Pontifícia Universidade Católica do Rio de Janeiro

Marcos Rogozinski

Deep learning applied to locating ferromagnetic foreign bodies in humans

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-Graduação em Metrologia of PUC-Rio as partial fulfillment of the requirements for the degree of Mestre em Metrologia.

Advisers: Prof. Carlos Roberto Hall Barbosa
Prof. Elisabeth Costa Monteiro

Rio de Janeiro
December 2021

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Marcos Rogozinski

Deep learning applied to locating ferromagnetic foreign bodies in humans

Dissertation presented to the Programa de Pós-Graduação em Metrologia of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Metrologia. Approved by the Examination Committee:

Prof. Carlos Roberto Hall Barbosa
Advisor
Programa de Pós-Graduação em Metrologia – PUC-Rio

Prof. Elisabeth Costa Monteiro
Co-advisor
Programa de Pós-Graduação em Metrologia – PUC-Rio

Prof. Karla Tereza Figueiredo Leite
UERJ

Prof. José Franco Machado do Amaral
UERJ

Prof. Jorge Luís Machado do Amaral
UERJ

Rio de Janeiro, December 3rd, 2021

Marcos Rogozinski

Graduated Suma Cum Laude on Music Production and Sound Engineering from Berklee College of Music, in 1997, and graduated in "Sistemas de Informação" (Information Systems) from Universidade Estácio de Sá, in 2015.

Bibliographic data

Rogozinski, Marcos

Deep learning applied to locating ferromagnetic foreign bodies in humans / Marcos Rogozinski ; advisers: Carlos Roberto Hall Barbosa, Elisabeth Costa Monteiro. – 2021.
95 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Centro Técnico Científico, Programa de Pós-Graduação em Metrologia, 2021.
Inclui bibliografia

1. Metrologia – Teses. 2. Metrologia para Qualidade e Inovação – Teses. 3. Corpos estranhos ferromagnéticos. 4. Aprendizado profundo. 5. Magnetômetro GMR. 6. Redes neurais convolucionais. 7. Rastreamento de dispositivo. I. Barbosa, Carlos Roberto Hall. II. Monteiro, Elisabeth Costa. III. Pontifícia Universidade Católica do Rio de Janeiro. Centro Técnico Científico. Programa de Pós-Graduação em Metrologia. IV. Título.

CDD: 389.1

For my father, who, despite not being able to attend school, always prioritized learning.
For my mother, who, despite her academic excellence, always prioritized the practice.

Acknowledgements

To my adviser Carlos Roberto Hall Barbosa, for accepting me into the metrology program and for the patience and guidance throughout my learning process.

To my adviser Elisabeth Costa Monteiro for all her support, to professor Raul Queiroz Feitosa for his teachings in Computer Vision, and professors Rodrigo Flora Calili and Maria Fatima Ludovico de Almeida for their teachings and motivation.

To PUC-Rio for the support and opportunity to develop this research.

To my family, friends, and colleagues.

To Veruska, Danilo and Nina.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001.

Abstract

Rogozinski, Marcos; Barbosa, Carlos Roberto Hall Barbosa (Advisor); Costa Monteiro, Elisabeth (Co-advisor). **Deep learning applied to locating ferromagnetic foreign bodies in humans.** Rio de Janeiro, 2021. 95p. Dissertação de Mestrado – Programa de Pós-Graduação em Metrologia, Pontifícia Universidade Católica do Rio de Janeiro.

Ferromagnetic foreign bodies accidentally inserted in patients usually need to be surgically removed. The methods conventionally employed for locating foreign bodies are often ineffective due to the low accuracy in determining the position of the object and pose risks arising from the exposure of medical staff and patients to ionizing radiation during long-term procedures. New methods using SQUID sensors successfully located foreign bodies in an innocuous and non-invasive way, but they have the drawback of presenting high cost and low portability. This work is part of new research that seeks to bring greater portability and low cost in locating foreign bodies in the human body using GMI and GMR sensors. The main objective of this work is to evaluate and apply the use of Deep Learning in the development of a portable and manual device based on a GMR sensor, including position tracking and orientation of this device from images of known patterns obtained by a camera integrated to the device and the solution of the inverse magnetic problem from the obtained magnetic mapping. The techniques presented are capable of tracking the device with good accuracy and detecting the localization of the foreign body with similar or better results than those obtained in previous works, depending on the parameter. The results obtained are promising as a basis for future developments.

Keywords

Metrology; Ferromagnetic Foreign Bodies; Deep Learning; GMR Magnetometer; Convolutional Neural Networks; Device Tracking

Resumo

Rogozinski, Marcos; Barbosa, Carlos Roberto Hall Barbosa; Costa Monteiro, Elisabeth. **Aprendizado Profundo aplicado na localização de corpos estranhos ferromagnéticos em humanos.** Rio de Janeiro, 2021. 95p. Dissertação de Mestrado – Programa de Pós-Graduação em Metrologia, Pontifícia Universidade Católica do Rio de Janeiro.

Corpos estranhos ferromagnéticos inseridos acidentalmente em pacientes geralmente precisam de remoção cirúrgica. Os métodos convencionalmente empregados para localizar corpos estranhos são frequentemente ineficazes devido à baixa precisão na determinação da posição do objeto e representam riscos decorrentes da exposição da equipe médica e dos pacientes à radiação ionizante durante procedimentos de longa duração. Novos métodos utilizando sensores SQUID têm obtido sucesso na localização de corpos estranhos de forma inócua e não invasiva, mas têm a desvantagem de apresentar alto custo e baixa portabilidade. Este trabalho faz parte de pesquisas que buscam trazer maior portabilidade e baixo custo na localização de corpos estranhos no corpo humano utilizando sensores GMI e GMR. O objetivo principal deste trabalho é avaliar e aplicar o uso de Aprendizado Profundo para a localização de corpos estranhos ferromagnéticos no corpo humano utilizando um dispositivo portátil e manual baseado em magnetômetro GMR, incluindo o rastreamento da posição e orientação deste dispositivo a partir de imagens de padrões conhecidos obtidas por uma câmera integrada ao dispositivo e a solução do problema inverso magnético a partir do mapeamento magnético obtido. As técnicas apresentadas se mostraram capazes de rastrear o dispositivo com boa precisão e detectar a localização do corpo estranho com resultados semelhantes ou melhores do que os obtidos em trabalhos anteriores, dependendo do parâmetro. Os resultados obtidos são promissores como base para desenvolvimentos futuros.

Palavras-chave

Metrologia; corpos estranhos ferromagnéticos; aprendizado profundo; magnetômetro GMR; redes neurais convolucionais; rastreamento de dispositivo

Table of contents

List of Figures

List of Tables

List of Symbols

ϕ – Rotation, Bank or Roll

θ – Inclination, Elevation or Pitch

Ψ – Heading or Yaw

h – Depth

p – Geometric center of the needle

Δ – Euclidean distance between the center of the extreme values of the magnetic field and the geometric center of the needle

List of Abbreviations

BCE – Binary Cross Entropy

CNN - Convolutional Neural Network

COVID-19 – Coronavirus Dise

CPU – Central Processing Unit

DDR3 – Double Data Rate 3

FC Layers - Fully Connected Layers

FCN - Fully Convolutional Network

GMI - Giant Magnetoimpedance

GMR - Giant Magnetoresistance

GPU – Graphics processing unit

ICU – Intensive Care Units

MAE – Mean Absolute Error

MS COCO – Microsoft Common Objects in Context

MSE - Mean Squared Error

PUC-Rio – Pontifical Catholic University of Rio de Janeiro

RMSE – Root Mean Squared Error

RoI Align – Region of Interest Alignment

RPN – Region Proposal Network

SQUID - Superconducting Quantum Interference Device

VGG - Visual Geometry Group (Department of Engineering Science, University of Oxford)

1
Introduction

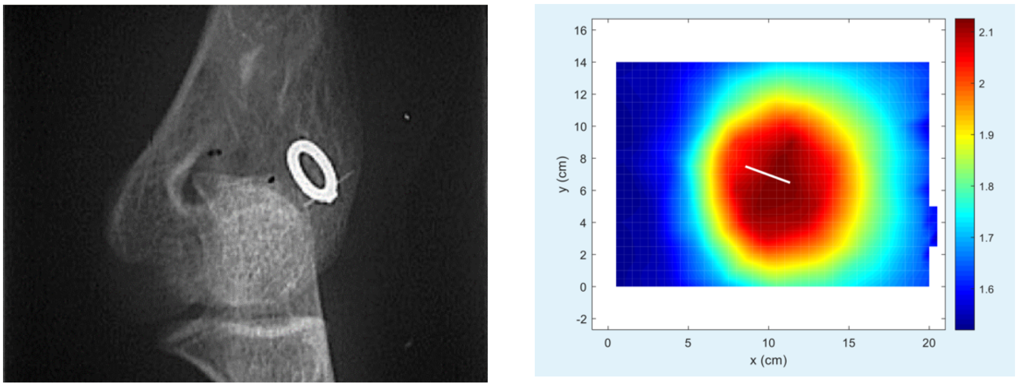
1.1.
Contextualization

There is a high incidence of cases of metallic objects [1], ferromagnetic or not, inserted in patients and locating the position of these so-called foreign bodies inside the human body is essential for the effectiveness of their surgical removal [1]–[10]. Radiological methods such as radiography, computed tomography and radioscopy procedures are the conventional procedures available for locating these objects [1], [4]–[10]. However, these conventionally employed methods for locating foreign bodies are often ineffective due to poor accuracy in determining the position and depth of the object to the skin [1], [4], [10]. Auxiliary radiological strategies sometimes used during the surgical procedure include the introduction of reference needles at various positions and the evaluation of the location of these references to the object to be removed employing X-ray films or radioscopy with takes at diverse incidence angles, which presents risks due to the the exposure of medical staff and patients to ionizing radiation during long-term procedures [1], [4], [10]. In [4], a method for locating ferromagnetic foreign bodies was developed and applied to guide surgical procedures. The developed method enabled the successful removal of metallic needles and fragments through procedures lasting about 10 minutes, with previous attempts taking up to 6 hours without success [4]. Magnetic field measurements in these studies were performed using a SQUID (Superconducting Quantum Interference Device) sensor, currently the most sensitive magnetometer available [2]–[4]. However, the operation of SQUID sensors requires cryogenic temperatures, which introduces a high cost, making it difficult to disseminate the technique in the clinical environment [11], [12].

Thus, many studies have been carried out to develop and improve non-invasive methods to locate foreign bodies in the human body based on the measurement of the magnetic field and to evaluate the possibility of using sensors with lower acquisition and operation cost [2]–[4], [6], [11]–[19]. Researchers at the

Pontifical Catholic University of Rio de Janeiro (PUC-Rio) have focused their efforts on studies with sensors based on the giant magnetoimpedance (GMI) phenomenon for biomedical applications [2], [6], [11]–[15], [17]–[26], among which the location of metallic foreign bodies.

Figure 1a shows a radiograph taken before a surgical procedure for the removal of a foreign body based on the location provided by magnetic mapping [4]. The location of the needle is indicated on the image with the aid of a radiopaque marker positioned on the skin, based on the markings obtained through magnetic mapping using a SQUID device. Initially, in the radiographic image, the needle fragment was interpreted as an artifact or a bone trabecula, but magnetic mapping, initially performed to locate another foreign body that had been clearly identified radiographically, made it possible to identify the presence and determine the position of the tiny magnetic dipole [4]. Figure 1b, on the other hand, shows a magnetic map generated by a 3.3 cm needle whose projection is indicated in the image obtained by a GMI sensor [6]. In addition to GMI sensors, GMR (Giant Magnetic Resistance) sensors are also an interesting option for locating foreign bodies, especially concerning their acquisition and operation cost [14], [15].



(a) (b)

Figure 1 – (a) Radiographic image indicating the location of a needle fragment, obtained through magnetic mapping using SQUID [4]; and (b) magnetic map of a 3.3 cm needle, whose position is projected onto the image, in *in vitro* measurements using a GMI sensor [6].

Systems employing GMI or GMR sensors, although not as sensitive, have the advantage of portability and lower operating and manufacturing complexity and cost compared to SQUID sensors [12], [15]. However, all systems developed so far have been configured using a fixed structure to carry out the mapping. Figure 2 illustrates the setup used for *in vivo* measurements using the SQUID system [4], in which the mapping is performed with the patient on a mobile bed and the transduction system positioned on a fixed structure.

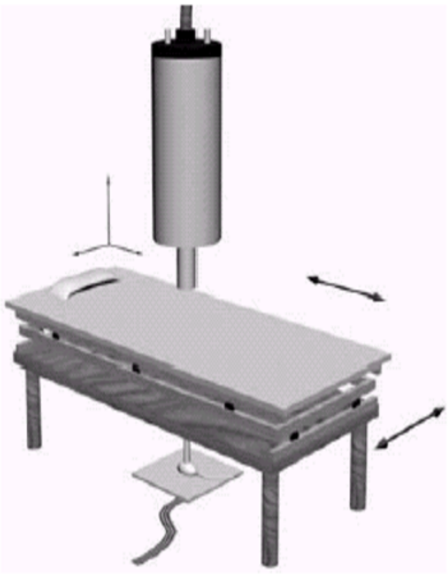


Figure 2 – Configuration of the measurement system using a SQUID device, positioned on a fixed structure, for magnetic mapping for the location of foreign bodies in the human body, developed by researchers from PUC-Rio (adapted from [4]).

More recently, researchers at PUC-Rio have been implementing efforts to develop a device that is not only portable but also manual, using lower-cost systems based on GMR sensors. The device is based on the Raspberry Pi platform and has a GMR sensor, a distance sensor, and a monocular camera in its initial configuration. The fact that the device is manual and portable, unlike currently existing systems based on a fixed positioning structure, means that its position and orientation in relation to the patient is not known *a priori*. It is, therefore, of fundamental importance that this device can track its position and orientation in relation to some fixed reference point on the patient so that, together with the magnetic field measurements performed by the GMR sensor, it is possible to solve the inverse problem and locate the foreign body in the patient.

1.2 Motivation

With the increase in computational capacity and the reduction in the mass and dimensions of electronic devices and sensors, different portable products appear on the market in the most diverse sectors. In the medical field, even with greater strictness in the approval of equipment due to strong regulations, portable equipment is already being used commercially for some tasks such as blood analysis, ultrasound, echocardiograms, blood glucose measurement, among others.

The company General Electric, for example, launched in 2010 a product called Vscan, a portable ultrasound system for performing quick diagnostics that can be used in any location and is already considered as a possible replacement for the traditional stethoscope.

Another possible application of portable medical devices is telemedicine or remote care. In this case, the patient or an assistant carrying a portable device can collect the data to be analyzed remotely by a specialist. The company Basil Leaf Technologies, 2017 winner of the Qualcomm Tricorder XPRIZE award [27], has been improving its DxtER™ product that seeks to diagnose patient health problems in a simple way, with a focus on telemedicine, emergency care, and primary healthcare. The device can collect the patient's vital signs and diagnose health conditions such as diabetes, atrial fibrillation, chronic obstructive pulmonary disease, urinary tract infection, sleep apnea, leukocytosis, whooping cough, stroke, tuberculosis, and pneumonia [28].

However, there is still a significant gap in the medical field of portable and low-cost devices that can help doctors without the need for large support infrastructure, which would facilitate the dissemination of these technologies, promote greater equality of technical conditions of care, especially in developing countries, in addition to enabling emergency care in remote areas or mobile care units.

For a portable device to be successful in locating metallic foreign bodies in patients, in addition to collecting magnetic field data, it needs to be able to track its position and orientation in relation to a fixed point on the patient, with six degrees of freedom: three linear, representing its position relative to the reference point in the X, Y and Z axes, and three angulars, representing the rotation angles relative to the reference point in these three axes (Tait–Bryan angles), called, respectively, bank or roll (ϕ), elevation or pitch (θ) and heading or yaw (Ψ). Figure 3 illustrates the positioning of a camera and its location and orientation parameters in three-dimensional space.

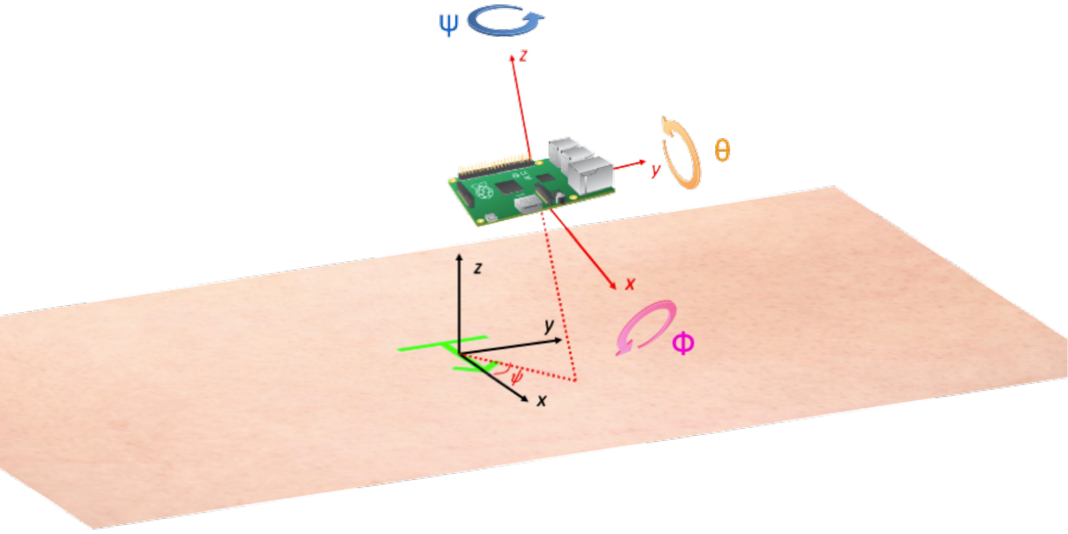


Figure 3 – Indication of the six degrees of freedom of positioning and rotating a camera in relation to an object: the position of the camera in relation to the object in the X, Y and Z axes and the rotation of the camera in each axis, indicated by the values **ϕ** (roll), **θ** (pitch) and **Ψ** (yaw).

The maturation of techniques in image processing and computer vision, especially with the use of Deep Learning, has allowed the advance in the use of low-cost portable cameras for detecting and tracking objects in various applications. Some of these techniques are listed below:

- Object detection: recognizes a predefined object within an image, returning a bounding box that best represents its position in the image;
- Semantic segmentation: classifies each pixel in the image into a predefined class. It extracts the shape of one or more objects of interest from the image;
- Regression using Convolutional Neural Networks: allows mapping images into real values, which in this research would serve as the final camera rotation data.

Much of the research aimed at finding the position and orientation of a camera from images has as its initial problem the detection of reference points that can be mapped in three dimensions [29]–[33]. More recent methods seek to solve the problem using regression from Convolutional Neural Networks to find the position and orientation of the camera directly from the image [34]–[41]. These methods, however, have achieved considerably lower accuracy than previously existing structured methods, such as 3D structure-based approaches [42].

In all the methods mentioned, it is important that the camera has a broad view of the scene to be analyzed so that the adopted methodology can find references in the images that serve as a basis for tracking the device.

Considering, in particular, the application in the clinical environment for mapping and locating foreign bodies in patients, measurements are made close to the patient, generating skin images with slight variation in color and depth, making most of the existing methodologies unfeasible. The controlled environment, however, allows a reference to be drawn or adhered to the patient's skin, to serve as a basis for determining the location and orientation of the camera, and to be the reference for the device's output data, providing accurate position diagnosis of the foreign body. In an ideal situation, the patient would first undergo a radiographic examination to find the approximate location of the foreign body and the reference would be put as close as possible to this location, so the GMR sensor could get better readings by scanning around the foreign body.

Figure 4 illustrates computer vision techniques with Deep Learning with the potential to contribute to the identification of camera position and orientation from a known pattern drawn on the patient's body.

During magnetic mapping for foreign body location, the use of a camera for data acquisition brings essential benefits, such as reducing cost, mass, and size of the final device, avoiding any interference in the magnetic sensors that could occur with other types of sensors, in addition to allowing readings very close to the patient. The data generated from the proposed methodology, combined with the magnetic sensor data, could serve as a basis for mapping the spatial distribution of the magnetic flux density generated by the foreign body. From this mapping, it is possible to apply different methodologies to find the geometric center of the foreign body, its depth, inclination, and rotation in relation to the reference position on the patient's skin. The approach could contribute to the reliability of clinical diagnoses and serve as a basis for augmented reality application in the surgical procedure for foreign body removal.

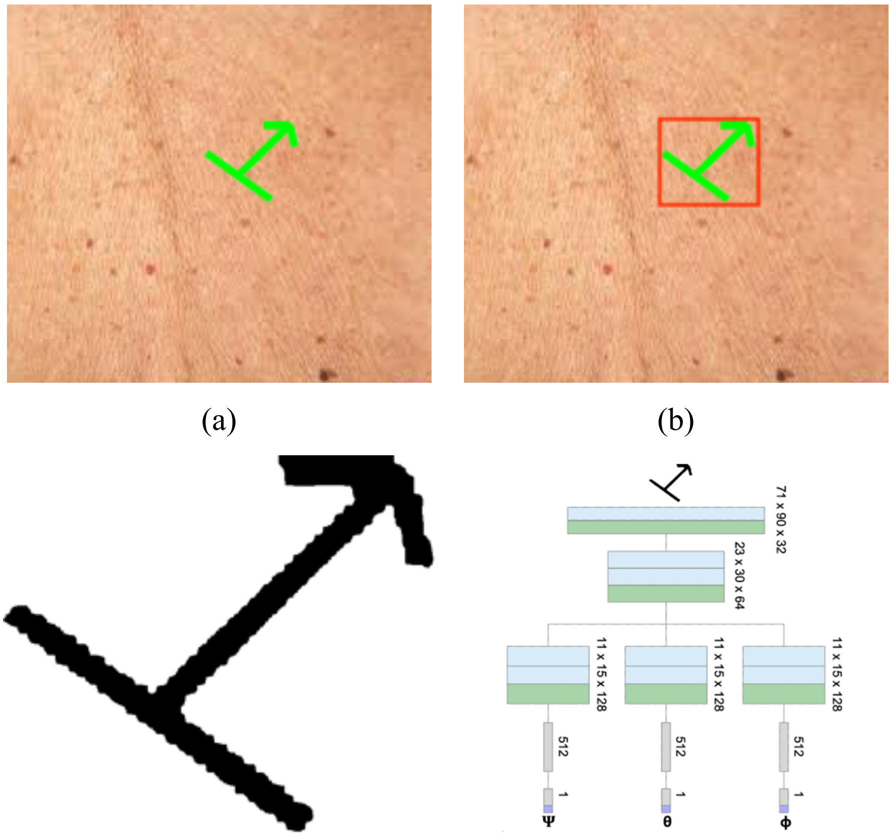


Figure 4 – Illustration of the various methods of computer vision with Deep Learning potentially applicable in identifying the position and orientation of the camera based on a reference on the patient's skin. (a) pattern drawn on the patient's skin; (b) object detection mechanism indicates the bounding box that best represents the location of the object in the image; (c) application of semantic segmentation in the content of the rectangle found in (b) for the extraction of pixels belonging only to the pattern; and (d) Convolutional Neural Network that receives the result of (c) and returns the rotation angles in the three axes. These techniques are explained in more detail in Chapter 2 of this dissertation.

1.3. Objectives

The research’s main objective is to evaluate and apply the use of Deep Learning for the location of ferromagnetic foreign bodies in the human body using a portable and manual device based on a GMR magnetometer (currently under development in another master's research), including position tracking and orientation of this device from images of known patterns obtained by a camera integrated to the device and the solution of the inverse magnetic problem from the obtained magnetic mapping.

To achieve this general objective, the following specific objectives are defined:

- Identify the Computer Vision Deep Learning methods that can be used for tracking the position and orientation of the portable device, selecting the most suitable for the problem at hand;
- Identify the Computer Vision Deep Learning methods that can be used to solve the magnetic inverse problem, selecting the most suitable for the problem in question;
- Develop a method using Deep Learning for tracking the position and orientation of a portable device, considering its use in locating foreign bodies in patients, from images of known patterns obtained by a camera integrated into the device;
- Develop a method using Deep Learning to solve the inverse magnetic problem in locating foreign bodies in patients, from the magnetic maps measured by the device; and
- Demonstrate the applicability of this method by performing synthetic tests based on computer simulations.

