation: Practice and Experience,* p. e5084, 2018.

[3] K. a. O. J. a. I. N. a. O. S. a. O. K. a. S. T. a. S. N. a. I. T. Umehara, “Super-resolution convolutional neural network for the improvement of the image quality of magnified images in chest radiographs,” in *Medical Imaging 2017: Image Processing,* International Society for Optics and Photonics, 2017, p. 101331P.

[4] C. Dong, “Learning a deep convolutional network for image super-resolution,” in *European conference on computer vision,* Springer, 2014, pp. 184-199.

[5] Y. a. L. H. a. D. J. a. F. G. Gao, “A deep convolutional network for medical image super-resolution,” in *2017 Chinese Automation Congress (CAC),* IEEE, 2017, pp. 5310--5315.

[6] H. a. X. J. a. W. Y. a. G. Q. a. I. B. a. X. L. Liu, “Learning deconvolutional deep neural network for high resolution medical image reconstruction,” *Information Sciences,* vol. 468, pp. 142--154, 2018.

[7] K. a. O. J. a. I. T. Umehara, “Super-resolution imaging of mammograms based on the super-resolution convolutional neural network,” *Open J. Med. Imaging,* vol. 180, 2017.

[8] K. Hao, “MIT Technology Review,” 04 04 2019. [Online]. Available: https://www.technologyreview.com/f/613264/the-fda-wants-to-regulate-machine-learning-in-healthcare/. [Accessed 12 05 2019].

[9] “PHG Foundation,” 06 09 2018. [Online]. Available: http://www.phgfoundation.org/briefing/legal-liability-machine-learning-in-healthcare. [Accessed 12 05 2019].

[10 “Observatorio de Resultados del Servicio Madrileño de Salud. Informe de hospitales.

[11 P. Harrell, “DocPlayer. GE Medical System Training in parternship module 12.,” 2016. [Online]. Available: https://docplayer.net/7517269-Ge-medical-systems-training-in-partnership-module-12-spin-echo.html. [Accessed 08 06 2019].

[12 J. S. a. R. K. Isaac, “ "Super resolution techniques for medical image processing.",” *International Conference on Technologies for Sustainable Development (ICTSD).,* 2015 .

[13] T. a. J. Y. Huang, “Image super-resolution: Historical overview and future challenges.,” in *Super-resolution imaging*, CRC Press., 2010, pp. 19-52..

[14] W. Z. X. T. Y. W. W. &. X. J. H. Yang, “Deep learning for single image super-resolution: A brief review.,” *arXiv preprint arXiv:1808.03344,* 2018.

[15] Cmglee, “Wikipedia. Interpolation.,” 15 04 2019. [Online]. Available: https://en.wikipedia.org/wiki/Interpolation#/media/File:Comparison_of_1D_and_2D_ nterpolation.svg.

[16] Y. Wun, “Self-Attention Convolutional Neural Network for Improved MR Image Reconstruction,” *Information Sciences,* vol. 490, pp. 317-328, 2019.

[17] H. Liu, “Learning deconvolutional deep neural network for high resolution medical image reconstruction,” *Information Sciences*, vol. 468, pp. 142-154, 2018.

[18] S. Farsiu, “Fast and Robust Multiframe Super Resolution,” *IEEE Transactions on Image Processing,* vol. 13, no. 10, pp. 1327-1344, 2004.

[19] N. J. Nilsson, The quest for artificial intelligence, Cambridge University Press, 2009.

[20] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress,” in *Computer Games I,* Springer, 1988, pp. 366-400.

[21] R. E. Schapire, “The boosting approach to machine learning: An overview,” in *Nonlinear estimation and classification*, Springer, 2003, pp. 149-171.

[22] A. Rosebrock, Deep Learning for Computer Vision with Python: ImageNet Bundle, PyImageSearch, 2017.

[23] W. S. a. P. W. McCulloch, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics,* vol. 5, no. 4, pp. 115-133, 1943.

[24] S. a. V. V. S. a. T. S. a. U. A. Scardapane, “Kafnets: kernel-based non-parametric activation functions for neural networks,” *arXiv preprint arXiv:1707.04035,* p. 2017.

[25] “A Medium Corporation US,” [Online]. Available: https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092 . [Accessed 27 12 2018].

[26] D. A. F. M. J. S. a. H. Z. Keim, “Challenges in visual data analysis,” in *Challenges ir visual data analysis,* IEEE, 2006, pp. 9-16.