1.4. Dissertation structure

This dissertation is structured in six chapters, ranging from the conceptualization of the Deep Learning techniques used to the proposed methodology, its application in a simulated situation, and the analysis of the results obtained.

Chapter 1 refers to the introduction of the dissertation, which seeks to establish the context of the problem to be worked on in this research, the studies related to the theme, the existing gaps that are sought to be solved, its main challenges, and the main steps that will be taken to obtain the expected results.

Chapter 2 describes in detail the existing Deep Learning in Computer Vision methodologies that seek to solve problems of object detection in images, semantic segmentation in images and regression in Convolutional Neural Networks. These techniques are used in the methodology proposed in this research.

In chapter 3, the methods proposed for three-dimensional tracking of the portable device and solution of the magnetic inverse problem are described in detail, the models used and their training process, the types of input and output data, and the expected results.

In chapter 4, the construction of the dataset used for training the models, the test environment, the metrics used in the measurements, the procedures performed, and the equipment and configurations used in carrying out the experiments are described.

Chapter 5 presents the experimental results obtained and analyzes their applicability in clinical situations.

Finally, Chapter 6 presents the final considerations and motivation for future work. It is considered mainly a solution that uses Deep Learning in all phases of the process of obtaining the location of foreign objects in the human body and the proposition of other applications that can make use of the approach proposed in this research.

2 Deep Learning in Computer Vision

2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) is a type of neural network specialized in treating images and tensors. In a traditional artificial neural network, neurons in one layer are connected to all neurons in the preceding layer, creating a series of fully connected layers (FC Layers). This structure tends to create a large number of parameters to be adjusted by the network during training, which makes it poorly scalable. CNN replace fully connected layers with convolutional layers (two-dimensional digital filters), reducing the number of network parameters and thus facilitating the treatment of more data and the creation of network architectures with more layers.

Figure 5 shows a CNN with three convolutional layers, a fully connected layer, and some operations commonly found in this type of network, such as batch normalization, activation function, pooling, flattening, and dropout. These operations will be explained in more detail in the following topics.

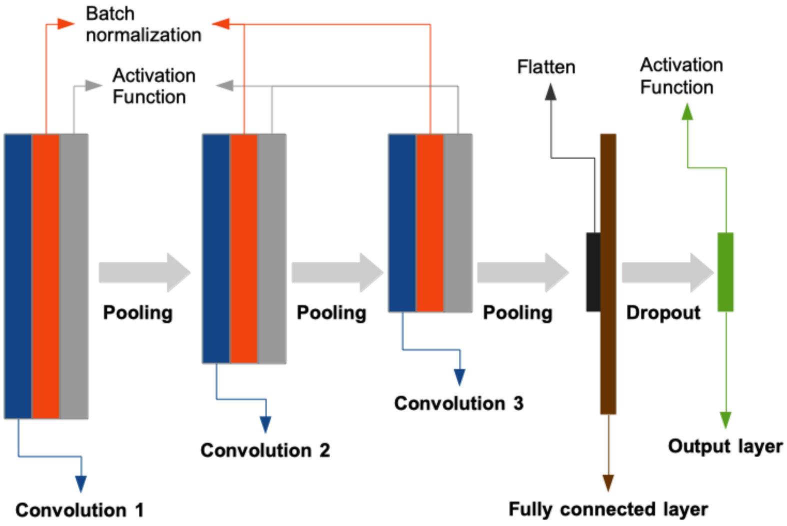


Figure 5 – Example of a Convolutional Neural Network architecture with 3 convolution layers, a fully connected layer, and several operations.

2.1.1 Convolutional layers

A convolutional layer is characterized by applying a convolution (filtering) operation on the input data. This operation is performed using a filter of size $n \times n \times N_{layers}$ that slides over the input data by multiplying the values of each filter position with the corresponding values in the input data and outputting the sum of the multiplied values. The process is repeated for each position of the input data. Figure 6 illustrates a 2D convolution in the first position of a matrix of size 4×4 using a filter of size 3×3 and the repetition of this process in all positions of the input matrix until obtaining the final result. If the input is a tensor, as in the case of an image with color channels, the filter will go through all layers, always generating a matrix as output for each filter, called activation map. The output depth of a convolution layer will be equal to the number of filters applied to the input data, generating their corresponding activation maps. These activation maps usually go through an activation function, adding non-linearity to the process. It is important to note that the values of each filter are the weights to be updated by the network, which drastically reduces the number of parameters that must be learned by the network.

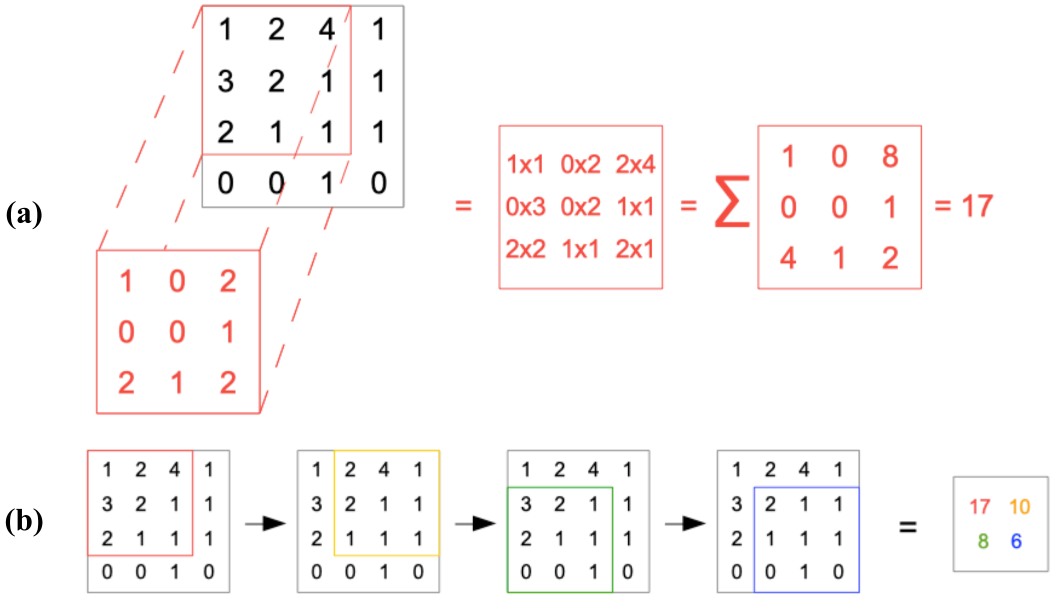


Figure 6 – Convolution process of an input matrix of size $\mathbf{4} \times \mathbf{4}$ by a filter of size $\mathbf{3} \times \mathbf{3}$ showing (a) a convolution in the first position of the matrix input and (b) the filter sliding through the possible positions of the matrix and the final result of the convolutions.

2.1.2 Activation function

Activation functions are mathematical operations, normally applied at the output of the inner layers of a convolutional network, which introduce nonlinear behaviors and seek to provide the network with the ability to approximate complex arbitrary functions [43]. The currently most used functions are: *Sigmoid* (equation 1), *tanh* (equation 2), *ReLU* (equation 3), *Leaky ReLU* (equation 4) and Parametric ReLU (equation 5), where *a* is learned by the network during training.

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

$$f(x) = tanh(x) \tag{2}$$

$$f(x) = max(0, x) \tag{3}$$

$$f(x) = max(0.1 * x, x) \tag{4}$$

$$f(x) = max(0, x) + a \cdot min(0, x) \tag{5}$$

2.1.3 Batch normalization

Batch normalization was proposed by Ioffe and Szegedy [2] as a way to improve the backpropagation pass through the network, allow higher learning rate values and reduce the dependence of a good initialization on the network weights. One of the difficulties in the learning process of neural networks is that, at each batch, the distribution of input data from a layer can vary when the weights from the previous layer are updated, making the network try to learn something in constant change. Batch Normalization normalizes the values at the input of each layer so that each training batch has an average equal to zero and unitary variance, stabilizing the learning process of the network and reducing the number of epochs necessary for convergence. The method is commonly used in convolutional networks and is usually applied before the activation function.

2.1.4 Pooling and unpooling layers

The pooling operation is generally used to reduce the dimensions of activation maps and is typically used between two layers of convolution. The operation consists of grouping the neighboring values of an activation map and using the maximum value in each group (max. pooling) or the average value of the group (average pooling). When using a pooling of 2 × 2, for example, four values will be grouped into a single value, and the dimensions of each activation map are reduced by half, thus reducing the computational complexity of the network.

Unpooling is the inverse process in which an input pixel is transformed into other neighboring pixels, either by repetition (nearest neighbor) or by adding zeros (bed of nails).

Both operations are computationally efficient as they do not involve learning new parameters. Figure 7 illustrates the operations described here.

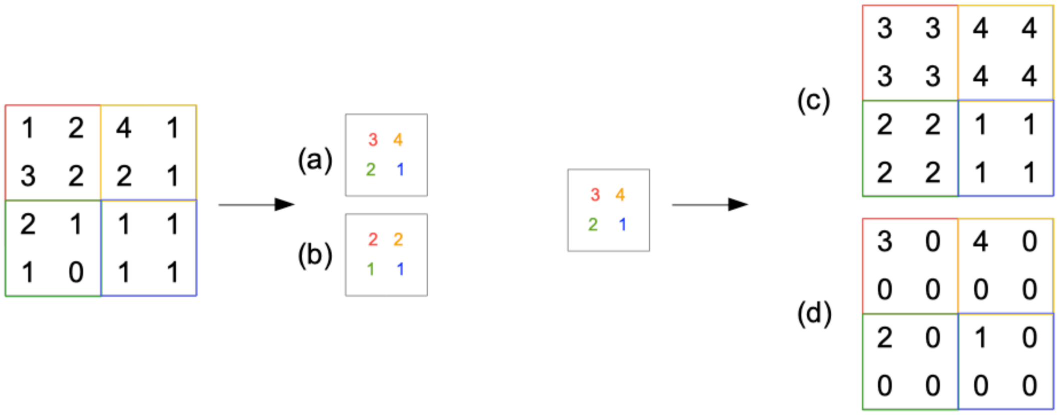


Figure 7 – Operations commonly used by CNNs to change the dimensions of images and activation maps. Dimensional reduction operations using (a) max. pooling and (b) average pooling; and dimensional increase using (c) nearest neighbor and (d) bed of nails. In this example, all operations use a pool size of $\mathbf{2} \times \mathbf{2}$.

2.1.5 Fully Connected Layers and Flatten Layer

In the fully connected layer, each neuron in the layer is connected to all the outputs of the previous layer, as in a traditional artificial neural network of the multi-layer perceptron type [44], [45]. Each link with the previous layer has a weight that must be learned. This weight will be multiplied by the input value and added to an offset value (bias), which is also learned by the network. The large number of connections that are usually established in this type of layer results in a large number of learning parameters, making the network computationally heavy.

In a convolutional network, for the fully connected layer to receive values compatible with its operating structure, it is necessary to apply the flatten operation to transform the output tensor of the previous layer into a vector.

2.1.6 Dropout

The dropout operation is a regularization method that randomly deactivates some neurons and their connections during training, resulting in different minor variations from the original network. During testing, all neurons are used. The operation prevents neurons from settling together at specific values, preventing overfitting [46], which is when the network adjusts very well to the training values but cannot generalize well to other input data.

2.1.7 Output layer

The output layer is usually composed of an activation function chosen according to the purpose of the network. For regression networks, where you want to obtain a single numerical value, a single output with a linear function is used. As for classification networks, the layer will have an output for each possible class. The softmax function is more used, as described by equation (6), which generates a normalized vector containing the probability of each class being the one contained in the image. In this way, it is possible to verify which class has the highest probability of being true among the possible classes of the network.

$$a_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for j } = 1,2,3 \ldots, \mathrm{K} \qquad (6)$$

Another function used in classification networks where the result can contain more than one class as output is the Sigmoid (equation 1), being possible to obtain an individual probability for each class.

2.2 CNN Architectures

Several different architectures have been created in recent years for specific purposes using convolutional neural networks. For the present work, in addition to the basic functionalities already presented for regression and classification networks, two areas of use of CNNs are of fundamental importance: object detection and semantic segmentation, both described below. To embrace those areas, we propose the use of a Mask R-CNN framework [47] in our research, since it covers both object detection and semantic segmentation and was already widely used at the time of this research. The many available code repositories of the Mask R-CNN allowed us for a quick implementation of the framework as a proof of concept. This section provides a brief description of the areas of object detection and semantic segmentation and an analysis of the Mask R-CNN structure.

2.2.1 Object detection

Image classification, as discussed above, is a form of object detection, as the objective is to identify what the image represents. Another area of computer vision called object location is concerned with locating the position of a particular class of objects within the image. In this modality, the network seeks to return the size and coordinates of a rectangle that surrounds the class in question. Object detection is more complete, looking for one or more classes within an image and returning the bounding boxes with their corresponding classes. Figure 8 shows examples of image classification, object location, and object detection.

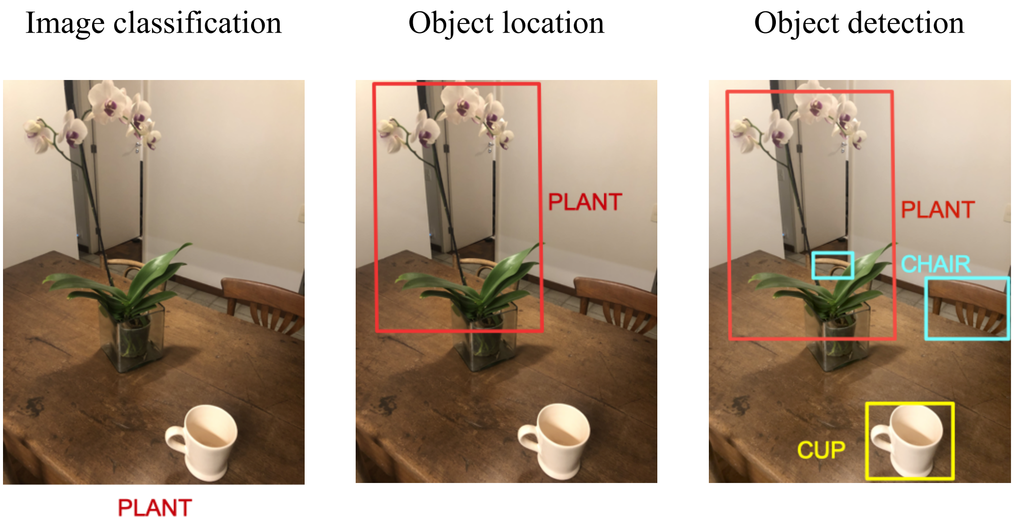


Figure 8 – Examples of results for image classification, object location, and object detection.

2.2.2 Semantic segmentation

Unlike the models presented so far, which are concerned with classifying images or part of the image, semantic segmentation seeks to classify each pixel of the image, generating regions that accurately represent the pixels belonging to a class. To enable this task, the fully connected layers at the end of the network are replaced by convolutional layers, creating a Fully Convolutional Network (FCN). Furthermore, the dimensions of the network output must be the same as the input image, but keeping all layers of the network at the same resolution is quite costly in computational terms. A solution used in many networks is to have architectures with downsampling layers (via pooling, for example) followed by upsampling layers (via unpooling) until the resolution of the input image is recovered.

There are also networks that perform instance segmentation, seeking to recognize the instances of each class within an image. Figure 9 shows examples of object detection, semantic segmentation, and instance segmentation.

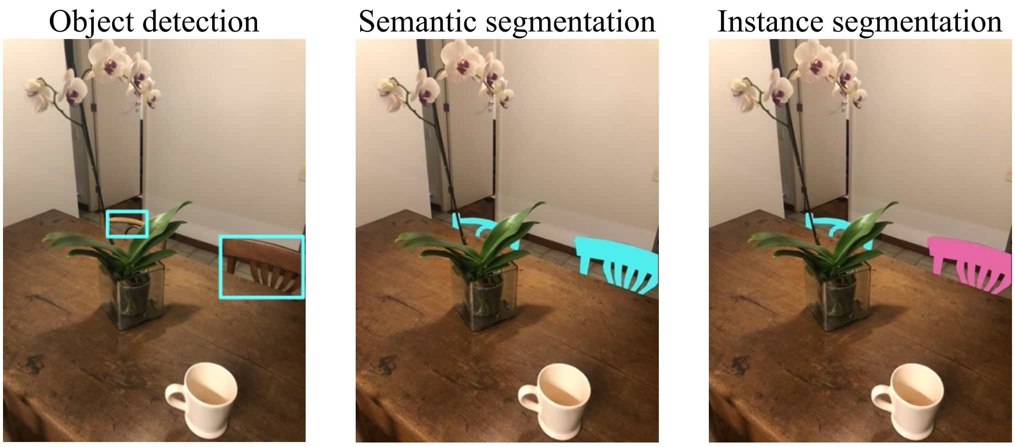


Figure 9 – Examples of results for object detection, semantic segmentation, and instance segmentation.

2.2.3 Mask R-CNN

In addition to classification and regression networks composed of the components already presented in this chapter, this dissertation uses a Mask R-CNN network [47]. This network is an unfolding of an evolutionary series of different object detection architectures initiated by the R-CNN network [48] and improved by the Fast R-CNN [49] and Faster R-CNN [50] networks. For this dissertation, the use of this network becomes interesting for not only detecting objects but also segmenting the objects found; for enabling its use in applications with speeds close to real-time; because it is already a well-established network in several applications with satisfactory results and because code repositories for several platforms already exist, facilitating its incorporation in the simulations of this dissertation.

The Mask R-CNN comprises two main stages: the first generates regions containing objects of interest, and the second makes the object segmentation. It is considered a framework since different algorithms can be used in each of the stages. For a better understanding of the network, its operation is presented divided into the three steps below:

1) Region Proposal Network (RPN): This step is inherited from the Faster R-CNN network [50] and is responsible for detecting regions in the image with a high probability of containing an object of interest. The input image is initially resized to a standard size and then passes through a backbone, which is a traditional convolutional neural network, such as a VGG [51] or ResNET101 [52], to extract feature maps from the image. The framework slides rectangles of pre-defined sizes over each attribute map to estimate if an object exists and what its size and shape are. The regions of interest that are most likely to contain an object are passed to the next step.
2) Region of Interest Alignment (RoI Align): Resizes the regions of interest found in the RPN step to the default size of the masks that will be generated in the next step. The method improves the RoI Pooling used by previous networks, generating an accuracy gain between 10 % and 50 % by making a better approximation of the values of the input pixels.
3) In this third step, each of the proposed regions that were resized in step 2 is applied to the attribute maps of step 1 and goes through a network of fully connected layers that, in one of its branches, improve the position and size parameters of the enclosing rectangle found in step 1 and, in another branch, classifies the object contained in the region in one of the existing classes. This operation is the same that already existed on the Faster R-CNN [50] network, and the novelty in this stage of the Mask R-CNN is the addition of another branch in the network with two convolutional layers, without dimensional reduction, which predicts the mask belonging to the found object. The cost function considers the three parameters of this step: attributes of the bounding box, object class, and object mask, training the branches together and making the network work in a multitasking way, returning all values in

parallel. The mask found at the end of the network is processed to recover its original resolution, as it was resized in the two previous steps.

With this, we have at the output of our network the various objects found, each one with the location and size of its bounding box, its class, and the pixels of the image that belong to the object instance. Figure 10 illustrates the main components of the Mask R-CNN network presented in this topic.

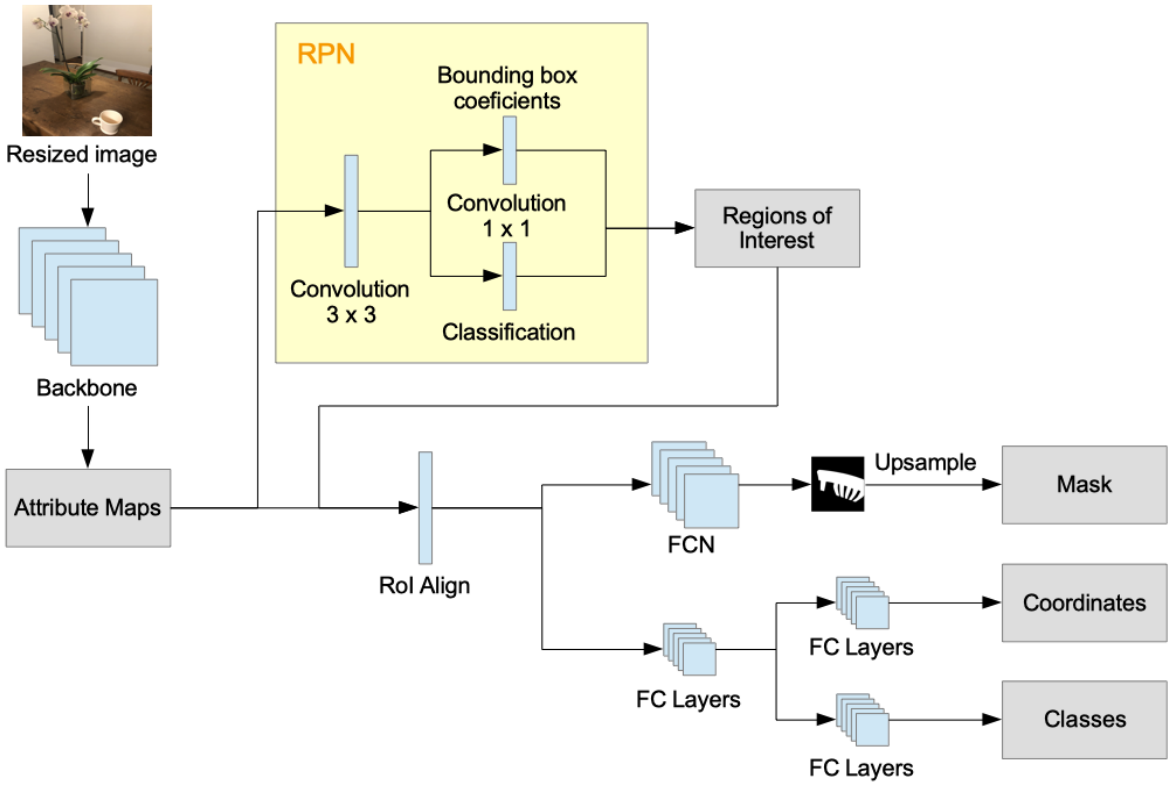


Figure 10 – Main components of a Mask R-CNN network.

3
Proposed Methods

The research work that encompasses the entire process of estimating the parameters of a ferromagnetic foreign body from a portable device was divided into three main steps:

1) Reading data from a GMR magnetometer and data from distance, accelerometer, and gyroscope sensors, integrated into a Raspberry Pi platform;
2) The tracking of the portable device from a monocular camera integrated into the device so that the coordinates in which the magnetic field is measured are known, as well as any inclinations of the device in relation to the inertial coordinate system;
3) The solution of the inverse problem using the data collected in the previous steps to find the center of the needle, its depth, its angle of inclination and its angle of rotation.

The first stage is the subject of another master's research in development at PósMQI/PUC-Rio, and will provide magnetic field measurements over the scanned area, together with an initial estimation of the device’s positions and orientations at each magnetic field measurement point, based on the physical sensors. This master's research sought to solve the last two stages of the work from computer simulations, presenting methodologies based mainly on deep learning networks for the various problems raised. Figure 11 illustrates the three stages of the complete research work, with the stages comprised in this research marked in yellow and red. In this chapter, we seek to present the proposed methods for solving these two steps.

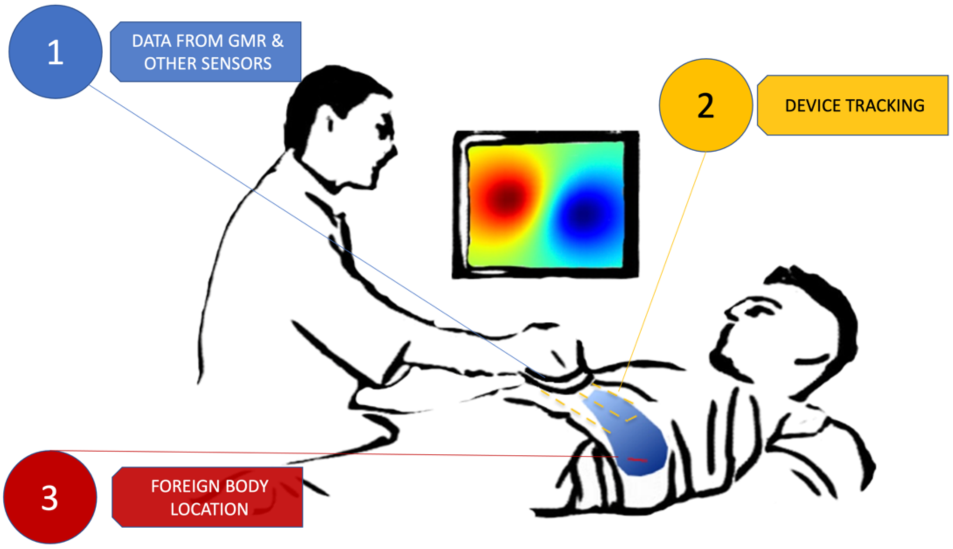


Figure 11 – Steps of the research work that encompasses the entire process of discovering foreign body parameters from a portable device. Steps 2 and 3 are the objects of this research.

For the whole project to be viable, it is necessary to consider the existence of a known pattern on the patient's skin, which serves as a reference image for the camera tracking and for presenting the final data to the health professional. In a surgical situation, the patient would initially undergo a radiographic examination, which would indicate an approximate position of the foreign body, and, from that position, the health professional would place such a reference pattern (e.g., by using a sticker) on the patient's skin, approximately aligned with the foreign body, which will be considered as the origin of the 3D coordinate system (because the system is portable, there is no fixed frame of reference), making it possible to assign coordinates to the magnetic field maps and allowing the solution of the inverse magnetic problem. Figure 12 shows the pattern used in the simulations of this dissertation as a reference adhesive to be applied to the patient's skin.

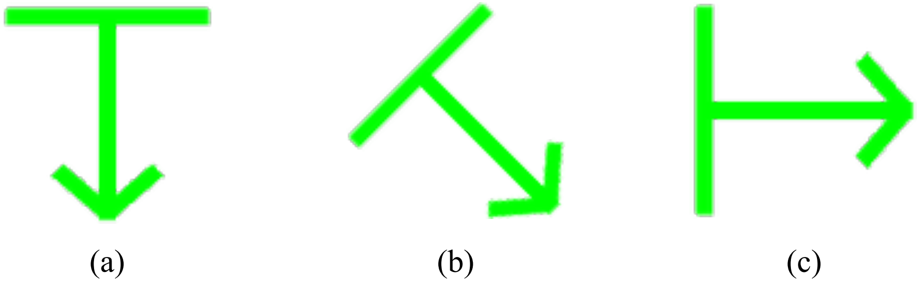


Figure 12 – Pattern used in simulations in the position of *yaw* = (a) 0º, (b) 45º, and (c) 90º.

In the following sections, the methods applied to the solution of both stages that this research deals with will be detailed.

3.1 Handheld device tracking

For a correct mapping of the magnetic field generated by the foreign body using a portable device, it is necessary to create a strategy for the location in the three-dimensional space of the device in relation to the human body at all measurement points so that it is possible to map the magnetic field in the form $\vec{B}(x, y, z)$, in addition to knowing the rotation angles concerning the x (roll or bank), y (pitch or elevation), and z (yaw or heading) axes.

Ideally, the z, roll, pitch, and yaw values would be constant in a real application, with roll, pitch, and yaw always equal to zero, which would greatly facilitate the experiment and improve the accuracy of the results. On a handheld device, however, it is not possible to guarantee that these values are constant, especially the z and yaw values, which are more difficult to control manually.

Thus, the strategy used in this research sought to estimate the x, y, z position values and the yaw value using only a low-cost monocular camera integrated into the device. The roll and pitch angles were not considered in this research because their combination yielded very similar patterns that could not be distinguished with the chosen methodology and because they can be obtained by the device's sensors and added to the values obtained by the methodology developed here.

In addition to the aforementioned adhesive pattern, it is also necessary to establish a reference camera so that it is possible to convert pixel values to centimeters. For the simulations, a 5th generation iPod monocular camera was considered, with a focal length of 3.3 mm and an aperture of f/2.4, which allowed an adhesive pattern with 4 cm of length to correspond to 349 pixels (px) in an image with a distance of 20 cm from the camera to the pattern. Ideally, the camera should have a shorter focal length to be used closer to the skin, as the GMR sensor loses accuracy as it moves away from the ferromagnetic foreign body.

Figure 13 shows an overview of the system used for tracking with all its components, each component being detailed in the following subsections.

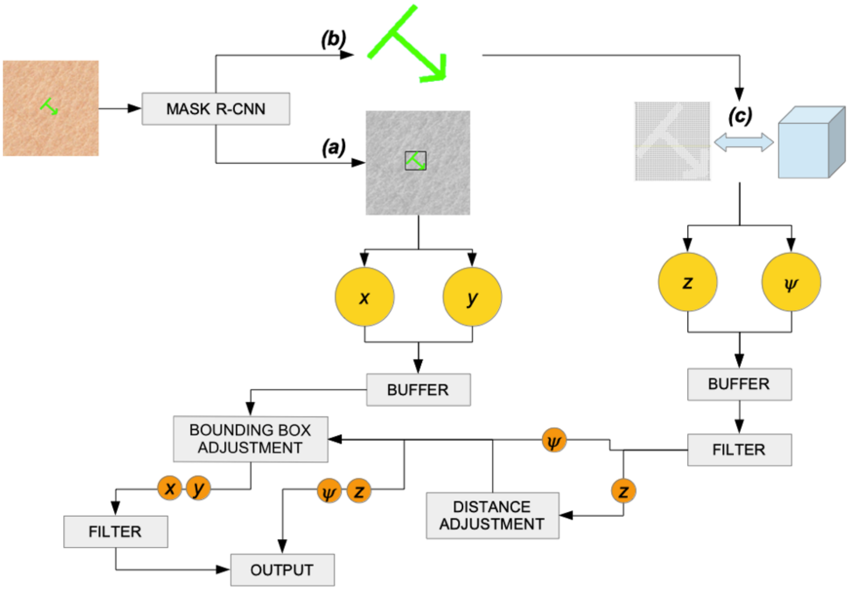


Figure 13 – Proposed system and its components for tracking the portable device: Mask R-CNN network detecting the pattern position (a) and the pattern semantic segmentation (b); calculation of yaw and *z* (c); buffers, filters, and bounding box and distance adjustments.

3.1.1 Mask R-CNN network

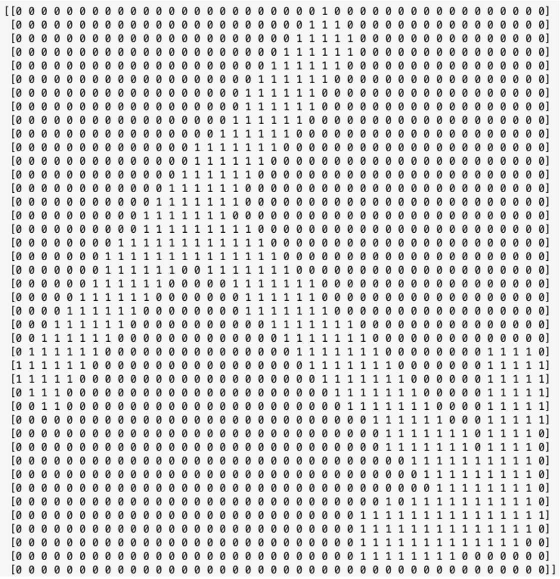
The first step for each image captured by the device's camera is performed by a Mask R-CNN network (Figure 10), responsible for performing two operations in parallel:

1) Detect in the image the position of the adhesive pattern on the patient's skin, returning a surrounding rectangle (bounding box) with the coordinates of the rectangle's upper left corner and its height and width. The center of this enclosing rectangle defines the initial x and y positioning values considered by the system (Figure 13a); and
2) Recognize the image pixels belonging to the adhesive pattern on the patient's skin. These pixels are used in the next step to calculate the yaw and *z* values (Figure 13b).

3.1.2 Yaw and *z* calculation

To obtain the yaw values, a three-dimensional matrix was created containing the representation of the adhesive pattern in all possible rotations in the z-axis at 1°