[27] D. H. a. W. T. N. Hubel, “Receptive fields of single neurones in the cat's striate cortex,” *The Journal of physiology},* vol. 148, no. 3, pp. 574-591, 1959.

[28] M. A. a. R. A. E. Fischler, “The representation and matching of pictorial structures.,” in *Transactions on computers*, IEEE, 1973, pp. 67-92.

[29] R. A. R. C. a. T. O. B. Brooks, “The ACRONYM model-based vision system,” in *Proceedings of the 6th international joint conference on Artificial intelligence*, Morgı Kaufmann Publishers Inc., 1979, pp. 105-113.

[30] D. a. A. V. Man, Vision: A computational investigation into the human representatioı and processing of visual information., 1982.

[31] D. G. Lowe, “Object recognition from local scale-invariant features,” *iccv,* vol. 99, nc 2, pp. 1150-1157, 1999.

[32] A. I. S. a. G. E. Krizhevsky, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[33] L. D. W. X. T. X. Sun Y, “Deepid3: Face recognition with very deep neural networks,” *arXiv preprint arXiv:1502.00873.*, 2015.

[34] I. Y. B. a. A. C. Goodfellow, Deep learning, MIT press, 2016.

[35] “Crossroads. The ACM Magazine for Students.,” [Online]. Available: https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/ . [Accessed 05 01 2019].

[36] D. A. B. T. R. M. Springenberg JT, “Striving for simplicity: The all convolutional net,” arXiv preprint arXiv:1412.6806, 2014.

[37] “Slide Share. Design your CNN: historical inspirations.,” [Online]. Available: https://www.slideshare.net/ssuser025470/design-your-cnn-historical-inspirations. [Accessed 08 06 2019].

[38] “A Medium Corporation US,” [Online]. Available: : https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76. [Accessed 06 01 2019].

[39] S. a. C. S. Ioffe, “Batch normalization: Accelerating deep network training by reducir internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.

[40] B. H.-P. C. N. P. D. W. M. A. H. D. D. A. a. M. M. G. Sahiner, “Classification of ma and normal breast tissue: a convolution neural network classifier with spatial domain and texture images,” *IEEE transactions on Medical Imaging* , vol. 15, no. 5, pp. 598-610, 1996.

[41] R. M. J. S. K. a. P. K. Rouhi, “Benign and malignant breast tumors classification base on region growing and CNN segmentation,” *Expert Systems with Applications,* vol. 4 no. 3, pp. 990-1002, 2015.

[42] D. P. a. J. B. Kingma, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[43] F. N. N. a. S.-A. A. Milletari, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 Fourth International Conference o 3D Vision (3DV),* 2016.

[44] G. a. W. Q. a. Q. L. a. H. X. Lin, “Image super-resolution using a dilated convolution neural network,” *Neurocomputing,* vol. 275, pp. 1219--1230, 2018.

[45] “Wikipedia. Peak signal-to-noise ratio,” [Online]. Available: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio. [Accessed 04 06 2019].

[46] Z. A. C. B. H. R. S. a. E. P. S. Wang, “Image quality assessment: from error visibility to structural similarity.,” *IEEE transactions on image processing,* vol. 13, no. 4, pp. 600-612, 2004.

[47] H. a. X. J. a. W. Y. a. G. Q. a. I. B. a. X. L. Liu, “Learning deconvolutional deep neural network for high resolution medical image reconstruction,” *Information Sciences,* vol. 468, pp. 142--154, 2018.

[48] Alemily, “Wikipedia,” 6 July 2006. [Online]. Available: https://en.wikipedia.org/wiki/1951_USAF_resolution_test_chart#/media/File:1951usa _test_target.jpg. [Accessed 07 05 2019].

[49] “MRI Procedure Manual for Examinations of,” SYNARC, [Online]. Available: http://oai.epi-ucsf.org/datarelease/operationsmanuals/mri_manualrev.pdf. [Accessed 25 04 2019].

[50] Eduardas, “Kaggle,” [Online]. Available: https://www.kaggle.com/eduardas/predicting-no-shows-with-a-neural-net-in-keras. [Accessed 16 4 2019].

[51] S. a. W. W. a. J. G. a. A. A. a. Y. X. Wei, “Improving resolution of medical images with deep dense convolutional neural network,” *Concurrency and Computation: Practice and Experience,* p. e5084.