steps. Each possible rotation is represented by a binary array 42 px wide by 42 px high, where the values “1” represents the pattern. Figure 14 shows an example of a matrix for the 45º yaw angle.



```
[[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0]
 [1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1]
 [1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1]
 [0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1]
 [0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

Figure 14 – Binary array for the pattern with yaw = 45º.

The mask received from the semantic segmentation process (Figure 13b) is transformed into a binary matrix, keeping only the values of the smallest rectangle representing the pattern. This rectangle is resized so that its longest side is 42 pixels wide, and the shortest side is filled with zeros to get a final array of 42 x 42 px. The inner product between this matrix and the binary matrix corresponding to each yaw value is then calculated. The estimated value for yaw is the one that corresponds to the highest value of these internal products. It is also possible to use negative values instead of zeros in the binary input matrix, penalizing values outside the input mask. For the experiments in this research, however, the matrix with zeros produced the best results. For this step, a CNN regression network that receives as input the mask resulting from the semantic segmentation and returns the value of yaw was also tested in place of the three-dimensional matrix. The CNN network, however, produced results similar to the ones obtained by the three-dimensional matrix but with a longer processing time and, therefore, it was not used.

Another three-dimensional matrix was created with the dimensions of the enclosing rectangle of a reference pattern for each yaw value, the distance *z* of this pattern being known based on the reference camera. From the input mask, reduced only to the rectangle that best represents the pattern, we calculate the ratio of this mask to the reference pattern of the already known yaw value and multiply by the reference *z* value to obtain the estimated *z* value.

3.1.3 Buffer

At the output of the two branches of the Mask R-CNN network, a buffer was applied to process the values obtained for *x*, *y*, *z,* and yaw. This buffer stores the last obtained values and replaces missing values or outliers with others using linear interpolation. Outliers are considered values beyond two standard deviations from the mean of the buffer values. In the case of the simulations, the buffer was not necessary due to the highly controlled environment. On the other hand, in tests with the reference camera, an 11-value buffer was enough to correct the outliers without causing much delay in the results.

3.1.4 Filter

Since the bounding box and pattern mask returned by the Mask R-CNN network suffer small fluctuations in their values, even in more controlled environments, as shown later in Chapter 5, we applied the Savitzky–Golay filter [53] to smooth all values in different places of the system: for yaw and *z* values, which serve as a basis for the adjustment of the other values, the filter was tested being applied before and after the adjustments; for *x* and *y* values, the filter is applied immediately before the output, after all the adjustments have been made. The filter was configured as a first-order polynomial with a window length of 7 coefficients and no derivative, making its processing quite fast.

3.1.5 Distance adjustment

The masks generated by the semantic segmentation tend to have the corners rounded by the network, producing patterns a little smaller than the real one, as illustrated in Figure 15, where can be seen how the arrowhead and the edges of the base of the pattern are rounded in the semantic segmentation process, generating a pattern prediction a little smaller than the actual value.

Experiments with different pattern sizes were performed at various rotations to compare the actual and predicted pattern sizes. The mean of all values was used as a factor to be multiplied by all pattern size values in the predictions. The final factor used was 1.072.

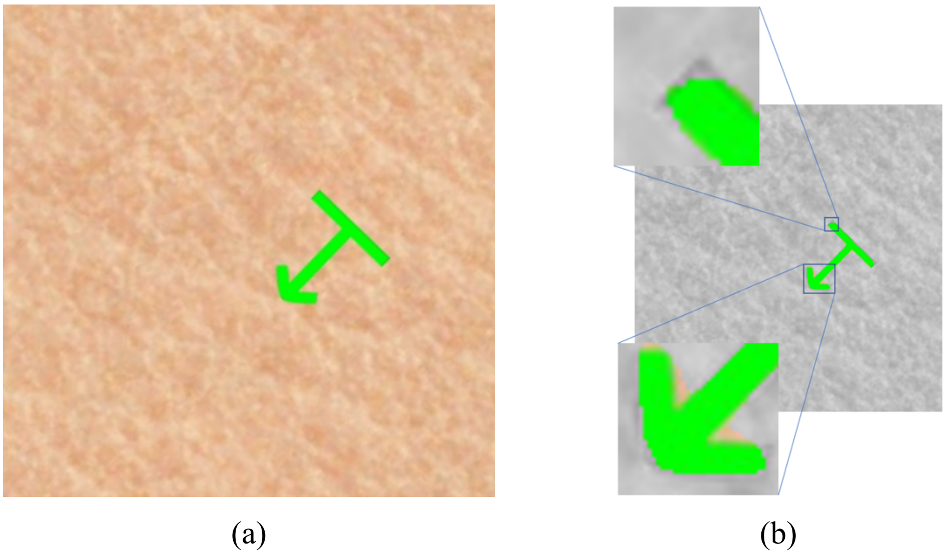
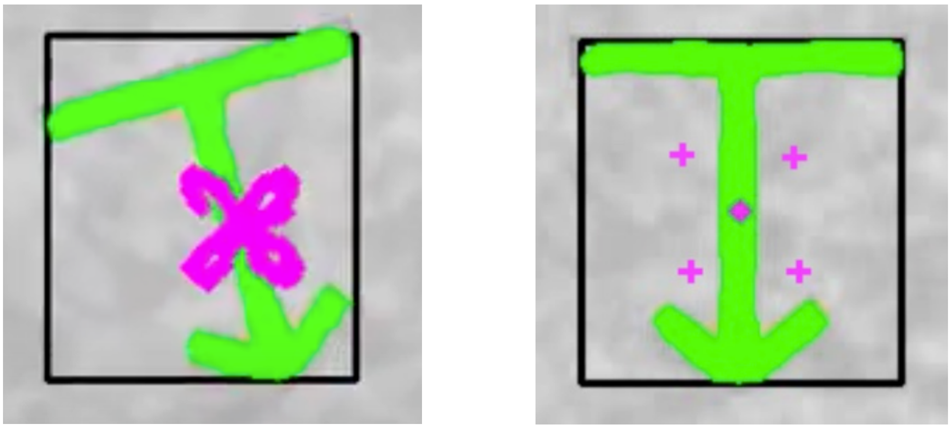


Figure 15 – Visualization of frame No. 253 of the simulation (a) and the semantic segmentation result (b) with a zoom in specific parts of the pattern showing the rounding effect, with the recognized pixels marked in green and the rest of the image transformed into grayscale.

3.1.6 Bounding Box Adjustment

Another adjustment that became necessary was due to the difference between the center of the pattern and the center of the bounding box generated by the object detection network when the pattern rotates according to the yaw angle. Through experimental analysis, it was observed that this difference reaches the highest value at the angles of 45º, 135º, 225º, and 315º, with a displacement of about 14 px in the x and y values for the reference pattern with 88 px height, returning to zero every 90 degrees. Figure 16a shows the difference between the center of the bounding box and the center of the pattern for each angle of rotation on the z-axis during a 360° rotation and Figure 16b shows the points of greatest difference between the centers.

For this adjustment, this behavior was considered linear instead of the behavior shown in Figure 16a. To obtain the adjustment, we used the yaw value and the ratio between the reference bounding box and the predicted bounding box, both the values found during the yaw and *z* calculation phase, the *z* value being already refined by the distance adjustment.



(a) (b)

Figure 16 – A simulation of the pattern during a 360º rotation around the *z*-axis resulting from the object detection network with (a) pink marking representing the difference between the center of the bounding box and the center of the pattern and (b) a pink diamond showing the center of the bounding box at yaw = 0º and pink crosses showing the highest difference between the center of the bounding box and the center of the pattern during the simulation.

As a first step, the value of 14 px is multiplied by the ratio between the sizes of the found pattern and the reference pattern; that is, a pattern twice the size of the reference pattern generates a maximum displacement of 28 px, while a pattern with half the size yields a maximum offset of 7 px. Then, transposing the predicted yaw to values between 0º and 90º, we obtain as a multiplier factor the value 1 for the angle of 45º and the value 0 for the angles of 0º or 90º, with the other angles being linearly converted to an intermediate value between 0 and 1. The final result will be the multiplication between this multiplier factor and the maximum displacement, added to the values of x and y found in the object detection step.

3.1.7 Output

At the system’s output, the values of *x*, *y*, and *z* are converted from pixels to centimeters using the values from the reference camera. From the value of *z* obtained in the prediction, equivalence is performed with the size of the 4 cm adhesive pattern, corresponding to 349 px at a distance of 20 cm from the camera to the pattern. After this conversion, the x and y values must still be converted, based on the yaw (ψ) value, between the camera coordinate system and the inertial reference system, using (7) and (8). Figure 17 shows the two coordinate systems related to the camera, the adhesive pattern, and the foreign body.

$$\mathrm{X}_c = (x.\cos\psi) - (y.\sin\psi) \tag{7}$$
$$\mathrm{Y}_c = (x.\sin\psi) + (y.\cos\psi) \tag{8}$$

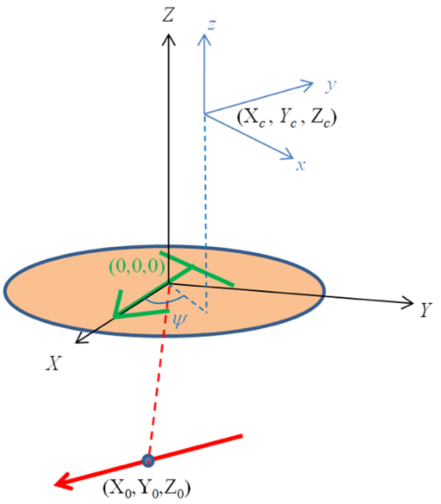


Figure 17 – The blue axes represent the values found by the simulation before the conversion to the inertial reference system, represented by the black axes, using (7) and (8). The position of the adhesive pattern is shown in green, and the center of the foreign body, with the coordinates (X_0, Y_0, Z_0), is shown in red.

3.2 Foreign body location

This section presents the input data required for locating the foreign body, as well as the expected data output from our system.

From the various magnetic field readings performed by the GMR sensor and the device tracking data collected by the system proposed in this chapter, it is possible to generate a representative matrix of magnetic flux density values and use a CNN to discover the parameters of foreign body positioning.

3.2.1 CNN Input Data

In the simulation developed in this dissertation, the magnetic field map used as input for the CNN is generated based on the Biot-Savart law for a magnetic dipole, integrated along the length of a needle and transformed into a color map to facilitate visualization throughout the process.

Most images generated from imperfect or off-center scans have missing pixels near their edges. Since interpolation cannot be applied in these situations, a procedure was created to fill in those missing pixels. The procedure checks each pixel in the image, from the center to its edges, in a spiral motion. If any pixels are missing, it fills them with the average value of the available surrounding pixels. Figure 18 shows an example of an image generated after a scan simulation without (Figure 18a) and with (Figure 18b) the procedure to fill in the missing pixels. The magnetic field image with the missing pixels filled in serves as input to the CNN.

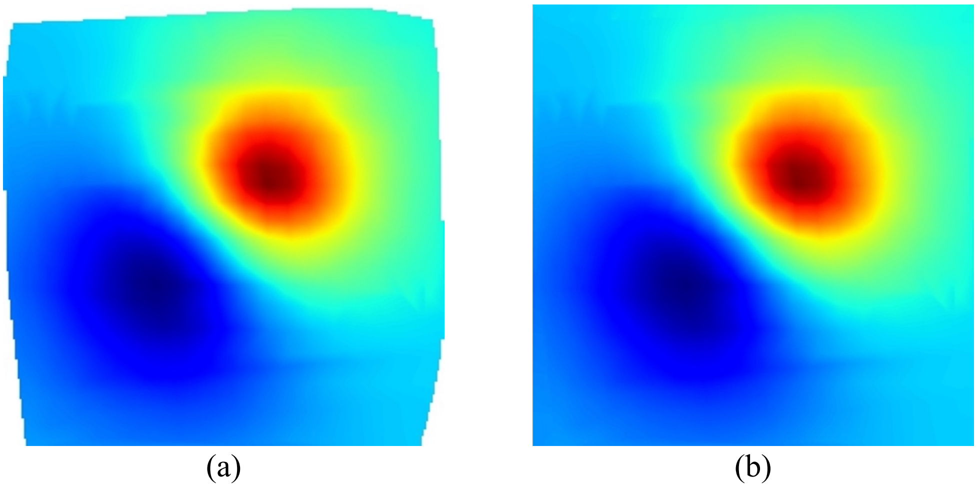


Figure 18 – Image examples of the magnetic field generated after a scanning simulation without (a) and with (b) the method for filling missing pixels.

3.2.2
CNN output data

Figure 19 illustrates a needle with the expected network output parameters. The angle of inclination θ, the angle of rotation ϕ, and the depth h of the needle are direct outputs of the network. The geometric center of the needle, represented in Figure 19 by p, is obtained by calculating the center between the extreme values of the magnetic field and adding to it an offset value Δ, also provided by CNN. Figure 20 shows an image of the magnetic field generated by the ferromagnetic foreign body, which serves as an input to the CNN, with markings added to highlight some parameters. The white markings are the minimum and maximum values of the magnetic field, and the black marking is the midpoint between these extremes, while the red marking is the geometric center of the needle.

The CNN network returns the value Δ, which corresponds to the Euclidean distance between the center of the extreme values of the magnetic field and the geometric center of the needle. The calculation of Δ follows the procedures presented in [3] and can be obtained by the ratio between the weighted sum of field values and position values that are below 95 % of the minimum extreme and the weighted sum of field values and position values that are above 95 % of the maximum extreme. In the case of a needle with an inclination θ = 0º, the value of Δ will also be zero, while in the case of an inclined needle, the geometric center tends to move away from the magnetic center towards the end of the foreign body closest to the sensor.

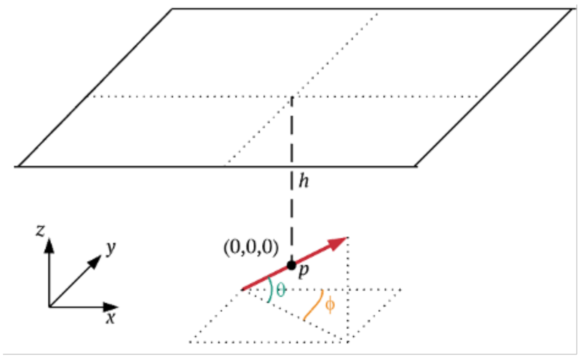


Figure 19 – Needle position in red with (0,0,0) as its center point (*p*), showing needle inclination angle (θ), rotation angle (ϕ), and depth (*h*).

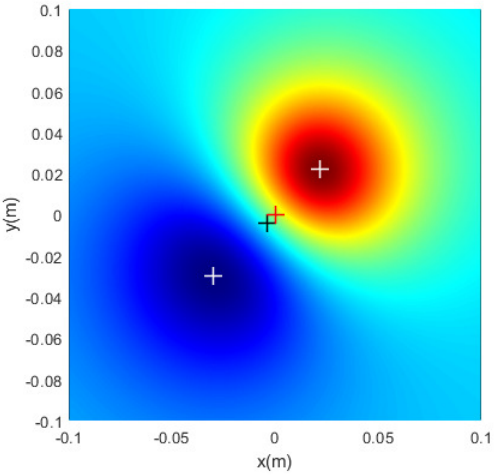


Figure 20 – Image of the color map representing the magnetic field with white markings on its extreme values, a black marking on the central position of these extremes, and a red marking on the geometric center of the ferromagnetic object that generated the map. The needle is aligned with the white markings.

3.3 Integration of the steps presented

Figure 21 shows a detailed outline of the steps presented by Figure 11 in the introduction to this chapter and detailed in the previous sections. For each reading made by the GMR sensor, step 2 is applied, in which the image obtained by the camera at the time of the sample is processed by the tracking system to obtain the positioning and yaw values of the device. After the scan is completed, the GMR readings are combined with the tracking data to produce the image of the magnetic field. In this research, as the GMR sensor reading data are not available, the tracking data goes through the magnetic field simulator, which generates the final map that serves as input to step 3. In this last step, the CNN predicts the location parameters of the foreign body.

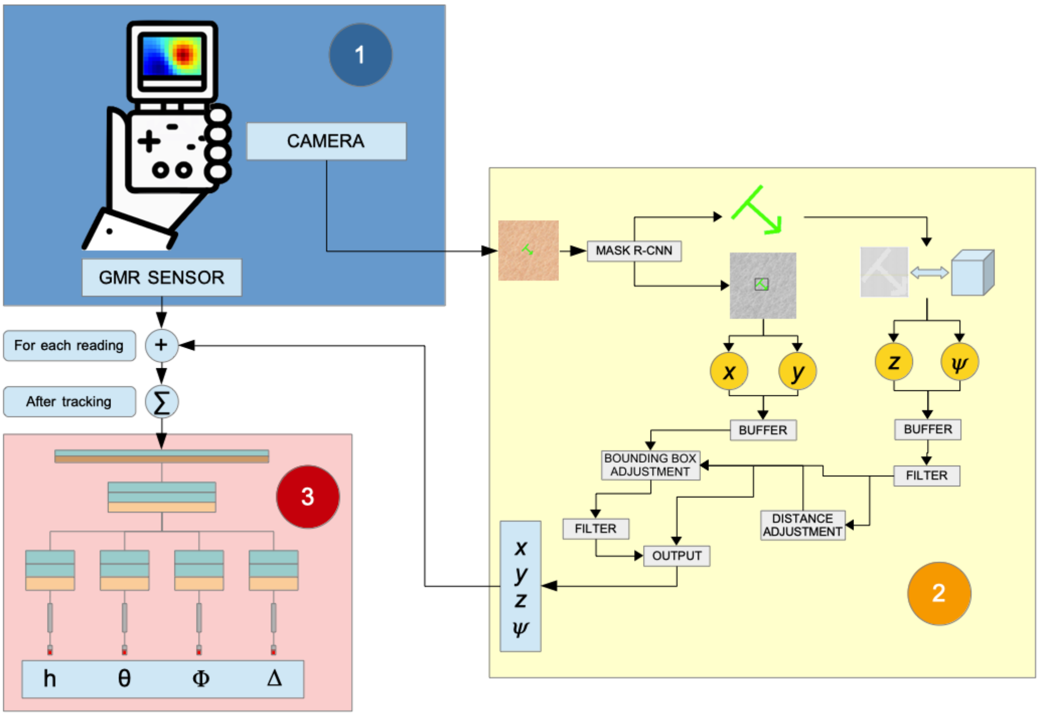


Figure 21 – Detailed outline of the steps presented in this chapter: (1) collecting data from the handheld device; (2) the device tracking system and; (3) the network that receives the collected magnetic field map and provides the foreign body localization data.

4 CNN training

This chapter covers the CNN training performed and the parameters and metrics used in these training for the two CNNs that make up the framework developed in this dissertation: the Mask R-CNN used to track the portable device, and the CNN used to locate the foreign body.

An independent simulation was performed for each type of network, and a final simulation was performed using the two networks together. It is important to highlight that, in the case of the Mask R-CNN, the same network was used in both simulations; on the other hand, the network used to locate the foreign body was trained with different datasets and parameters for each simulation, despite using a very similar architecture.

4.1 Development environment

As a development and training environment for the convolutional networks, the Google Collaborative Platform [54] was used with the Python programming language in version 3.7 [55] and the machine learning libraries Tensorflow in version 1.15 [56] and Keras in version 2.3.1 [4]. The platform has a GPU and 12 Gb of RAM, in addition to 20 Gb of disk space shared with other Google services.

4.2 Mask R-CNN

4.2.1 Dataset

To create the dataset, images of the adhesive pattern were superimposed on images of skin texture collected on the internet. Generating these images artificially allows us to create a wide variety of combinations of skin types and pattern placement, which is very important for creating a network with good generalization, although it doesn't exactly match a real image of the patient.

For the adhesive pattern, images with variations of roll, pitch, and yaw were generated since, at this time of the research, the intention was to work with the device's rotation in the three axes. The pattern was implemented in green with values in the RGB system of (0,255,0) as this is a hue rarely found on the naked body, except in cases of tattoos, being a way to facilitate its identification by the CNN or any other method of recognition. However, as the proposed device is portable and can be used in different lighting situations, such as remote assistance or assistance in a mobile ICU, three variations of green were created for each image of the pattern: the first with the reference green; the second with random values of green between 205 and 255 and random values of red and blue between 0 and 25; and the last with random values of green between 153 and 255, and random values of red and blue between 0 and 50. In total, 1 836 images of the pattern were generated, considering the variations in inclination angles and shades of green.

The selected skin texture images sought to represent not only the different types of skin tone existing but also different types of ambient lighting, different body hair thicknesses, the presence of wounds or changes in the skin, the presence of black and colored tattoos, and the presence of other objects not associated with the patient but positioned nearby. This variety seeks to create a network capable of recognizing the pattern in the most diverse situations, making the algorithm more robust than a methodology based only on image colors. Initially, 249 images were selected, which were enlarged using rotation in 90º, 180º, and 270º, making a final total of 996 images.

In the end, each adhesive pattern image was resized to simulate different distances and combined with a skin pattern image, generating 1 836 final images. Figure 22 shows some examples of the final images, resized to a square format.

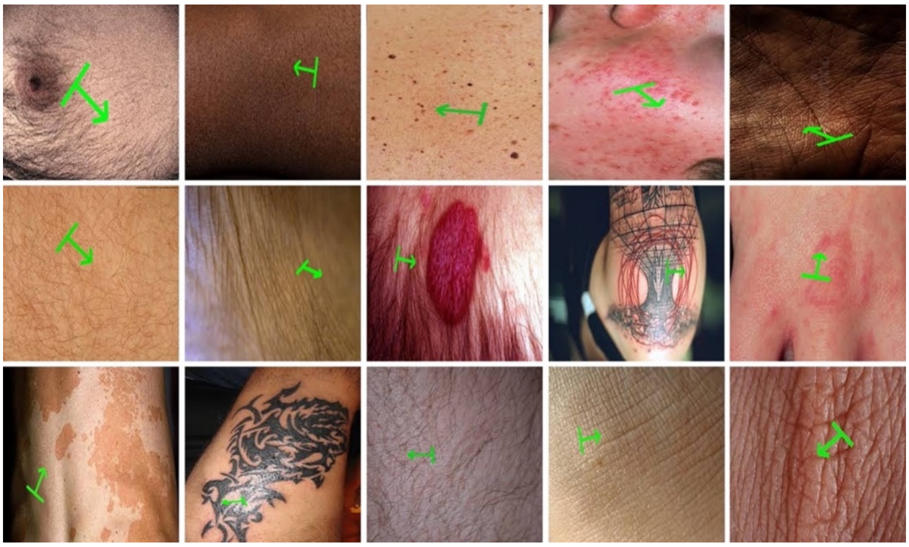


Figure 22 – Example of images generated to compose the dataset of the Mask R-CNN network.

After creating the final set of images, each image was manually annotated using the VGG Image Annotator software [57] to indicate the polygon that best represents the adhesive pattern. The final number of images in the dataset could easily be extended with new combinations of the adhesive pattern with the skin images. However, because the annotation process is laborious and time-consuming, the final number of images was limited to the 1 836 files already mentioned and, in the training process, new images were generated along with the corresponding annotations using data augmentation, as presented in the following subsections. Figure 23 shows an annotation made on one of the dataset images.

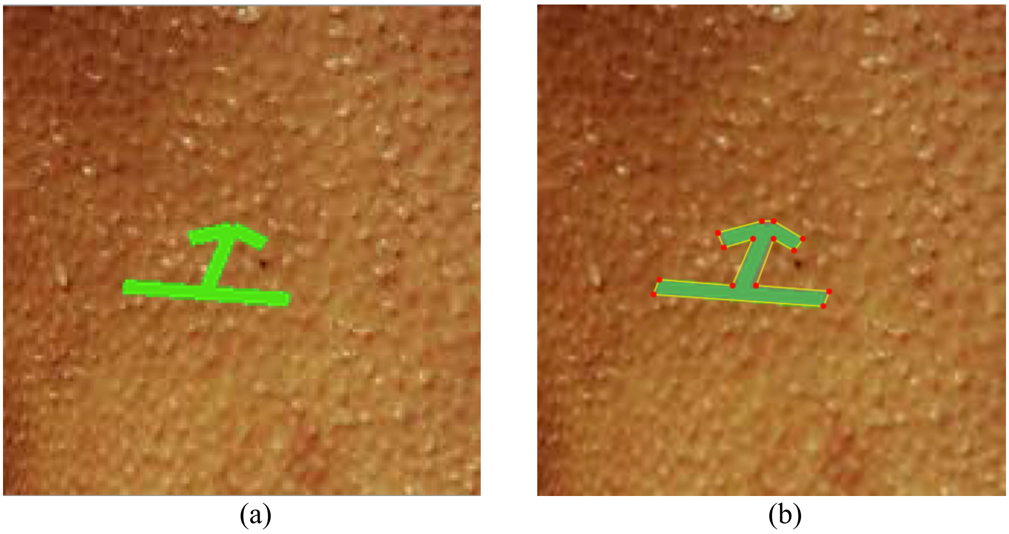


Figure 23 – Dataset image detail without (a) and with (b) the polygon indicating the adhesive pattern.

4.2.2 Training parameters

As a basis for the convolutional network training process, a previously realized implementation of the Mask R-CNN network for Python 3, Keras, and TensorFlow [58] was used. For the framework backbone (Figure 9), a ResNet101 network [52] pre-trained with the MS COCO dataset [59] was used. Due to the reduced size of the dataset described in section 4.2.1 for a network of this size, only the stages five and higher of the network were trained, leaving the others fixed with the pre-trained weights. To increase the number of images in the dataset (data augmentation), horizontal and vertical flip, rotation at every 90º, the addition of gaussian blur, and pixel multiplication were used, with these operations being applied and combined at random during training.

The images were resized to 2048 px wide by 2048 px high with three color channels for training. Among the parameters used, the training had a learning rate of 0.01, a value of 500 steps per epoch with 100 validation steps per epoch, which in this case it represents a training set of 250 images with 50 validation images for each epoch and a limit for selecting regions of interest of 90 %. For all other parameters, the code repository defaults were used.

4.2.3 Training metrics

The model was trained in different configurations of the parameters presented in the previous subsection, including data augmentation variations, until reaching the parameters with the best results. Each training was run for a maximum of 120 epochs, usually interrupted by Google Collaborative runtime outages. In this case, the new training was started from the final weights of the previous training.

No automatic way to end the training (early stopping) was used. Decisions were taken based on a visual analysis of the graphs for the total loss function over the time for training and validation and of the result obtained with the test images.

The total network loss function is the sum of the several Mask R-CNN network loss functions. The first set of losses concerns the RPN (described in subsection 2.2.3 and illustrated in Figure 10) and can be divided into the classification loss and the bounding box loss, according to

$$\mathcal{L}(\{p_i\}) = \underbrace{\frac{1}{N_{cls}}\sum_i BCE(p_i, \hat{p}_i)}_{classification} + \underbrace{\frac{\lambda}{N_{reg}}\sum_i \sum_{j\in\{x,y,w,h\}} \hat{p}R\big(t_{ij} - \hat{t}_{ij}\big)}_{bounding\ box}, \quad (9)$$

where p_i is the output value for anchor i, $\hat{p}_i$ is the real value for anchor i, N_{cls} is the size of the mini-batch, and N_{reg} is the number of anchor positions.

The function R is given by

$$R(x) = \begin{cases} 0{,}5x^2, & if\ |x| < 1 \\ |x| - 0{,}5, & otherwise \end{cases}, \quad (10)$$

and the input parameters of $R(x)$ are

$$t_{ix} = \frac{(x_i - x_{ia})}{w_{ia}}, \quad (11)$$

$$t_{iy} = \frac{(y_i - y_{ia})}{h_{ia}}, \quad (12)$$

$$t_{iw} = log\left(\frac{w_i}{w_{ia}}\right), \quad (13)$$

$$t_{ih} = log\left(\frac{h_i}{h_{ia}}\right), \quad (14)$$

$$\hat{t}_{ix} = \frac{(\hat{x}_i - x_{ia})}{w_{ia}}, \quad (15)$$

$$\hat{t}_{iy} = \frac{(\hat{y}_i - y_{ia})}{h_{ia}}, \quad (16)$$

$$\hat{t}_{iw} = log\left(\frac{\hat{w}_i}{w_{ia}}\right), \text{ and} \quad (17)$$

$$\hat{t}_{ih} = log\left(\frac{\hat{h}_i}{h_{ia}}\right). \quad (18)$$

In equations (11) to (18), x_i, y_i, w_i, and h_i are the central coordinates, width, and height of rectangle i; x_i, x_{ia} and $\hat{x}_{ia}$ are the x values for the prediction, for the anchor and ideal rectangle value, respectively (same for y_i, w_i , and h_i).

Finally, the BCE in equation (9) is the binary cross entropy, given by

$$\mathcal{L}_{BCE} = \sum_{i=1}^{2} t_i log(p_i), \quad (19)$$

where t_i is the true binary value and p_i is the softmax probability (equation 6) for class i.

The second set of losses concerns the framework's final network (Figure 10) with its three outputs, according to

$$\mathcal{L}(\{p_i\}) = \sum_i \left[\underbrace{-log(p_i)}_{classification} + \underbrace{\sum_{j\in\{x,y,w,h\}} R\big(t_{ij} - \hat{t}_{ij}\big)}_{bounding\ box} + \underbrace{\sum_{1\le k\le K} BCE(p_{ik}, \hat{p}_{ik})}_{mask} \right] \quad (20)$$

where p_i is the true class output/probability value for rectangle *i* and *K* is the number of classes.

Figure 24 shows the evolution of the total loss over the 40 training epochs of the network chosen for this research, and Figure 25 shows some examples of the application of the test set on the network with the mask found and the confidence value of the network in the class found, which, in this case, is naturally high because a single class is used.

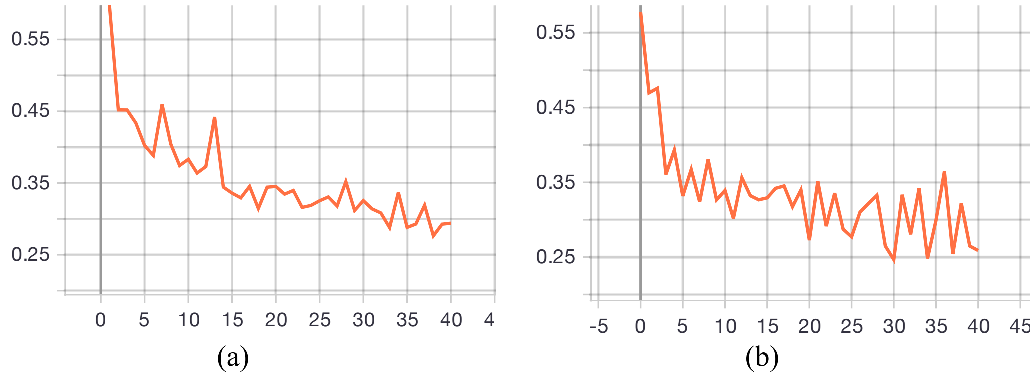


Figure 24 – Graph representing the value of the loss function (*y*-axis) over the epochs (*x*-axis) for training (a) and validation (b) of the Mask R-CNN network used in this research.

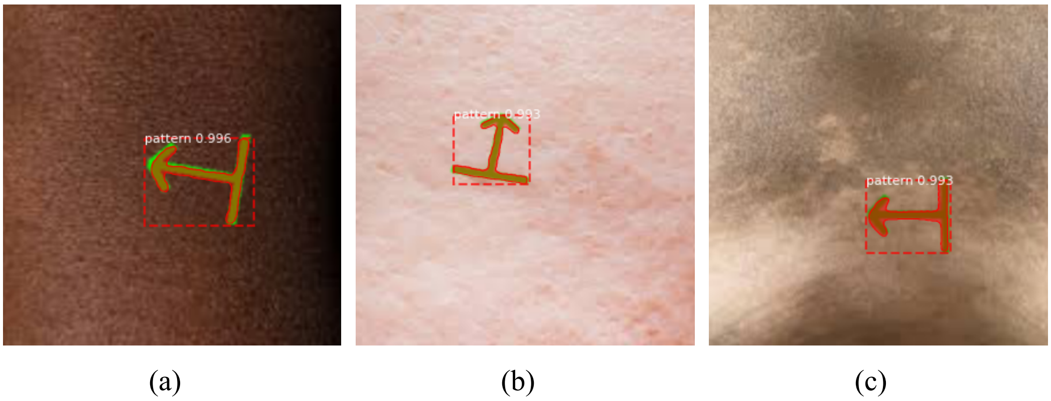


Figure 25 – Prediction of the Mask R-CNN network for images in the test set, indicating the mask found and a confidence value in predicting the class of 0.996 (a), 0.993 (b), and 0.993 (c).

4.3 CNN for foreign body location

4.3.1 Architecture

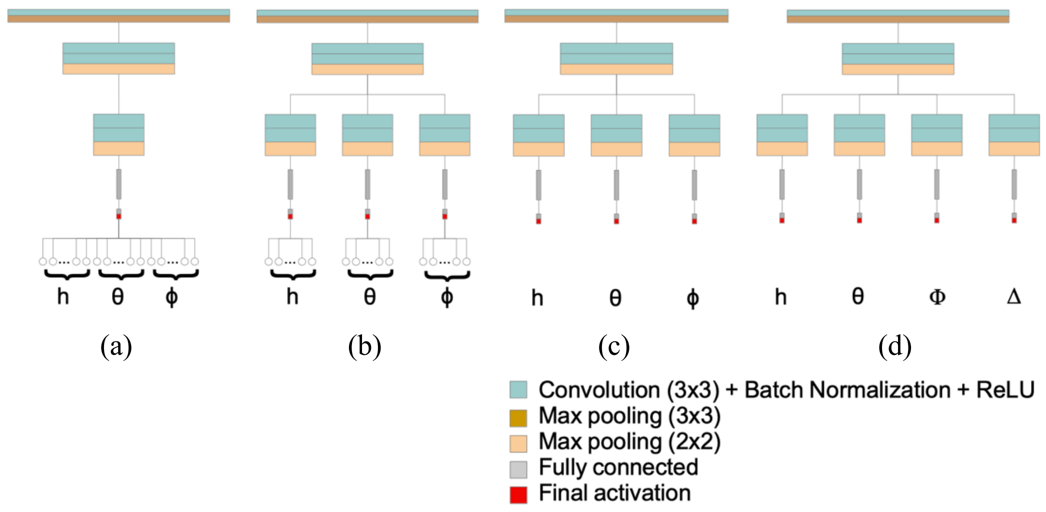
For the location of the foreign body, an architecture based on the VGG network [51] was used, with fewer convolutional layers due to the low complexity of the problem. As explained in the introduction to the chapter, different networks were used in the two simulations performed for the location of the foreign body. However, the architecture used in all tests was similar, varying only the number of learnable parameters and output values that each network produces. Figure 26 illustrates the different architectures used, detailing their layers and the type of output produced. Networks (a) and (b) are classification networks, with network (a) producing a single probability vector for all output values and network (b) producing a probability vector for each output value. Networks (c) and (d) are regression networks, the only difference between them being the fact that network (d) has an output for the value of Δ. The final activation function and loss function also vary depending on the type of output desired. Regression networks use a linear activation function and an MSE (mean squared error) loss function, given by

$$MSE(\hat{\theta}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{\theta}_i - \theta_i)^2, \tag{21}$$

where θ is a scalar parameter, $\hat{\theta}$ is an estimator of the scalar parameter, and N is the number of samples. The network (a), as it has a single output vector, uses a Sigmoid activation function (equation 1) and a binary cross entropy loss function (equation 10), producing an individualized probability for each position of the output vector. The network (b) uses a softmax activation function (equation 6) and a categorical cross entropy loss function, given by

$$\mathcal{L}_{CE} = \sum_{i=1}^{n} t_i log(p_i) \tag{22}$$

where t_i is the true value and p_i is the softmax probability (equation 6) for class i, producing a vector whose sum of probabilities equals 1 for each output value.



(a) (b) (c) (d)

Figure 26 – Architecture variations used in simulations for foreign body location.

4.3.2 Dataset

The two training sessions performed for the location of the foreign body had different datasets and training parameters, despite being very similar. In this and the following subsections and chapters, only the training of the model used in the final simulation will be detailed, which is an unfolding of the tests performed in the first simulation. Details of the initial simulation can be analyzed in [60], annex 1 to this dissertation.

The dataset for the final localization network was constructed by simulating the magnetic field data as described in subsection 3.2.1, simulating a reading from the handheld device during a full scan. Each scan corresponds to a path taken by the device in a zigzag pattern along a 20 cm x 20 cm area centered on the adhesive pattern.

The scans were created by applying random variations to a reference path on the values of x, y, z, and yaw, with the roll and pitch values kept constant at zero.

Four different irregular paths were created and, for each of them, 2 376 images were created representing the magnetic field generated by a magnetic foreign body of the straight needle type. These images cover a variation in depth (h) from 50 mm to 150 mm every 10 mm, a variation in inclination angle (θ) from -60° to 60° every 15°, and a variation in the rotation angle (ϕ) from 0° to 360° every 15°. The delta value (Δ) was calculated from the magnetic field values for each input according to the procedure presented in [3]. The final dataset comprises 9 504 images with corresponding *h*, θ, ϕ, and Δ values. Figure 27 shows in (a) an unaltered scan path, with an ideal magnetic field map, and in (b) to (e), the four paths used in the dataset, with their corresponding magnetic field maps for the parameters *h* = 60 mm, θ = 15º, ϕ = 45º, and Δ = 0.56 mm. The images went through the process described in subsection 3.2.1 to fill in missing pixels in the final image.

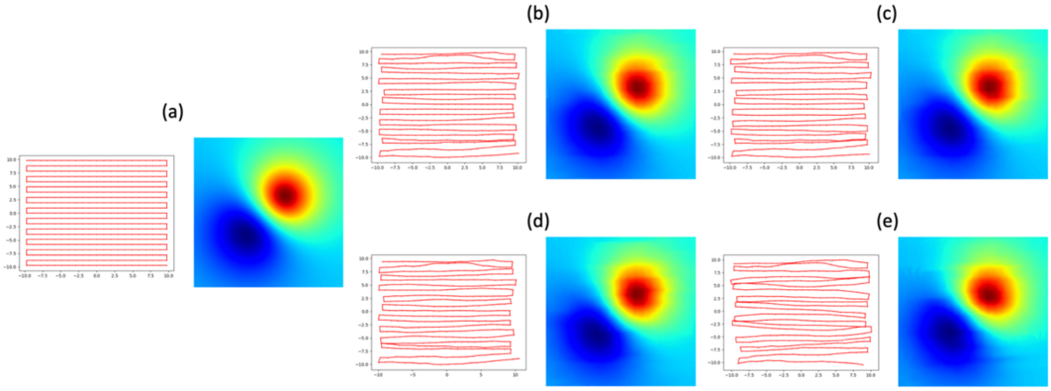


Figure 27 – A perfect scan and the generated magnetic field map (a) and the tracking variations used in the dataset (b, c, d, e) with the corresponding magnetic field map for h = 60 mm, θ = 15º, ϕ = 45º, and Δ = 0.56 mm.

4.3.3 Training parameters

The final configuration of the model, illustrated in Figure 26d, has a total of 1 534 788 parameters to be learned, with 56 960 of them being shared between *h*, θ, ϕ, and Δ. The final configuration of filters per layer can be seen in Figure 28. Since the rotation can also be inferred directly from the magnetic field values, an equivalent model without the rotation branch was also tested, with a total of 1 165 251 parameters to be learned. Both models, with and without the rotation branch, were also tested with absolute Δ values and with negative and positive Δ values.

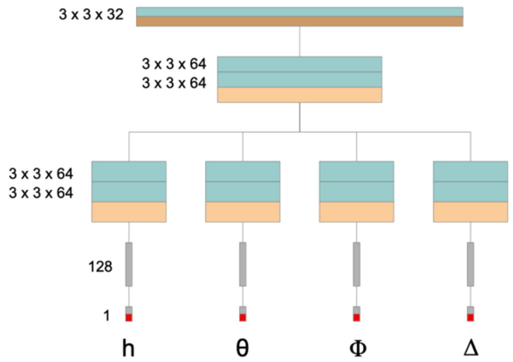


Figure 28 – Foreign body localization CNN showing the filter size and number of filters ($N \times N \times M$) for the convolutional layers and the number of units for the dense layers.

The network expects a input image 80 px per 80 px. The dataset was randomly separated into disjoint sets with 60 % of the data for training, 20 % for validation, and 20 % for testing. After experimental analysis, the Adam optimizer was used with a learning rate of 0.0001, and dropout of 0.25 for the convolutional layers and of 0.5 for the dense layer was used to prevent overfitting, when the network perfectly learns the training set but cannot generalize well to new input data. To control the training time, early stopping was used to end the training if there was no improvement in the loss function result in the validation for ten epochs, only the model with the best result being preserved during the entire training. Figure 29 shows the evolution of training and validation data while Table 1 shows the final values for all values associated with the magnetic source location parameters (h, θ, ϕ, and Δ) in MSE (equation 12) and MAE (mean absolute error), given by

$$MAE(\hat{\theta}) = \frac{1}{N}\sum_{i=1}^{N}|\theta_i - \hat{\theta}_i|, \tag{23}$$

where θ is a scalar parameter, $\hat{\theta}$ is an estimator of the scalar parameter, and N is the number of samples.

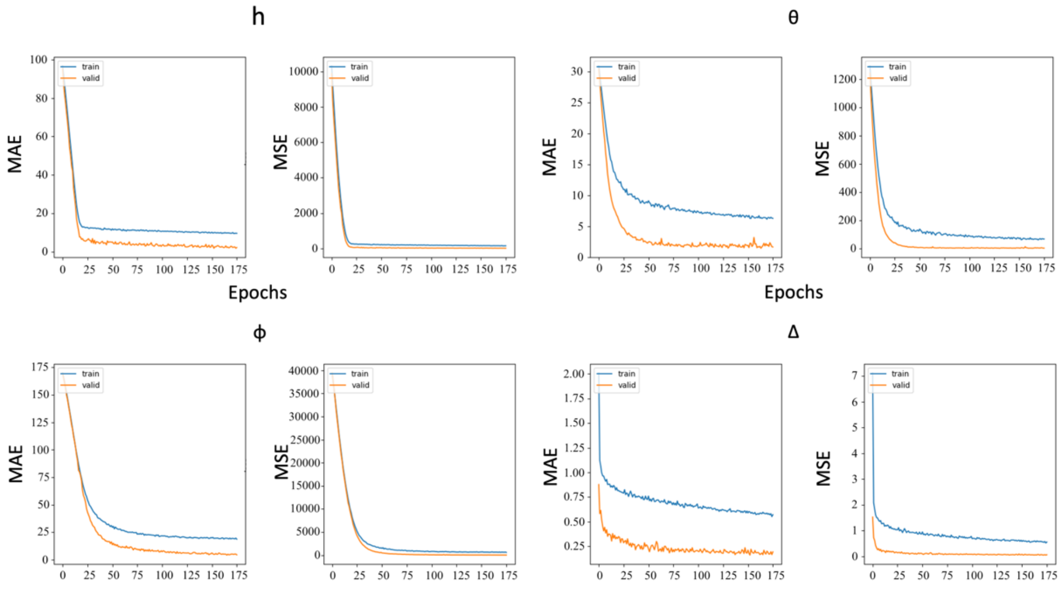


Figure 29 – Evolution of mean absolute error (MAE) and mean squared error (MSE) for training (blue) and validation (orange) data along the epochs for depth (*h*), inclination (θ), rotation (Φ), and delta (Δ) values.

Table 1 – Final mean absolute error (MAE) and mean squared error (MSE) values for training and validation data.

		h	θ	Φ	Δ
MAE	training	2.14	1.86	4.16	0.16
	validation	2.16	1.92	4.15	0.16
MSE	training	7.27	5.00	29.47	0.05
	validation	7.51	5.21	29.62	0.05

5
Results

This chapter presents the experimental results obtained in the simulations carried out. In section 5.1, only the simulation performed for the device's tracking system is analyzed, and section 5.2 describes the final simulation results, including the foreign body's location.

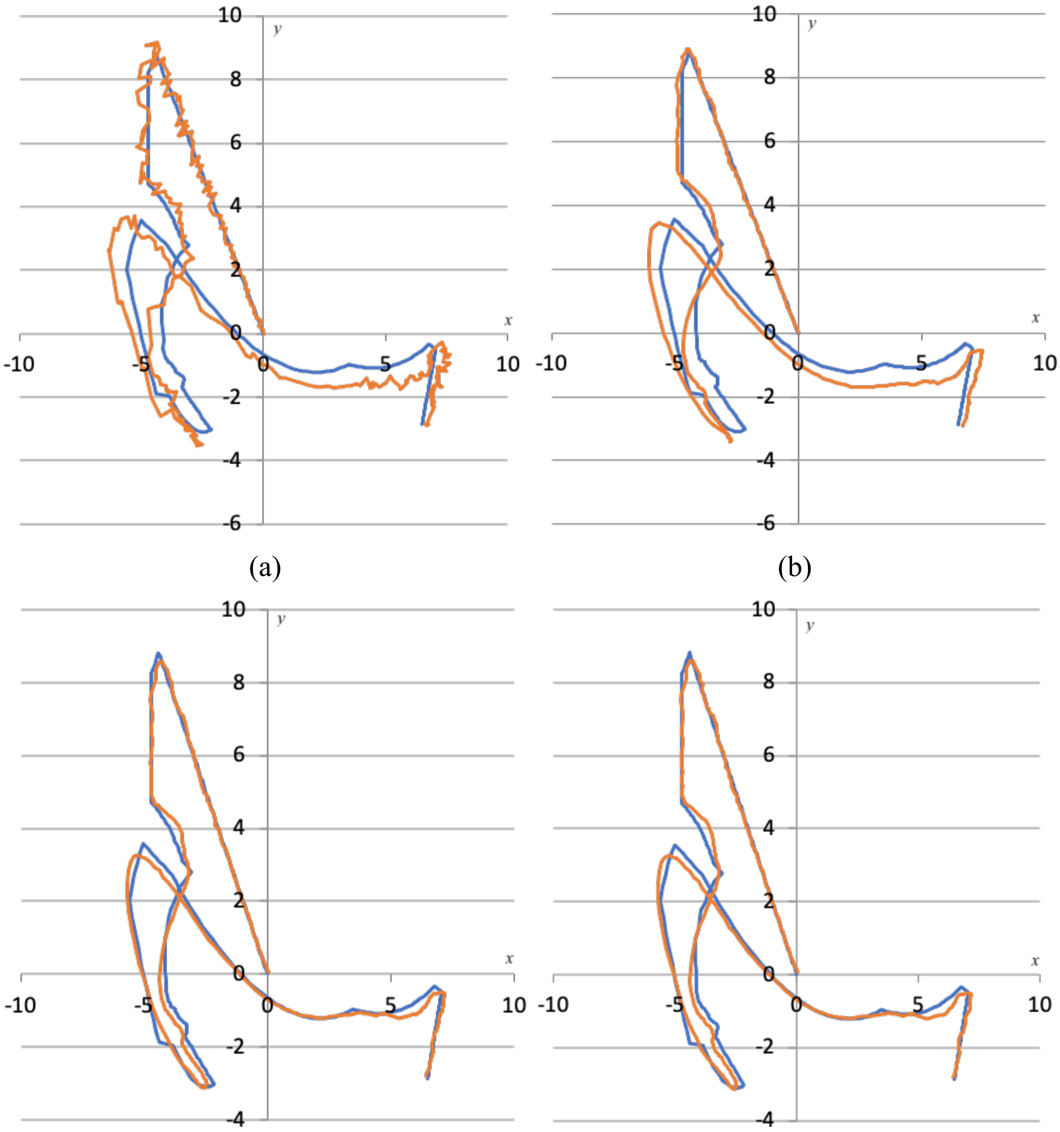
5.1
Mobile device tracking system results

The system adjustments shown in Figure 13 were created according to the needs that arose during the experiments carried out, having as a starting point the results of the Mask R-CNN network. Figure 30 shows the results for the *x* and *y* coordinates using four different system configurations.

Configuration A represents the output of the Mask R-CNN network without the proposed adjustments, and it is possible to observe how the predicted values oscillate around the real values. This fluctuation is intensified by varying the values of *z* and yaw, as both are used in converting pixels to centimeters and converting the camera coordinate system to the inertial reference system.

In configuration B, filters are included in the system to smooth out the values, solving the fluctuation problem and bringing the output values closer to the real values, especially when there is no rotation or variation in *z*.

In configurations C and D, all adjustments are used, obtaining results very close to the real ones, with the only difference between each configuration being the position of the *z* and yaw filters, which are applied at the system output in configuration C and before the adjustments in configuration D. This difference mainly affects the *x* and *y* values, since they are dependent on the yaw and *z* values for the bounding box adjustment and the conversion to the inertial reference system, although in the simulation both configurations presented similar results.



(a)

(b)

(c)

(d)

Figure 30 – View of the device's *x* and *y* positions for four different system configurations: without the proposed adjustments (A), with filters (B), with the complete set of adjustments, and filters applied at the output (C) or before the adjustments (D). The blue line represents the real values, and the orange line represents the output of the proposed system, with values in cm.

Table 2 presents, for each network output variable, considering the system configurations presented in Figure 30, the RMSE errors, given by

$$RMSE(\hat{\theta}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2}, \tag{24}$$

where θ is a scalar parameter, $\hat{\theta}$ is an estimator of the scalar parameter, and N is the number of samples.

The values present a slight improvement when applying only the filter (Fig. 30b). In contrast, with all the proposed adjustments (Fig. 30c and Fig. 30d), the complete system produces considerably lower errors for *x*, *y*, and *z*.

Table 2 – RMSE values for the different system configurations with the best results obtained marked in bold

System Configuration	x (cm)	y (cm)	z (cm)	yaw (º)
A	0.368	0.318	3.667	1.577
B	0.348	0.290	3.555	**1.059**
C	**0.157**	**0.091**	**1.159**	**1.059**
D	**0.157**	0.092	1.236	**1.059**

Figure 31 shows a breakdown of the results of configurations B and C. This figure shows how the adjustments affect the values of *x*, *y*, and *z*. The *z* values are affected by the distance adjustment, while the x and y values are affected by the improvement in *z* values combined with the bounding box adjustment.

The results obtained were satisfactory, mainly for the values of x and y with RMSE results close to 1 mm. It is important to highlight that, although the simulation promotes a much more controlled environment, producing images that facilitate the work of the Mask R-CNN, the speed in the variation of values and the amplitude of these variations in the simulation is much greater than those expected in a real situation. For example, *z* values have a variation greater than 40 mm, and yaw values have a variation greater than 90º, when both should be approximately constant in a real situation, with yaw = 0º.

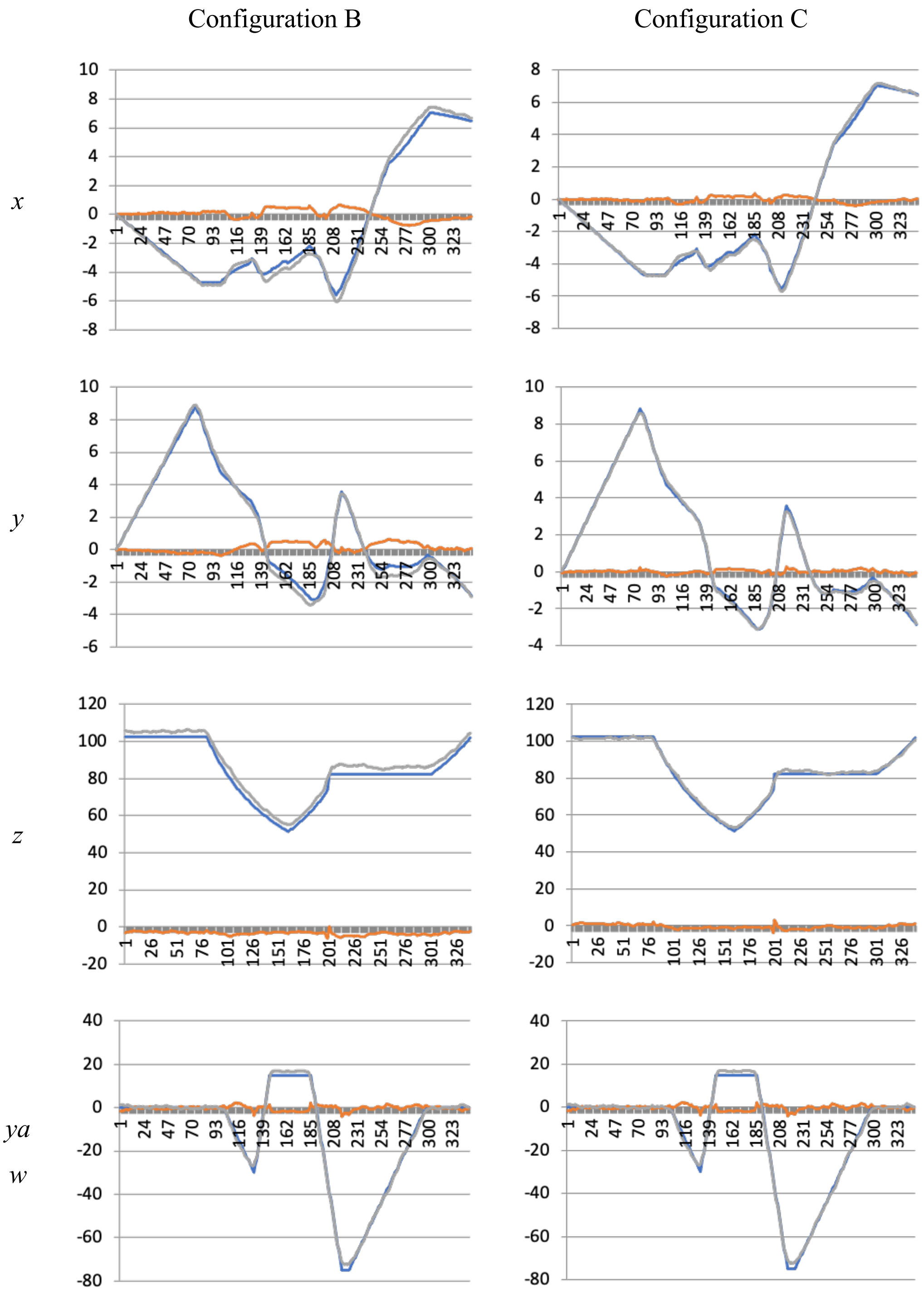


Figure 31 – Comparison between configurations B and C for each analyzed variable, with real values in blue, network output values in gray, and the difference between them in orange. *x*, *y*, and *z* values are in cm and, yaw values are in degrees.

5.2 Simulation results for foreign body location

5.2.1 Mobile device tracking

The first step in the final simulation was the creation of a new scan trajectory for the device, different from those used in the Mask R-CNN training and closer to what would occur in a clinical application, as described in subsection 4.3.2. This trajectory is performed manually by the clinician around the object in order to get as close as possible to the ideal trajectory shown in Figure 27a. With these new trajectories, the configuration C of the tracking network was performed, generating the magnetic maps with the necessary distortions, which served as input to the foreign body location network.

Figure 32 shows the difference between the *x* and *y* coordinate values between the simulation and the output of the proposed system. For this simulation, samples were generated every 1 cm. It is important to highlight that the original simulation trajectory is already different from the ideal trajectory shown in Figure 27a, and the variation of this trajectory will generate a magnetic field map with a different distortion, but not necessarily worse than that of the simulation. The experiment serves mainly to compare the results of this new trajectory, closer to the real one, with those obtained in the initial simulation, presented in section 5.1.

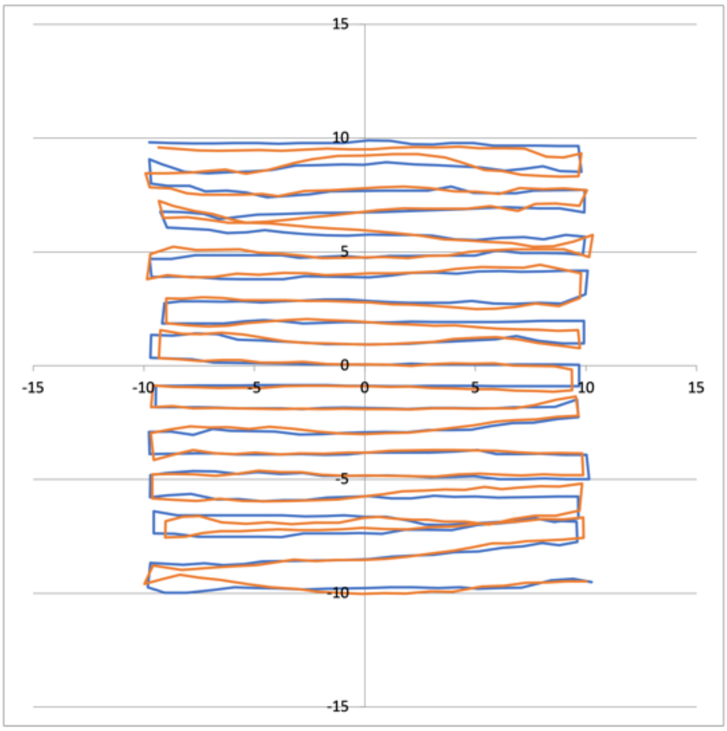
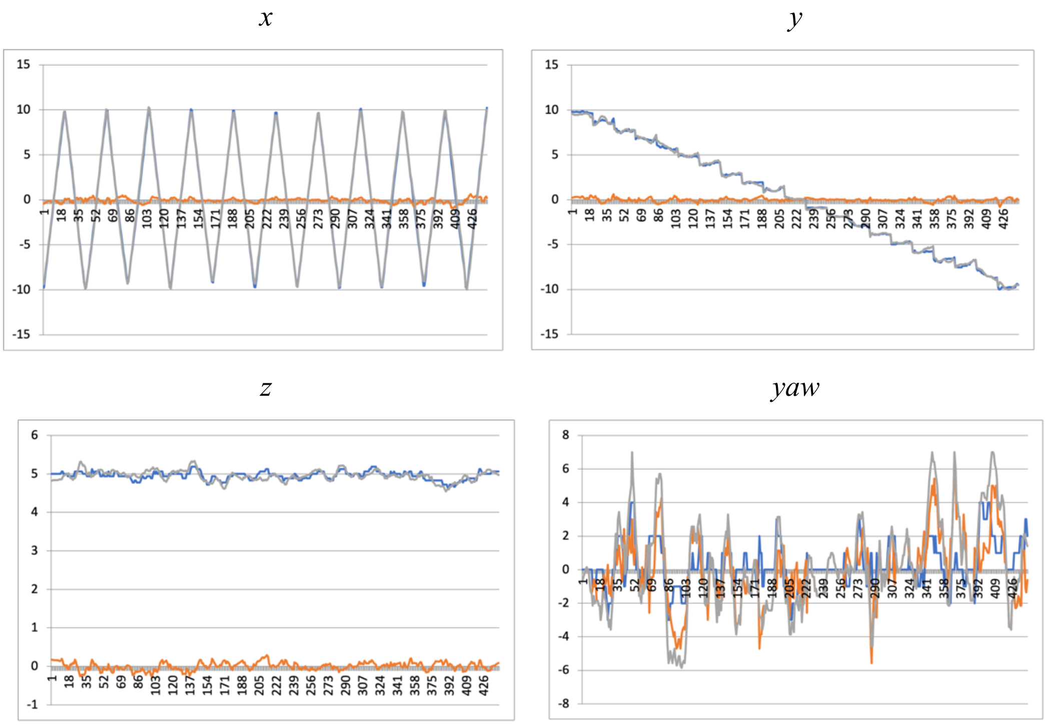


Figure 32 – Visualization of *x* and *y* positions of the device's trajectory for simulating the scan performed for foreign body location. The blue line represents the real values, and the orange line represents the output of the proposed system, with values in cm.

Figure 33 compares the real values and the system output for x, y, z, and yaw values. In this figure, system's difficulty in detecting small variations in the yaw value is evident. With values varying widely within a small margin, the filter also fails to bring the expected benefit. The values of z, on the other hand, undergo a significant improvement as they are practically constant, while x and y still show good results. Table 3 details, for all system output variables, the RMSE values (equation 24), the maximum error, and the expanded type A uncertainty, given by

$$u_A(\hat{\theta}) = 2 \times \sqrt{\frac{1}{N-2}\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2} \tag{25}$$

where θ is a scalar parameter, $\hat{\theta}$ is an estimator of the scalar parameter, and N is the number of samples.


x

y

z

yaw

Figure 33 – Comparison between real values in blue and network output values in gray for each network output value, with the difference between them indicated in orange. *x*, *y*, and *z* values are in cm, and yaw values are in degrees.

Table 3 – RMSE, Maximum Error and Expanded Type A uncertainty for Device Tracking

Output value	RMSE	Maximum Error	Expanded Type A uncertainty (95 % coverage)
x	0.255 cm	0.919 cm (@ 4.0 cm)	0.205 cm
y	0.198 cm	0.799 cm (@ -10.0 cm)	0.034 cm
z	0.102 cm	0.292 cm (@ 5.1 cm)	0.001 cm
yaw	1.963º	7.000º (@ 0º)	3.932º

Considering an Intel Dual-Core i5 2.7 GHz CPU with 8 GB of DDR3 memory, a processing time of approximately 7.5 s was obtained for each processed image, with the Mask R-CNN network being responsible for more than 99 % of this value. In the case of the simulation performed with 440 frames and 44 s duration, the total processing time would be approximately 55 min without using the buffer, an unfeasible time for a clinical application. Using the Google Collaborative configuration with a GPU, the time for each image drops to 2 s, with the total time for the simulation being close to 15 min. Even in this configuration, the processing time is still high, in addition to the increased cost that the GPU brings to the device.

Nevertheless, the use of a newer version of Tensorflow and a non-shared environment should improve the performance of the system.

5.2.2 Foreign body localization

Among the variations of the model described in subsection 4.3.3., the network with the rotation branch and absolute Δ values was used, as it presents in the test set a total MSE value (equation 21) slightly better than the network with positive and negative Δ values, mainly due to the better result of the rotation parameter. The other individual values of MSE and MAE (equation 23) are equivalent between the two networks, with the network without the rotation branch obtaining much lower values, as shown in Table 4.

Table 4 – Mean absolute error (MAE) and mean square error (MSE) values for the network with the rotation branch and absolute Δ values (A), the network with the rotation branch and real Δ values (B), and the network without the rotation branch and absolute Δ values (C).

		h (cm)	θ (º)	Φ (º)	Δ (cm)
MAE	A	2.21	1.84	4.23	0.15
	B	2.13	1.70	4.45	0.17
	C	3.86	2.62	-	0.35
MSE	A	7.78	4.82	30.73	0.05
	B	7.26	4.38	34.14	0.05
	C	22.26	10.26	-	0.28

As the selected network only presents absolute values of Δ at the output, the system must invert the sign when the rotation value is between 90º and 270º with negative inclination values and when the rotation value is between 270º and 90º with positive inclination values.

Table 5 presents the results of the selected network for the test set on RMSE values (equation 24), maximum error, and expanded type A uncertainty (equation 25) for all network output values.

Table 5 – RMSE, maximum error and expanded type A uncertainty for foreign body localization

Output value	RMSE	Maximum Error	expanded type A uncertainty (95 % coverage)
depth *h*	2.8 mm	12.6 mm (@ 130 mm)	5.6 mm
inclination θ	2.2º	7.0º (@ 60º)	4.4º
rotation ϕ	5.5º	19.9º (@ 345º)	11.1º
displacement Δ	0.2 mm	1.1 mm (@ 8.6 mm)	0.4 mm

The proposed model shows an improvement in the maximum error for Δ compared to the results obtained in [3] while providing similar results to those obtained in the initial simulation [60], Annex 1 to this dissertation, for the other variables in terms of expanded uncertainty of the type A. Figure 34 presents the scatter plots between the network output values and the true values, and Figure 35 presents the prediction error histograms for each network output value, both for the test dataset. It is important to note that, although the network has a maximum Δ error of 1.1 mm, most of the Δ error values are contained within a range between -0.5 mm and 0.5 mm.

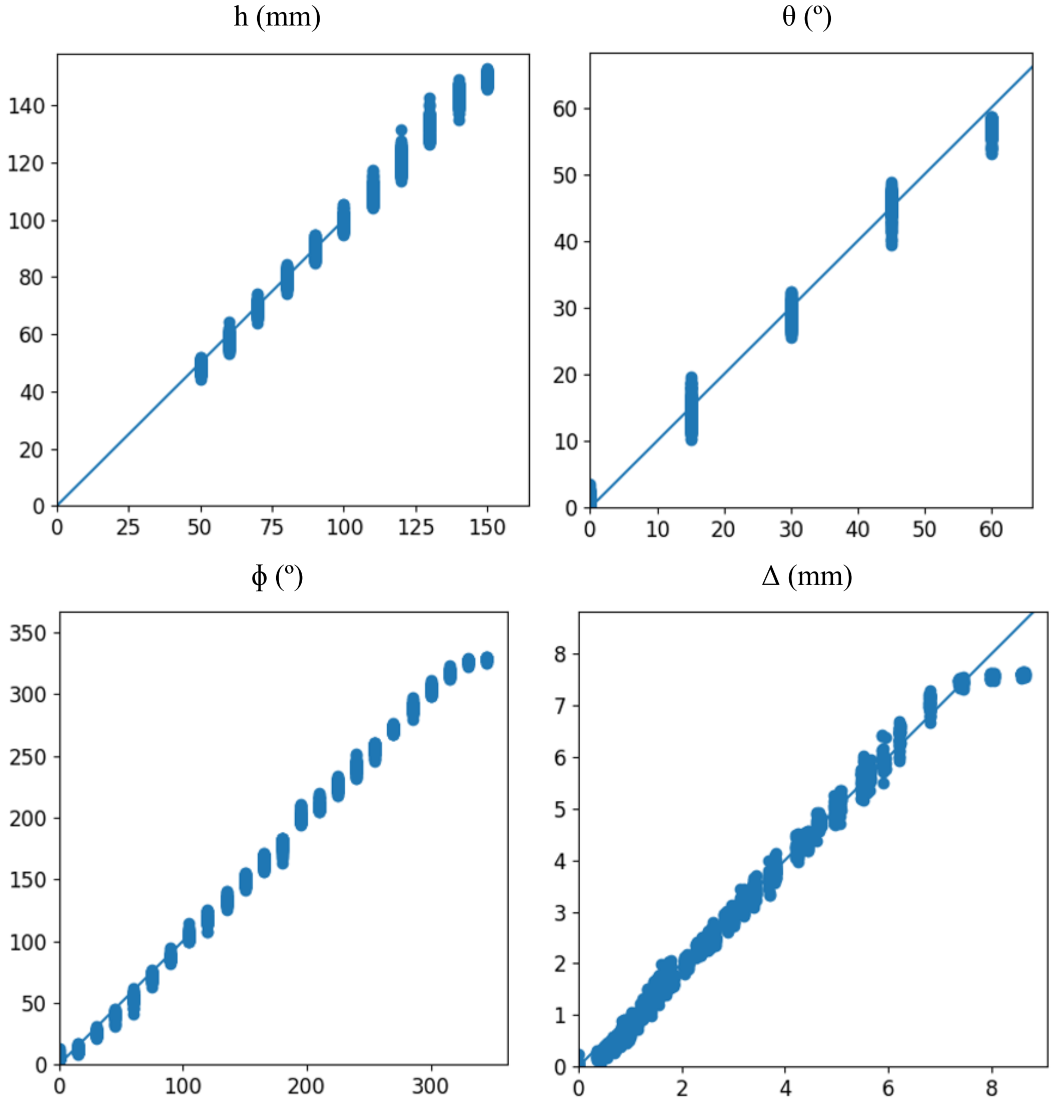


Figure 34 – Scatter plots for the test dataset between the network output values (y-axis) and the true values (x-axis) for depth (*h*), inclination (θ), rotation (ϕ), and displacement (Δ).

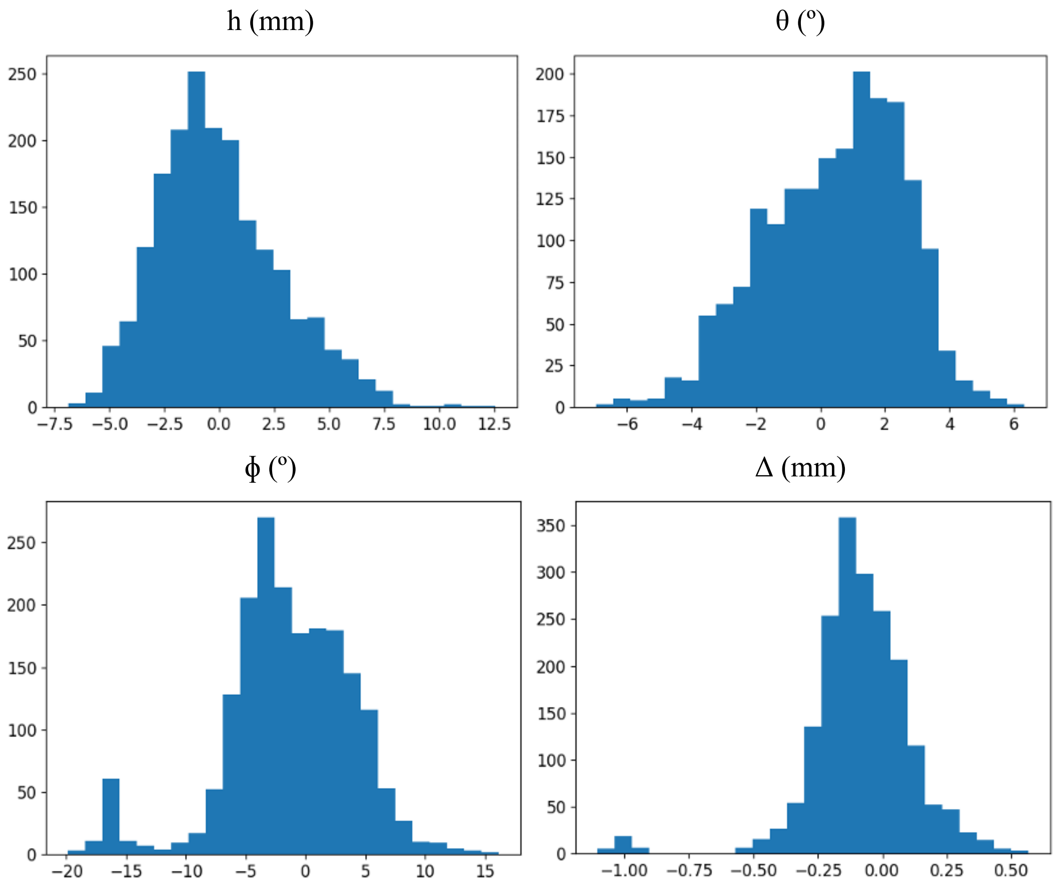


Figure 35 – Histogram for the test dataset for the network prediction error with values count (y-axis) and error values (x-axis) for depth (*h*), slope (θ), rotation (ϕ), and offset (Δ).

The network inference time was approximately 0.38 s using the same CPU configuration described in the previous subsection, which makes the methodology suitable for applications with low response time on low-cost devices.

6
Conclusions and Future Works

6.1
Conclusions

This research aimed to evaluate the application of recent advances in artificial intelligence in proposing a low-cost portable device for locating foreign bodies accidentally inserted into the human body, based on the magnetic field generated by the foreign body. For this purpose, a GMR sensor and a monocular camera commonly found in electronic devices were considered, a configuration costing about 10 000 times less than the high-precision SQUID devices currently found on the market. In addition to the cost, the proposed device brings other benefits such as lightness, portability, safety, non-invasiveness, being able to work at room temperature, in addition to bringing the possibility of remote care or mobile ICU care.

The research was divided into two main phases, one that seeks to track the position and orientation of the device in relation to a reference point on the patient's body based on a camera integrated into the device, and another that seeks to discover the position, rotation and inclination of the foreign body based on the data collected in the previous step together with the GMR sensor data.

In chapter 2, the basic concepts for the implementation of the methods proposed for the two phases were presented, and these methods are presented in chapter 3. In section 3.1, the system based on the Mask R-CNN network for device tracking and, in section 3.2, the convolutional neural network that serves as the basis for the localization of the foreign body was detailed. In chapter 4, the training processes of the proposed systems were detailed, and in chapter 5, the results of the simulations were presented.

The results obtained in tracking the device were quite satisfactory, especially for the positioning values in relation to the reference point on the patient, with RMSE results ranging between 1 mm and 2.6 mm for x, y, and z values. These results are of particular importance, as knowing the relative position between the device and the patient is fundamental for the inverse problem algorithm, and, as the device is portable, there is no fixed reference available for this measurement. A possible alternative would be to employ inertial navigation from triaxial accelerometer signals, but the typical drift of such sensors would significantly reduce the quality of the position estimate, in addition to the difficulty in defining an initial reference point on the patient skin. In the case of the device orientation angles, although it is desirable that the image-based system would also produce reliable results, there are low-cost gyrometers on the market that can fulfill this role more robustly or be used in conjunction with the camera to produce better results. The processing time of the network, however, needs to improve so that it can be used in a clinical situation. More modern and efficient convolutional networks or a simulation with fewer samples per second can help mitigate the problem.

The CNN results for foreign body localization are also positive. The value of Δ, used to find the center of the needle, with an RMSE of 0.2 mm, presents a significant improvement over previous works [3] and, combined with a depth value with an RMSE of 2.8 mm, indicates a good result in locating the foreign body in the patient. The foreign body inclination value also had good results, with an RMSE of 2.2° and a maximum error of 7.0°, while the foreign body rotation value, which had the worst results with an RMSE of 5.5°, can be inferred from the magnetic field data, possibly yielding more reliable results. Another positive factor in using the CNN for foreign body localization is that it showed generalizability by producing results from skewed tracking data that was not in its training dataset. This quality indicates that errors in the device tracking process can be smoothed over by the location network, creating a relationship between the two steps distinct from a simple overlap of errors.

6.2 Future works

Several improvements can be proposed for the methods presented in this research. As the Deep Learning area is recent and undergoes constant changes, new methods of object detection and semantic segmentation appear every year with better results and faster response times. These methodologies can not only increase the accuracy of the results and the response time but also dismiss the adjustments that were created in this research to compensate for the errors presented by the Mask R-CNN network. As an example, we can mention the PANet network [61], considered an evolution of the Mask R-CNN network; the SiamMask [62], which promises a speed of 55 fps and has as one of its outputs the bounding box rotated towards the object, which could maybe be used by the system as an initial yaw value; and the EfficientDet [63] network, which has obtained good results in detecting objects with low use of computational resources and can be used alone or as a backbone of the Mask R-CNN network. A pure semantic segmentation approach is also feasible, with the position of the pattern being obtained by the bounding box of the mask found by the network. The UNet network [64] is a good starting point of experimentation. Finally, other techniques like Template Matching and Watershed should also be tested for accuracy and speed analysis.

Another important point to be analyzed is the possibility of integrating the tracking system with data obtained by physical sensors, mainly for device orientation. A distance sensor can also bring greater reliability to the results in combination with the *z* value obtained by the system.

For the foreign body localization CNN, it would be interesting to perform a reference simulation with perfect values of the magnetic field and test different distortions of this field from different trajectories in order to try to assess how much the network can maintain the results regardless of the generated distortions.

Finally, it is essential to implement the system on the Raspberry PI platform to perform the integration with the GMR sensor to validate the results in a real situation. The processing time issue found in this research could be solved with one of the solutions proposed in this section or by implementing a Bluetooth connection between the Raspberry Pi device and an external machine with GPU capabilities. Unfortunately, due to the COVID-19 pandemic, laboratory work was impaired during the period of this research, limiting the results to the simulations presented.

Bibliography

[1] BIRISIK, Y. B. F. Retrospective analysis of puncture wounds with retained foreign bodies in the extremities requiring surgical intervention -. **Medicine Science | International Medical Journal**, v. 10, n. 2, p. 416–21, 2021.

[2] BARBOSA, C. Localization of firearm projectiles in the human body using a superconducting quantum interference device magnetometer: A theoretical study. **Review of Scientific Instruments**, v. 75, p. 2098–2106, 24 may 2004.

[3] BARBOSA, C. H.; COSTA MONTEIRO, E.; LIMA, E. A.; SANTOS, S. F.; CAVALCANTI, E. G.; RIBEIRO, P. C. Improvement of a technique for localization of steel needles in humans using a SQUID magnetometer. **IEEE Transactions on Applied Superconductivity**, v. 11, n. 1, p. 677–680, mar. 2001.

[4] MONTEIRO, E.; BARBOSA, C.; LIMA, E.; RIBEIRO, P.; BOECHAT, P. Locating steel needles in the human body using a SQUID magnetometer. **Physics in medicine and biology**, v. 45, p. 2389–402, sep. 2000.

[5] ARAS, M. H.; MILOGLU, O.; BARUTCUGIL, C.; KANTARCI, M.; OZCAN, E.; HARORLI, A. Comparison of the sensitivity for detecting foreign bodies among conventional plain radiography, computed tomography and ultrasonography. **Dentomaxillofacial Radiology**, v. 39, n. 2, p. 72–78, feb. 2010.

[6] OLIVEIRA, B. R.; LOUZADA, D. R.; COSTA MONTEIRO, E.; BARBOSA, C. R. H. Automatic system for locating magnetic foreign bodies using GMI magnetometer. **Journal of Physics**, Conference Series (PRINT). 2020.

[7] GINSBURG, M. J.; ELLIS, G. L.; FLOM, L. L. Detection of soft-tissue foreign bodies by plain radiography, xerography, computed tomography, and ultrasonography. **Annals of Emergency Medicine**, v. 19, n. 6, p. 701–703, jun. 1990.

[8] MCQUIRTER, J. L.; ROTHENBERG, S. J.; DINKINS, G. A.; MANALO, M.; KONDRASHOV, V.; TODD, A. C. The Effects of Retained Lead Bullets on Body Lead Burden. **Journal of Trauma and Acute Care Surgery**, v. 50, n. 5, p. 892–899, may 2001.

[9] VENTER, N. G.; JAMEL, N.; MARQUES, R. G.; DJAHJAH, F.; MENDONÇA, L. DE S. Avaliação de métodos radiológicos na detecção de corpo estranho de madeira em modelo animal. **Acta Cirúrgica Brasileira**, v. 20, p. 19–26, 2005.

[10] MARALAKUNTE, M.; DEBI, U.; SINGH, L.; PRUTHI, H.; BHATIA, V.; DEVI, G.; MS, S. Foreign Body Imaging-Experience with 6 Cases of Retained Foreign Bodies in the Emergency Radiology Unit. **Archives of Clinical and Medical Case Reports**, v. 4, n. 5, p. 952–968, 21 sep. 2020.

[11] MONTEIRO, E. C. Biometrologia: confiabilidade nas biomedições e repercussões éticas. **Metrologia e Instrumentação**, v. 6, p. 6–12, 2007.

[12] COSTA MONTEIRO, E.; LEON, L. F. Metrological Reliability of Medical Devices. **Journal of Physics: Conference Series**, v. 588, p. 012032, feb. 2015.

[13] BARBOSA, C.; MONTEIRO, E.; POMPÉIA, F. **Localization of Magnetic Foreign Bodies in Humans using Magnetic Field Sensors**. XVII IMEKO World Congress. **Proceedings**...Dubrovnik: 1 jun. 2003

[14] FORTALEZA, L. G. E S.; BARBOSA, C. R. H.; COSTA MONTEIRO, E.; SILVA, E. C. DA; GUSMÃO, L. A. P. DE. **Detecting non-magnetic metallic foreign bodies by GMR sensors through the use of eddy currents**. XI International Congress on Electrical Metrology (SEMETRO). **Proceedings**...2015Available: <http://localhost:8080/jspui/handle/2050011876/1012>. Accessed: 26 oct. 2020

[15] FORTALEZA, L. G. S.; COSTA MONTEIRO, E.; BARBOSA, C. R. H.; SILVA, E. C.; GUSMÃO, L. A. P. Biomedical comparison of magnetometers for non-ferromagnetic metallic foreign body detection. **Journal of Physics: Conference Series**, v. 1044, p. 012013, jun. 2018.

[16] JING, D.; LUO, E.; SHEN, G.; CAI, J.; TANG, C.; YAN, Y.; JING, B. **Fast method of locating metallic foreign body in the human body**. 2009 9th International Conference on Electronic Measurement Instruments. **Proceedings**... In: 2009 9TH INTERNATIONAL CONFERENCE ON ELECTRONIC MEASUREMENT INSTRUMENTS. aug. 2009

[17] MEDINA, J. P.; LOUZADA, D. R.; MONTEIRO, E. C.; BARBOSA, C. R. H.; PINTO, R. D. DE O. **Resolução do problema inverso por Redes Neurais Artificiais para localização de fonte de campo magnético**. IX Congresso Brasileiro de Metrologia. **Proceedings**...2017Available: <http://localhost:8080/jspui/handle/2050011876/474>. Accessed: 3 dec. 2020

[18] POMPÉIA, F.; GUSMÃO, L. A. P.; BARBOSA, C. R. H.; COSTA MONTEIRO, E.; GONÇALVES, L. A. P.; MACHADO, F. L. A. Ring shaped magnetic field transducer based on the GMI effect. **Measurement Science and Technology**, v. 19, n. 2, p. 025801, jan. 2008.

[19] VELÁZQUEZ, D. R. T.; MONTEIRO, E. C.; LOUZADA, D. R.; BARBOSA, C. R. H. Multiparametric quality by design-fuzzy model applied in the development of a biomedical measuring system. **International Journal of Metrology and Quality Engineering**, v. 11, p. 12, 2020.

[20] FORTALEZA, L. G. S.; SILVA, E. C.; BARBOSA, C. R. H.; GUSMÃO, L. A. P.; MONTEIRO, E. C. Electronic circuit for excitation of inductive loads with high currents. **Electronics Letters**, v. 51, n. 22, p. 1808–1809, 2015.

[21] MONTEIRO, E. C. Magnetic Quantities: Healthcare Sector Measuring Demands and International Infrastructure for providing Metrological Traceability. In: **TMQ – Techniques, Methodologies and Quality**. [s.l: s.n.]. v. 1p. 42–50.

[22] SILVA, E. C.; GUSMÃO, L. A. P.; BARBOSA, C. R. H.; MONTEIRO, E. C. An electronic approach to homogenize the impedance phase characteristics of heterogeneous GMI sensors. **ACTA IMEKO**, v. 1, n. 1, p. 70–76, 10 jul. 2012.

[23] SILVA, E. C.; GUSMÃO, L. A. P.; BARBOSA, C. R. H.; MONTEIRO, E. C.; MACHADO, F. L. A. High sensitivity giant magnetoimpedance (GMI) magnetic transducer: magnitude versus phase sensing. **Measurement Science and Technology**, v. 22, n. 3, p. 035204, feb. 2011.

[24] SILVA, E. C.; GUSMAO, L. A. P.; BARBOSA, C. R. H.; MONTEIRO, E. C. Electronic approach for enhancing impedance phase sensitivity of GMI magnetic sensors. **Electronics Letters**, v. 49, n. 6, p. 396–397, mar. 2013.

[25] SILVA, E. C.; BARBOSA, C. R. H.; GUSMÃO, L. A. P.; LEIPNER, Y.; FORTALEZA, L. G. S.; MONTEIRO, E. C. Point matching: a new

electronic method for homogenizing the phase characteristics of giant magnetoimpedance sensors. **The Review of Scientific Instruments**, v. 85, n. 8, p. 084708, aug. 2014.

[26] SILVA, E. C.; GUSMÃO, L. A. P.; BARBOSA, C. R. H.; MONTEIRO, E. C. An enhanced electronic topology aimed at improving the phase sensitivity of GMI sensors. **Measurement Science and Technology**, v. 25, n. 11, p. 115010, oct. 2014.

[27] **Qualcomm Tricorder XPRIZE | XPRIZE Foundation**. Available: <https://www.xprize.org/prizes/tricorder>. Accessed: 27 jun. 2021.

[28] **Basil Leaf Technologies**. Available: <http://www.basilleaftech.com>. Accessed: 1 dec. 2020.

[29] BRACHMANN, E.; KRULL, A.; NOWOZIN, S.; SHOTTON, J.; MICHEL, F.; GUMHOLD, S.; ROTHER, C. **DSAC — Differentiable RANSAC for Camera Localization**. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). **Proceedings**... In: 2017 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). jul. 2017

[30] BRACHMANN, E.; ROTHER, C. **Learning Less is More - 6D Camera Localization via 3D Surface Regression**. 1 jun. 2018

[31] MENG, L.; CHEN, J.; TUNG, F.; LITTLE, J. J.; VALENTIN, J.; SILVA, C. **Backtracking regression forests for accurate camera relocalization**. 1 sep. 2017

[32] SATTLER, T.; LEIBE, B.; KOBBELT, L. Efficient Effective Prioritized Matching for Large-Scale Image-Based Localization. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 9, p. 1744–1756, sep. 2017.

[33] TAIRA, H.; OKUTOMI, M.; SATTLER, T.; CIMPOI, M.; POLLEFEYS, M.; SIVIC, J.; PAJDLA, T.; TORII, A. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, p. 1–1, 2019.

[34] KENDALL, A.; CIPOLLA, R. **Modelling uncertainty in deep learning for camera relocalization**. 2016 IEEE International Conference on Robotics and Automation (ICRA). **Proceedings**... In: 2016 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). may 2016

[35] KENDALL, A.; CIPOLLA, R. **Geometric Loss Functions for Camera Pose Regression with Deep Learning**. 1 jul. 2017

[36] KENDALL, A.; GRIMES, M.; CIPOLLA, R. **PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization**. 2015 IEEE International Conference on Computer Vision (ICCV). **Proceedings**... In: 2015 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV). dec. 2015

[37] NASEER, T.; BURGARD, W. **Deep regression for monocular camera-based 6-DoF global localization in outdoor environments**. 1 sep. 2017

[38] RADWAN, N.; VALADA, A.; BURGARD, W. VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry. **IEEE Robotics and Automation Letters**, v. 3, n. 4, p. 4407–4414, oct. 2018.

[39] VALADA, A.; RADWAN, N.; BURGARD, W. **Deep Auxiliary Learning for Visual Localization and Odometry**. 15 may 2018

[40] WALCH, F.; HAZIRBAS, C.; LEAL-TAIXÉ, L.; SATTLER, T.; HILSENBECK, S.; CREMERS, D. **Image-Based Localization Using**

LSTMs for Structured Feature Correlation. 2017 IEEE International Conference on Computer Vision (ICCV). **Proceedings**... In: 2017 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV). oct. 2017

[41] WU, J.; MA, L.; HU, X. **Delving deeper into convolutional neural networks for camera relocalization**. 2017 IEEE International Conference on Robotics and Automation (ICRA). **Proceedings**... In: 2017 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA). may 2017

[42] SATTLER, T.; ZHOU, Q.; POLLEFEYS, M.; LEAL-TAIXE, L. **Understanding the limitations of cnn-based absolute camera pose regression**. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. **Proceedings**...2019Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85076638739&doi=10.1109%2fCVPR.2019.00342&partnerID=40&md5=6d8e2773e52006cb31eaef813513e883>

[43] STINCHCOMBE; WHITE. **Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions**. International 1989 Joint Conference on Neural Networks. **Proceedings**... In: INTERNATIONAL 1989 JOINT CONFERENCE ON NEURAL NETWORKS. 1989

[44] NEWELL, A. Perceptrons. An Introduction to Computational Geometry. Marvin Minsky and Seymour Papert. M.I.T. Press, Cambridge, Mass., 1969. vi + 258 pp., illus. Cloth, $12; paper, $4.95. **Science**, v. 165, n. 3895, p. 780–782, 22 aug. 1969.

[45] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, 1 jan. 1989.

[46] SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, v. 15, n. 56, p. 1929–1958, 2014.

[47] HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. **Mask R-CNN**. 2017 IEEE International Conference on Computer Vision (ICCV). **Proceedings**... In: 2017 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV). oct. 2017

[48] GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. **Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation**. 2014 IEEE Conference on Computer Vision and Pattern Recognition. **Proceedings**... In: 2014 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. jun. 2014

[49] GIRSHICK, R. **Fast R-CNN**. 2015 IEEE International Conference on Computer Vision (ICCV). **Proceedings**... In: 2015 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV). dec. 2015

[50] REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 6, p. 1137–1149, jun. 2017.

[51] SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. **arXiv:1409.1556 [cs]**, 10 apr. 2015.

[52] HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. **arXiv:1512.03385 [cs]**, 10 dec. 2015.

[53] **Smoothing and Differentiation of Data by Simplified Least Squares Procedures. | Analytical Chemistry**. Available: <https://pubs.acs.org/doi/abs/10.1021/ac60214a047>. Accessed: 7 aug. 2021.

[54] BISONG, E. Google Colaboratory. In: BISONG, E. (Ed.). . **Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners**. Berkeley, CA: Apress, 2019. p. 59–64.

[55] VAN ROSSUM, G.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009.

[56] ABADI, M. et al. **Tensorflow: A system for large-scale machine learning**. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). **Proceedings**...2016

[57] DUTTA, A.; ZISSERMAN, A. **The VIA Annotation Software for Images, Audio and Video**. Proceedings of the 27th ACM International Conference on Multimedia. **Proceedings**...: MM '19.New York, NY, USA: ACM, 2019Available: <https://doi.org/10.1145/3343031.3350535>

[58] ABDULLA, W. **Mask R-CNN for object detection and instance segmentation on Keras and TensorFlowGitHub repository**Github, , 2017. Available: <https://github.com/matterport/Mask_RCNN>

[59] LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. **Microsoft COCO: Common Objects in Context**. (D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, Eds.)Computer Vision – ECCV 2014. **Proceedings**...: Lecture Notes in Computer Science.Cham: Springer International Publishing, 2014

[60] ROGOZINSKI, M.; BARBOSA, C. R. H.; FEITOSA, R. Q.; OLIVEIRA, B. R. C.; MONTEIRO, E. C.; LOUZADA, D. R. Localization of magnetic foreign bodies using CNN and GMI magnetometer. **Measurement: Sensors**, v. 18, p. 100133, 2021.

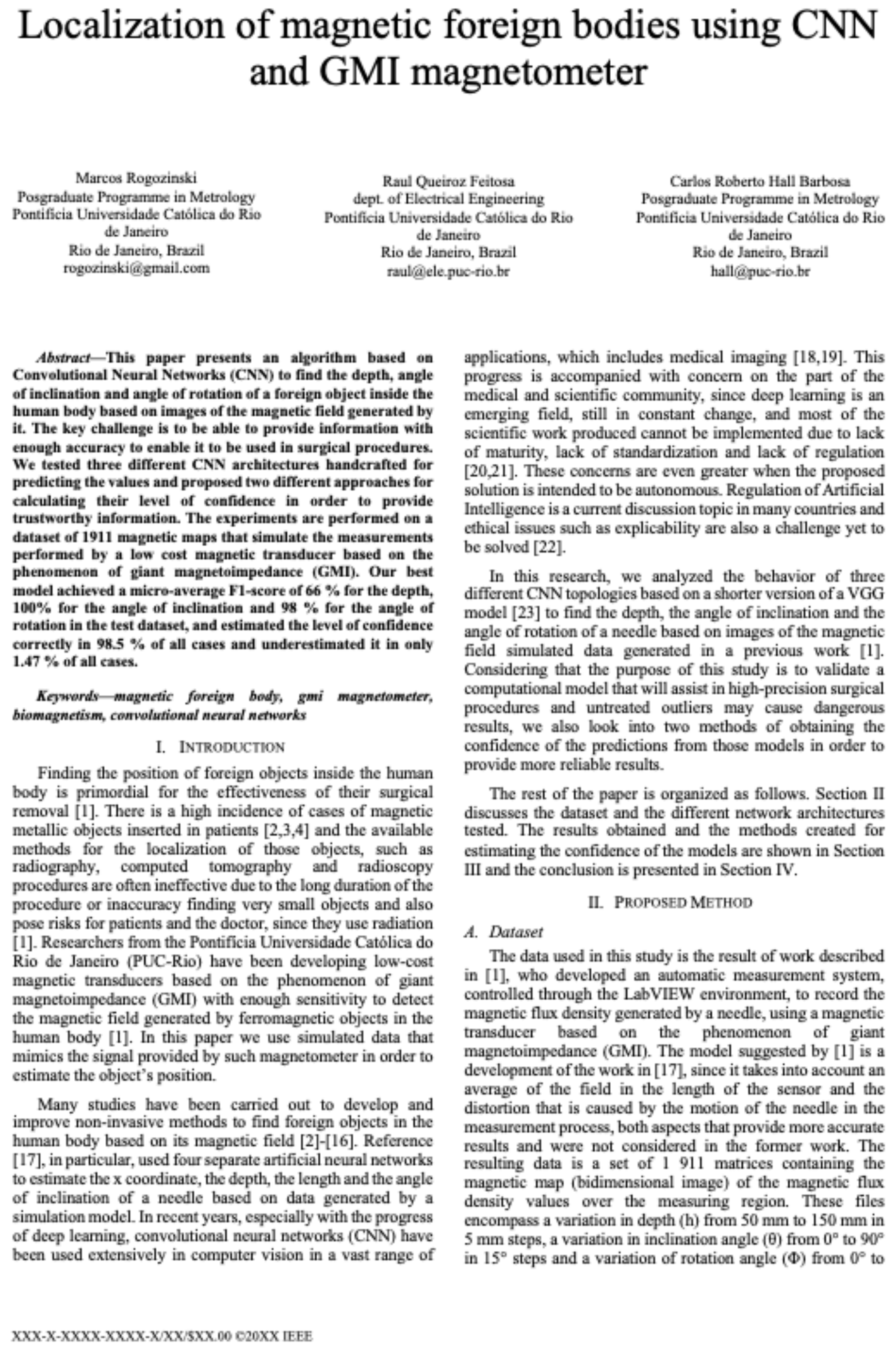
[61] LIU, S.; QI, L.; QIN, H.; SHI, J.; JIA, J. **Path Aggregation Network for Instance Segmentation**. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. **Proceedings**... In: 2018 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. jun. 2018

[62] WANG, Q.; ZHANG, L.; BERTINETTO, L.; HU, W.; TORR, P. **Fast Online Object Tracking and Segmentation: A Unifying Approach**. 1 jun. 2019

[63] TAN, M.; PANG, R.; LE, Q. V. **EfficientDet: Scalable and Efficient Object Detection**. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). **Proceedings**... In: 2020 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). jun. 2020

[64] RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. **arXiv:1505.04597 [cs]**, 18 may 2015.

Appendix A: Localization of magnetic foreign bodies using CNN and GMI magnetometer



Localization of magnetic foreign bodies using CNN and GMI magnetometer

Marcos Rogozinski
Posgraduate Programme in Metrology
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brazil
rogozinski@gmail.com

Raul Queiroz Feitosa
dept. of Electrical Engineering
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brazil
raul@ele.puc-rio.br

Carlos Roberto Hall Barbosa
Posgraduate Programme in Metrology
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, Brazil
hall@puc-rio.br

***Abstract*—This paper presents an algorithm based on Convolutional Neural Networks (CNN) to find the depth, angle of inclination and angle of rotation of a foreign object inside the human body based on images of the magnetic field generated by it. The key challenge is to be able to provide information with enough accuracy to enable it to be used in surgical procedures. We tested three different CNN architectures handcrafted for predicting the values and proposed two different approaches for calculating their level of confidence in order to provide trustworthy information. The experiments are performed on a dataset of 1911 magnetic maps that simulate the measurements performed by a low cost magnetic transducer based on the phenomenon of giant magnetoimpedance (GMI). Our best model achieved a micro-average F1-score of 66 % for the depth, 100% for the angle of inclination and 98 % for the angle of rotation in the test dataset, and estimated the level of confidence correctly in 98.5 % of all cases and underestimated it in only 1.47 % of all cases.**

***Keywords*—magnetic foreign body, gmi magnetometer, biomagnetism, convolutional neural networks**

I. Introduction

Finding the position of foreign objects inside the human body is primordial for the effectiveness of their surgical removal [1]. There is a high incidence of cases of magnetic metallic objects inserted in patients [2,3,4] and the available methods for the localization of those objects, such as radiography, computed tomography and radioscopy procedures are often ineffective due to the long duration of the procedure or inaccuracy finding very small objects and also pose risks for patients and the doctor, since they use radiation [1]. Researchers from the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) have been developing low-cost magnetic transducers based on the phenomenon of giant magnetoimpedance (GMI) with enough sensitivity to detect the magnetic field generated by ferromagnetic objects in the human body [1]. In this paper we use simulated data that mimics the signal provided by such magnetometer in order to estimate the object's position.

Many studies have been carried out to develop and improve non-invasive methods to find foreign objects in the human body based on its magnetic field [2]-[16]. Reference [17], in particular, used four separate artificial neural networks to estimate the x coordinate, the depth, the length and the angle of inclination of a needle based on data generated by a simulation model. In recent years, especially with the progress of deep learning, convolutional neural networks (CNN) have been used extensively in computer vision in a vast range of applications, which includes medical imaging [18,19]. This progress is accompanied with concern on the part of the medical and scientific community, since deep learning is an emerging field, still in constant change, and most of the scientific work produced cannot be implemented due to lack of maturity, lack of standardization and lack of regulation [20,21]. These concerns are even greater when the proposed solution is intended to be autonomous. Regulation of Artificial Intelligence is a current discussion topic in many countries and ethical issues such as explicability are also a challenge yet to be solved [22].

In this research, we analyzed the behavior of three different CNN topologies based on a shorter version of a VGG model [23] to find the depth, the angle of inclination and the angle of rotation of a needle based on images of the magnetic field simulated data generated in a previous work [1]. Considering that the purpose of this study is to validate a computational model that will assist in high-precision surgical procedures and untreated outliers may cause dangerous results, we also look into two methods of obtaining the confidence of the predictions from those models in order to provide more reliable results.

The rest of the paper is organized as follows. Section II discusses the dataset and the different network architectures tested. The results obtained and the methods created for estimating the confidence of the models are shown in Section III and the conclusion is presented in Section IV.

II. Proposed Method

A. Dataset

The data used in this study is the result of work described in [1], who developed an automatic measurement system, controlled through the LabVIEW environment, to record the magnetic flux density generated by a needle, using a magnetic transducer based on the phenomenon of giant magnetoimpedance (GMI). The model suggested by [1] is a development of the work in [17], since it takes into account an average of the field in the length of the sensor and the distortion that is caused by the motion of the needle in the measurement process, both aspects that provide more accurate results and were not considered in the former work. The resulting data is a set of 1 911 matrices containing the magnetic map (bidimensional image) of the magnetic flux density values over the measuring region. These files encompass a variation in depth (h) from 50 mm to 150 mm in 5 mm steps, a variation in inclination angle (θ) from 0° to 90° in 15° steps and a variation of rotation angle (Φ) from 0° to

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

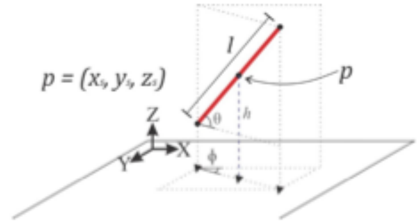


Fig. 1. Visualization of a needle position (in red) with *p* as its center point, showing the needle length (*l*), inclination angle (θ), rotation angle (Φ) and depth (*h*) [17].

180° in 15° steps. Fig. 1 shows the different parameters used in this study. We had to eliminate 252 files from the dataset, since rotation of the needle with an inclination angle of 90° in any given depth always produce the same magnetic flux density values. That issue reduced our dataset to 1 659 files and left us with very few examples for that inclination angle.

To allow for a broader and less biased range of results, some steps were taken to augment the data. The first step, done in MATLAB, was to invert the magnetic flux values to generate maps similar to the original ones but with the magnetic poles of the needle inverted (e.g., a data with parameters h = 55 mm, θ = 60° and Φ = 0° would become h = 55 mm, θ = -60° and Φ = 180°). Another modification done in MATLAB was the introduction of Gaussian noise with randomly selected standard deviations in the magnetic flux values in order to make our CNNs more robust and simulate signal degradation compatible with those found in commercial equipment. We created a dataset with low noise using random standard deviation values between 0.001 and 0.002 and a high noise dataset using standard deviation values between 0.002 and 0.003. Finally, we rotated the generated images to achieve a full range of inclination angles between -90° and 90° and rotation angles between 0° and 360°. By combining all files, we ended up with 16 758 images in our dataset. Fig. 2 illustrates the process of data augmentation from our original files to the final dataset and Fig. 3 show examples of generated images varying in depth, inclination, rotation and noise by their smallest step value. It is important to note the subtlety in the depth changes, which are hard to notice by the human eye and will also be harder for the CNNs to predict precisely, especially considering images with the presence of noise.

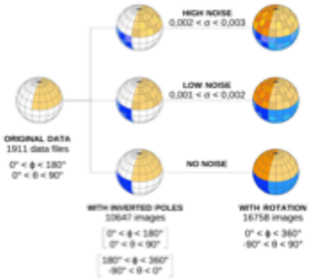


Fig. 2. The process of data augmentation from the original files to the final dataset. Magnetic flux values are inverted to simulate pole invertion and Gaussian noise with random standard deviation is added to simulate signal degradation. Finally, images are rotate to achive all possible positions.

B. Convolutional Neural Networks Models

Although the challenge of finding numeric values with a neural network is primarily a regression problem, we also wanted to treat it as a classification problem so that we could analyze the vector of probabilities generated by that kind of network to suggest the confidence of the network's response. We also compared two classification architectures, one that shares all the layers to produce a single vector of probabilities and one that shares only part of the layers and outputs one vector of probabilities for each task. In the model that outputs a single vector of probabilities containing all the classes (Model A), all layers of the network are shared between the tasks, which should provide for a good generalization of the network [24,25]. After training, the prediction vector is separated into three smaller vectors, one for each value we want to predict (depth, angle of inclination and angle of rotation) so we can more easily analyze the results and compare with the other models. In Model B, the first layers are shared, and the last layers are specific for each task. This approach should also provide for good generalization while keeping most part of the learnable parameters specific for each task. This model returns three vectors of probabilities, one for each task, with all the probabilities in a vector summing to one. The last network architecture tested (Model C) is the same as Model B, except for the last layer, which outputs a single real number for each task.

We tested the networks with different number of layers to find the right size for the amount of data available. If more data becomes available in the future or the number of tasks increases, such as finding also the size of the needle, we could possibly add more layers to the network to obtain similar or better results. The parameters were kept mainly equal between the models, with the exception of Model A using a learning rate of 10^{-4} while Model B and C used a learning rate of 10^{-3}. Initially, Model A had a bad result in predicting depth values, so a single task model with the same architecture was trained only for the depth values and the weights from that network after training were transferred to Model A to be used as the initial weights for the network. This procedure allowed for a 10% to 15% improvement for all metrics in the prediction of the depth values while not compromising the other tasks. Fig. 4 shows the network architecture for the final models with detailed information about the parameters and functions used.

It is important to note that Model A uses a sigmoid activation function in the last layer with a binary crossentropy loss function in order to produce a single output vector with

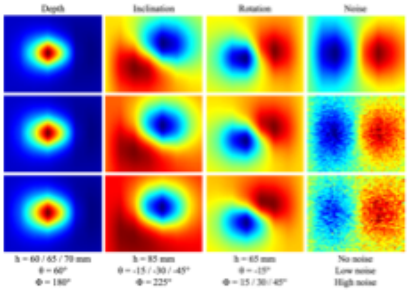


Fig. 3. Examples of generated images varying in depth, inclination, rotation and noise by their smallest step value.

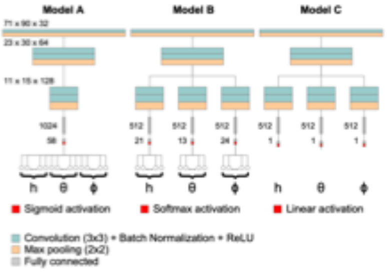


Fig. 4. Network architecture for the final models. Model A outputs a single vector with independent probabilities, Model B outputs one vector with probabilities summing to one for each task and Model C outputs a real number for each task.

independent probabilities for each class; Model B uses a softmax activation function in the last layer with a categorical crossentropy loss function in order to produce vectors of probabilities that sum to one for each task; and Model C uses a linear activation function in the last layer with a mean square error loss function in order to produce real numbers as outputs for each task. Another important distinction between the networks is the difference in the number of parameters that have to be trained: 4 928 634 for Model A, 7 638 138 for Model B and 1 793 763 for Model C. Since most of the parameters in a CNN are concentrated in the fully connected layers in the end of the network, Model C has much less parameters since its last layer has only one neuron, while Model B has fewer parameters than Model A since its layers are shared between all tasks.

III. Results

A. Convolutional Neural Networks Results

To test the performance of the proposed models, we used 60% of our images for training, 20% for validation and 20% for testing. This division was done in a way to keep the same number of images per class in each dataset, which is especially important in the case of the images with 90° of rotation, where we had fewer samples and needed to ensure a good representation in all phases of the process. We trained our data in mini batches of 64 samples and limited our training to 40 epochs but allowed it to terminate earlier if the validation accuracy did not improve for 5 epochs.

TABLE I. Results from the Test Data

	Depth	*Inclination*	*Rotation*
	micro-average / macro-average		
Model A			
Precision	0.48 / 0.47	0.99 / 0.99	0.91 / 0.91
Recall	0.48 / 0.48	0.99 / 0.87	0.91 / 0.91
F1-score	0.48 / 0.47	0.99 / 0.89	0.91 / 0.91
Model B			
Precision	0.67 / 0.67	1.00 / 1.00	0.98 / 0.98
Recall	0.66 / 0.67	1.00 / 0.99	0.98 / 0.98
F1-score	0.66 / 0.67	1.00 / 1.00	0.98 / 0.98
Model C			
Precision	0.53 / 0.52	0.99 / 0.84	0.87 / 0.87
Recall	0.53 / 0.53	0.99 / 0.85	0.87 / 0.87
F1-score	0.53 / 0.52	0.99 / 0.84	0.87 / 0.87

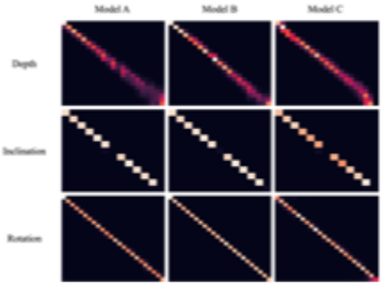


Fig. 5. Heat map of the confusion matrix for each task in each model for the test data. The depth predictions are scattered but not too far from the true value, and they lose precision as the depth values gets higher. In the rotation graphs, the "wholes" in the 90° and -90° values due to lack of samples compared to the other rotation values.

The results obtained with the test data are shown in Table 1 and evidence a greater difficulty of the network in predicting depth values, while obtaining very good results in the other two tasks. The precision metric is the ratio of correctly predicted positive observations to the total predicted positive observations, the recall metric is the ratio of correctly predicted positive observations to the all observations in the actual class and the F1 score is the weighted average of Precision and Recall [26]. The macro-average method computes the metric independently for each class and then take its average while the micro-average method will aggregate the contributions of all classes to compute the average metric, which will take into account any class imbalance. Micro-average and macro-average values are very similar, showing a good balance of prediction values for all the classes.

Another view of the results is shown in Fig. 5, where we plotted a heat map of the confusion matrix for each task in each model for the test data. From these maps we can better see that the depth predictions are scattered but not too far from the true value, and they lose precision as the depth values gets higher. In the rotation graphs we can see the "holes" in the 90°

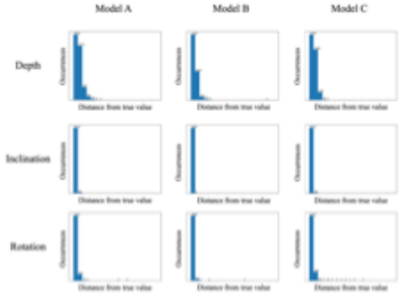


Fig. 6. Plot of the distance of the predicted values in the test set to the true values. Values from Model C were rounded to the nearest class of the original dataset.

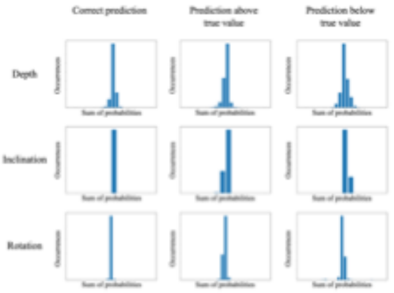


Fig. 7. Result of summing the probability vectors for each task for correct predictions and predictions above and below the true value. Vectors were shifted so the predicted value is always in the middle of the vector. The graphs show how the weights of the probabilities tend towards the side of the true value.

and -90° values due to lack of samples compared to the other rotation values, but this does not imply worse results.

As the data that support this study are very specific, we did not find other results that could be compared with those obtained.

B. Confidence Measurements

Since the results of our predictions are meant to be used in surgical procedures, it is important to convey not only the predicted values, but also the confidence level of the result. The fact that those types of images are not commonly used by doctors and are not easy to interpret makes the importance of a reliable information even greater. For that purpose, in Fig. 6 we plotted the distance of the predicted values in the test set to the true values. Values from the regression model (Model C) were rounded to the nearest class of the original dataset. From those plots we can see that the values are not scattered, but even in the rotation predictions of our best model (Model B), were we got close to perfect results, we get values that are far from the true value and could lead to disastrous outcome in a surgical procedure if not accompanied by some kind of warning.

Our first approach was to calculate the root mean square error (RMSE) between the prediction values and the true values in the test data for each class in each task and use that as a basis for confidence. The resulting value of the RMSE for each class is used directly by the regression model while the classification models rounds it up to the next step value.

We also created an approach to benefit from the probabilities vector returned by the classification methods. We analyzed those probabilities in three different situations: when the prediction was correct, when the prediction was above the correct value and when the prediction was below the correct value. Fig. 7 shows the results of summing the probability vectors for each task in the situations described above. We shifted the vectors, so the predicted value is always in the middle of the vector. From the plots we can clearly see that when the prediction is correct, the nearby probabilities are somehow symmetric, while when prediction is not correct, the probabilities are skewed towards the side of the correct value. The graphs shown are only for Model B, but Model A presented a similar behavior. Taking advantage of this, we created a procedure to sum the class values weighted by the probability vector and calculate the absolute difference of this result to the predicted value. That final value gives us an idea of how sure the network is about its prediction. When we divide the resulting value by the step value of the class, we get the number of steps of confidence for the prediction. In our tests, we defined that the minimum level of confidence is one step and results above that value are rounded down to the nearest step.

TABLE II. COMPARISON OF PREDICTED VALUES WITH AND WITHOUT UNCERTAINTY MEASURES FOR EACH MODEL

Without Uncertainty				
		Correct	***Wrong***	
Model A	Dep.	1652	1809	
	Inc.	3418	43	
	Rot.	3142	319	
	Tot.	79.09%	20.91%	
Model B	Dep.	2305	1156	
	Inc.	3455	6	
	Rot.	3380	81	
	Tot.	88.03%	11.97%	
Model C	Dep.	1765	1587	
	Inc.	3323	29	
	Rot.	2902	450	
	Tot.	79.46%	20.54%	
Weighted Sum Error				
		Overrated	***Correct***	***Underrated***
Model A	Dep.	2169	1176	116
	Inc.	24	3436	1
	Rot.	956	2496	9
	Tot.	30.33%	68.46%	1.21%
Model B	Dep.	0	3312	149
	Inc.	0	3461	0
	Rot.	3	3454	4
	Tot.	0.03%	98.50%	1.47%
RMSE Error				
		Overrated	***Correct***	***Underrated***
Model A	Dep.	999	2299	163
	Inc.	0	3461	0
	Rot.	148	3296	17
	Tot.	11.05%	87.22%	1.73%
Model B	Dep.	513	2852	96
	Inc.	0	3461	0
	Rot.	0	3456	5
	Tot.	4.94%	94.09%	0.97%
Model C	Dep.	259	2910	183
	Inc.	0	3352	0
	Rot.	213	3079	60
	Tot.	4.69%	92.89%	2.42%

[a] Dep. = Depth, Inc. = Inclination, Rot. = Rotation, Tot. = Total

Table 2 shows the percentage of correct and wrong predictions in the test set by each model and the results of using both confidence procedures presented in this section. The overrated column represents values where the confidence predicted was higher than needed, while the underrated

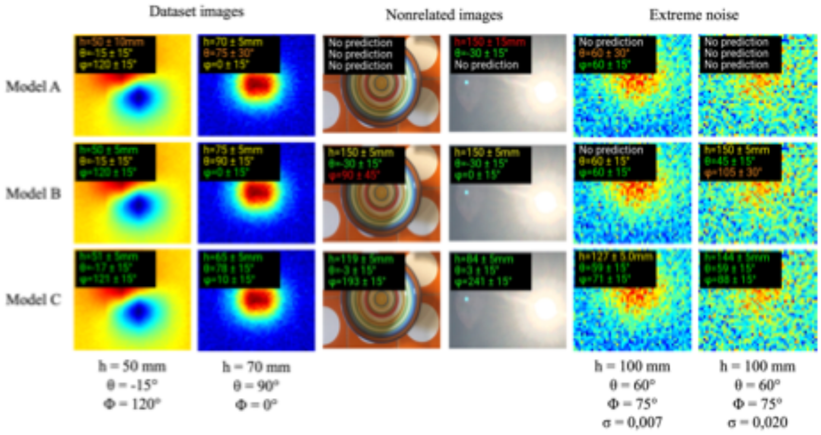


Fig. 8. Visualization of the predictions with the corresponding image in different situations for all models.

column represents values where the confidence predicted was not enough to reach the true value. All results got less than 2.5% of underrated confidence, and Model B using the weighted sum procedure obtained the best correctness with 98.50% of correct value/confidence prediction. Another important aspect is that all underrated predictions have a low level of confidence and are within two steps of distance from the true value, so when we present our model's predictions, not only we show the confidence values, but we also emphasize it with colors ranging from green, for high confidence, to red, for low confidence so doctors can clearly interpret the models' response.

Fig. 8 shows the visualization of the predictions with the corresponding image in different situations for all models. All models are accurate in their responses but present different characteristics. Model B seems to be overconfident in extreme situations while Model A is more conservative even in more usual predictions. This is an interesting behavior since the confidence measurement is compensating for the model's precision. Model C is the only one only able to predict values beyond those presented in the training process, but the lack of a vector of probabilities makes it extremely confident even in unusual situations.

IV. Conclusion and Future Work

This paper proposed different CNN architectures to be used as a tool for finding foreign objects in the human body based on the image of their magnetic field. We analyzed the obtained results and suggested different procedures to calculate the confidence of the predicted values, due to the importance of not proposing very dispersed values without a warning for use in surgical applications. The results obtained in this study show that CNN have great potential to provide reliable results for future use in commercial applications. Our best model achieved a micro-average precision of 66 % for the depth, 100 % for the angle of inclination and 98 % for the angle of rotation in the test dataset, and estimated confidence correctly in 98,50% of all cases and underestimated it in only 1,47% of all cases. Further research should be done with larger datasets to include more parameters such as the size of the needle, its thickness and other characteristics that could be found in real situations.

References

[1] Oliveira, B. R. C.; Louzada, D. R. ; Costa Monteiro, E. ; Hall Barbosa, C. R. Automatic system for locating magnetic foreign bodies using gmi magnetometer. In: XIII Congresso Internacional de Metrologia Elétrica (SEMETRO), 2019, Florianópolis. Anais do XIII Congresso Internacional de Metrologia Elétrica (SEMETRO), 2019.

[2] Monteiro, E. C. et al. Locating steel needles in the human body using a SQUID magnetometer. Physics in Medicine & Biology, v. 45, n. 8, p. 2389, 2000. ISSN 0031-9155.

[3] Barbosa, C. H. et al. Improvement of a technique for localization of steel needles in humans using a SQUID magnetometer. IEEE transactions on applied superconductivity, v. 11, n. 1, p. 677-680, 2001. ISSN 1051-8223.

[4] Barbosa, C. H. Localization of firearm projectiles in the human body using a superconducting quantum interference device magnetometer: A theoretical study. Review of scientific instruments, v. 75, n. 6, p. 2098-2106, 2004. ISSN 0034-6748.

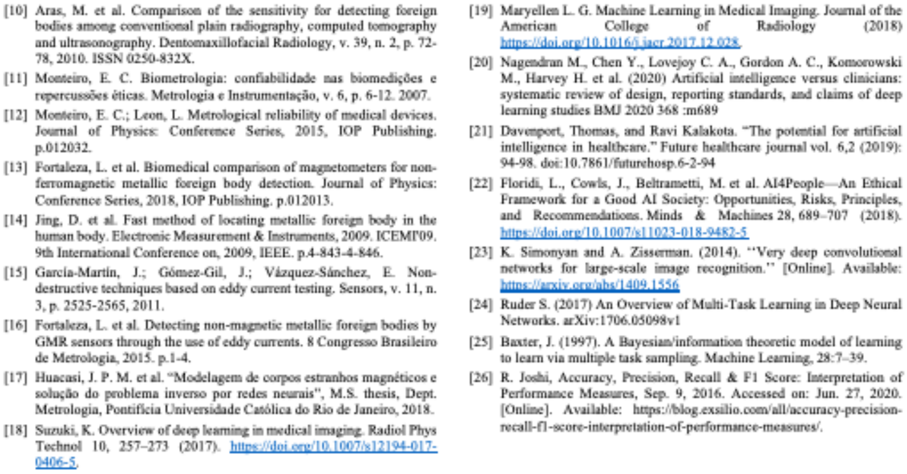
[5] Mcquirter, J. L. et al. The effects of retained lead bullets on body lead burden. Journal of Trauma and Acute Care Surgery, v. 50, n. 5, p. 892-899, 2001. ISSN 2163- 0755.

[6] Miranda, J. A.; Borges, M.; Mendes, E. Projéctil de Arma de Fogo como Corpo Estranho Nasal: Relato de Caso. Arquivos de ORL, v. 10, n. 4, p. 98-105, 2006.

[7] Venter, N. G. et al. Avaliação de métodos radiológicos na detecção de corpo estranho de madeira em modelo animal. Acta Cirurgica Brasileira, v. 20, p. 19-26, 2010.

[8] Oliveira Filho, K. S. Fundamentos de radiodiagnóstico por imagem. Instituto de Física da UFRGS. Disponível on-line em: <http://www.if.ufrgs.br/ast/med/imagens/index. htm>. Último acesso em, v. 19, n. 06, 2006.

[9] Ginsburg, M. J.; Ellis, G. L.; Flom, L. L. Detection of soft-tissue foreign bodies by plain radiography, xerography, computed tomography, and ultrasonography. Annals of emergency medicine, v. 19, n. 6, p. 701-703, 1990. ISSN 0196-0644.



[10] Aras, M. et al. Comparison of the sensitivity for detecting foreign bodies among conventional plain radiography, computed tomography and ultrasonography. Dentomaxillofacial Radiology, v. 39, n. 2, p. 72-78, 2010. ISSN 0250-832X.

[11] Monteiro, E. C. Biometrologia: confiabilidade nas biomedições e repercussões éticas. Metrologia e Instrumentação, v. 6, p. 6-12. 2007.

[12] Monteiro, E. C.; Leon, L. Metrological reliability of medical devices. Journal of Physics: Conference Series, 2015, IOP Publishing. p.012032.

[13] Fortaleza, L. et al. Biomedical comparison of magnetometers for non-ferromagnetic metallic foreign body detection. Journal of Physics: Conference Series, 2018, IOP Publishing. p.012013.

[14] Jing, D. et al. Fast method of locating metallic foreign body in the human body. Electronic Measurement & Instruments, 2009. ICEMI'09. 9th International Conference on, 2009, IEEE. p.4-843-4-846.

[15] García-Martín, J.; Gómez-Gil, J.; Vázquez-Sánchez, E. Non-destructive techniques based on eddy current testing. Sensors, v. 11, n. 3, p. 2525-2565, 2011.

[16] Fortaleza, L. et al. Detecting non-magnetic metallic foreign bodies by GMR sensors through the use of eddy currents. 8 Congresso Brasileiro de Metrologia, 2015. p.1-4.

[17] Huacasi, J. P. M. et al. "Modelagem de corpos estranhos magnéticos e solução do problema inverso por redes neurais'', M.S. thesis, Dept. Metrologia, Pontifícia Universidade Católica do Rio de Janeiro, 2018.

[18] Suzuki, K. Overview of deep learning in medical imaging. Radiol Phys Technol 10, 257–273 (2017). https://doi.org/10.1007/s12194-017-0406-5.

[19] Maryellen L. G. Machine Learning in Medical Imaging. Journal of the American College of Radiology (2018) https://doi.org/10.1016/j.jacr.2017.12.028.

[20] Nagendran M., Chen Y., Lovejoy C. A., Gordon A. C., Komorowski M., Harvey H. et al. (2020) Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies BMJ 2020 368 :m689

[21] Davenport, Thomas, and Ravi Kalakota. "The potential for artificial intelligence in healthcare." Future healthcare journal vol. 6,2 (2019): 94-98. doi:10.7861/futurehosp.6-2-94

[22] Floridi, L., Cowls, J., Beltrametti, M. et al. AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. Minds & Machines 28, 689–707 (2018). https://doi.org/10.1007/s11023-018-9482-5

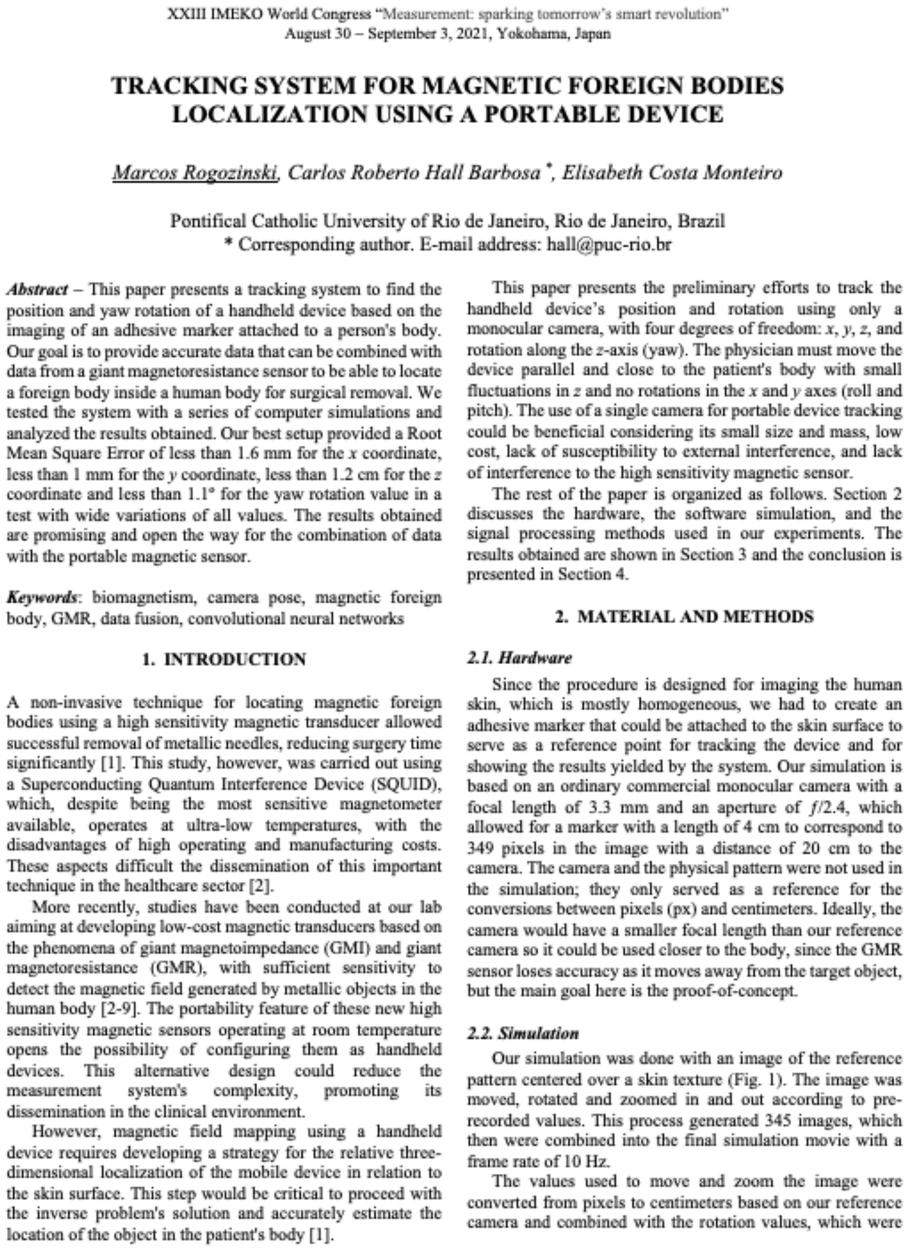
[23] K. Simonyan and A. Zisserman. (2014). ''Very deep convolutional networks for large-scale image recognition.'' [Online]. Available: https://arxiv.org/abs/1409.1556

[24] Ruder S. (2017) An Overview of Multi-Task Learning in Deep Neural Networks. arXiv:1706.05098v1

[25] Baxter, J. (1997). A Bayesian/information theoretic model of learning to learn via multiple task sampling. Machine Learning, 28:7–39.

[26] R. Joshi, Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures, Sep. 9, 2016. Accessed on: Jun. 27, 2020. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.

Appendix B: Tracking system for magnetic foreign bodies localization using a portable device



XXIII IMEKO World Congress "Measurement: sparking tomorrow's smart revolution"
August 30 – September 3, 2021, Yokohama, Japan

TRACKING SYSTEM FOR MAGNETIC FOREIGN BODIES LOCALIZATION USING A PORTABLE DEVICE

Marcos Rogozinski, *Carlos Roberto Hall Barbosa* *, *Elisabeth Costa Monteiro*

Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
* Corresponding author. E-mail address: hall@puc-rio.br

Abstract – This paper presents a tracking system to find the position and yaw rotation of a handheld device based on the imaging of an adhesive marker attached to a person's body. Our goal is to provide accurate data that can be combined with data from a giant magnetoresistance sensor to be able to locate a foreign body inside a human body for surgical removal. We tested the system with a series of computer simulations and analyzed the results obtained. Our best setup provided a Root Mean Square Error of less than 1.6 mm for the x coordinate, less than 1 mm for the y coordinate, less than 1.2 cm for the z coordinate and less than 1.1° for the yaw rotation value in a test with wide variations of all values. The results obtained are promising and open the way for the combination of data with the portable magnetic sensor.

Keywords: biomagnetism, camera pose, magnetic foreign body, GMR, data fusion, convolutional neural networks

1. INTRODUCTION

A non-invasive technique for locating magnetic foreign bodies using a high sensitivity magnetic transducer allowed successful removal of metallic needles, reducing surgery time significantly [1]. This study, however, was carried out using a Superconducting Quantum Interference Device (SQUID), which, despite being the most sensitive magnetometer available, operates at ultra-low temperatures, with the disadvantages of high operating and manufacturing costs. These aspects difficult the dissemination of this important technique in the healthcare sector [2].

More recently, studies have been conducted at our lab aiming at developing low-cost magnetic transducers based on the phenomena of giant magnetoimpedance (GMI) and giant magnetoresistance (GMR), with sufficient sensitivity to detect the magnetic field generated by metallic objects in the human body [2-9]. The portability feature of these new high sensitivity magnetic sensors operating at room temperature opens the possibility of configuring them as handheld devices. This alternative design could reduce the measurement system's complexity, promoting its dissemination in the clinical environment.

However, magnetic field mapping using a handheld device requires developing a strategy for the relative three-dimensional localization of the mobile device in relation to the skin surface. This step would be critical to proceed with the inverse problem's solution and accurately estimate the location of the object in the patient's body [1].

This paper presents the preliminary efforts to track the handheld device's position and rotation using only a monocular camera, with four degrees of freedom: x, y, z, and rotation along the z-axis (yaw). The physician must move the device parallel and close to the patient's body with small fluctuations in z and no rotations in the x and y axes (roll and pitch). The use of a single camera for portable device tracking could be beneficial considering its small size and mass, low cost, lack of susceptibility to external interference, and lack of interference to the high sensitivity magnetic sensor.

The rest of the paper is organized as follows. Section 2 discusses the hardware, the software simulation, and the signal processing methods used in our experiments. The results obtained are shown in Section 3 and the conclusion is presented in Section 4.

2. MATERIAL AND METHODS

2.1. Hardware

Since the procedure is designed for imaging the human skin, which is mostly homogeneous, we had to create an adhesive marker that could be attached to the skin surface to serve as a reference point for tracking the device and for showing the results yielded by the system. Our simulation is based on an ordinary commercial monocular camera with a focal length of 3.3 mm and an aperture of f/2.4, which allowed for a marker with a length of 4 cm to correspond to 349 pixels in the image with a distance of 20 cm to the camera. The camera and the physical pattern were not used in the simulation; they only served as a reference for the conversions between pixels (px) and centimeters. Ideally, the camera would have a smaller focal length than our reference camera so it could be used closer to the body, since the GMR sensor loses accuracy as it moves away from the target object, but the main goal here is the proof-of-concept.

2.2. Simulation

Our simulation was done with an image of the reference pattern centered over a skin texture (Fig. 1). The image was moved, rotated and zoomed in and out according to pre-recorded values. This process generated 345 images, which then were combined into the final simulation movie with a frame rate of 10 Hz.

The values used to move and zoom the image were converted from pixels to centimeters based on our reference camera and combined with the rotation values, which were

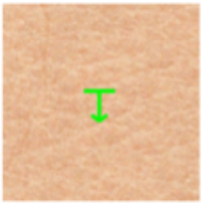


Fig. 1. Pattern simulation at yaw = 0° combined with the skin texture image used in the simulation.

already defined in degrees, to serve as ground truth values for later comparison with the test predictions.

The x and y values, representing the distance from the center of the reference pattern to the center of the image, were further processed with (1) and (2) to convert them from the camera coordinate system to an inertial frame of reference, with yaw values (ψ) in radians.

$$X_c = (x \,.\cos\psi) - (y \,.\sin\psi) \qquad (1)$$

$$Y_c = (x \,.\sin\psi) + (y \,.\cos\psi) \qquad (2)$$

Fig. 2 illustrates how finding the position and rotation angles of the device in the inertial frame of reference is one of the necessary steps to solve the magnetic inverse problem to estimate the position, rotation and inclination of the foreign object inside the human body.

2.3. Prediction process

2.3.1. Convolutional Neural Network

The first step in the prediction process is to feed each image into a Convolutional Neural Network (CNN) to detect the pattern position in the image and apply semantic segmentation to separate the pixels pertaining to the background pattern. We used the Mask R-CNN architecture [10] since it is capable of doing both procedures simultaneously with a frame rate of 5 Hz.

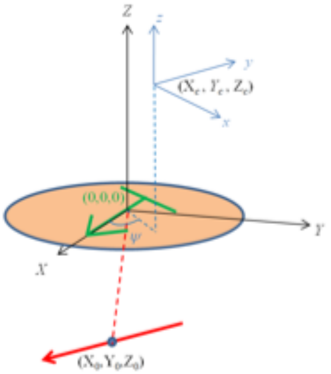


Fig. 2. The blue axis represents the values found by the simulation before convertion to the inertial frame of reference, which is represented by the black axis. The position of the pattern sticker is shown in green and the center of the foreign body with center in (X_0,Y_0,Z_0) is shown in red.

To train our model we built a dataset of 1835 images composed of skin textures downloaded from the internet with or without hair, bruises and tattoos, combined with views of our pattern in different shades of green and different angles of rotation in the z-axis (yaw). Each image was annotated by hand with polylines for training. Our model uses a ResNet 101 backbone [11] initialized with pre-trained weights from the MS COCO dataset [12], with only stage 5 and up of the ResNet being trained with our dataset. We also used random data augmentation during training with flips, rotation, gaussian blur and pixel multiplication by random values. One output of the Mask R-CNN is a bounding box representing the pattern's position in the image. The other output is a mask of the pixels that represent the pattern in the image.

2.3.2. Mask tensor

To find the yaw of the pattern and to better approximate the x, y and z values, we created a tensor representing the pattern in each possible yaw angle in steps of 1 degree. We first created an image of the pattern with a height of 88 px in each angle with a transparent background. We then cropped each image to fit the pattern perfectly, padded the sides of the image as needed with transparency to make it into a square shape, resized it to 48 x 48 pixels and transformed the image into a matrix of zeros and ones, with ones representing the pattern and zeros representing the background. Finally, we combined all matrices into a tensor where the index value represents the yaw angle. When the system receives the mask representing the pattern from the semantic segmentation model, it does the same steps of cropping, padding, resizing and converting to a matrix so it can be compared to our reference tensor. After conversion, the pattern representation is multiplied by every matrix in the tensor and each matrix is summed, so the index with the highest value indicates the yaw of the pattern.

Along with the mask tensor, we also created a separate tensor representing the size of the bounding box that perfectly matches the reference pattern with a height of 88 px at any given yaw angle. With that, we can compare the size of the bounding box returned by the object detection model to find the height of the predicted pattern and estimate the Z_c value using the dimensions found with our reference camera of 20 cm distance for a 349 px pattern.

2.3.3. Signal processing

Different adjustments are made during the prediction process. A buffer was created in the output of the CNN to detect outliers or missing values. In both cases, the value is substituted by linear interpolation. Outliers are defined as values that are more than two standard deviations away from the buffer's mean value. We found out that, for our simulations, the buffer was not needed, since the output of the CNN had no missing frames or any high inaccuracy, due to the controlled environment of the simulation, but in tests made with the real camera a buffer with 11 values was necessary to overcome discrepant data.

Another required adjustment is due to the difference between the center of the pattern and the center of the bounding box when the pattern rotates. By experimental analysis, we found that this difference peaks at 45°, 135°, 225°, and 315° angles, with a displacement of around 14 px in x and y values for the reference pattern with 88 px of height, returning to zero at every 90 degrees. For our estimation, we considered this behavior to be linear and used the yaw value and the ratio between the reference bounding box and the predicted bounding box to find the adjustment needed.

During the simulations, we also discovered that the masks generated by the semantic segmentation were rounding the corners of the pattern, leading to a consistently shorter prediction of the marker size. We did some experiments with different sizes in various positions and rotations to compare the amount of difference between the real and predicted pattern size and used the mean of all values as a factor to be multiplied by all pattern size values of our predictions.

Since the predicted bounding box and pattern mask have small fluctuations in its values even in more controlled environments, we applied the Savitzky–Golay filter [13] to smooth all values in different places of our system: for yaw and Z_c values, which serve as a basis for adjusting the other values, we tested the filter being applied before and after the adjustments; for x and y values the filter is applied just before the output, after all adjustments have been made. The filter was set up as a first-order polynomial with a window length of 7 coefficients and no derivatives. After all processing stages, the x and y values are converted to the inertial frame of reference using (1) and (2). Fig. 3 shows the prediction process with all its components.

3. RESULTS

3.1. Simulation results

The results were obtained in an iterative process of visualizing the data for each variable alone and in groups to find out how they interact, and by creating the corresponding adjustments. Fig. 4 shows the combination of X_c and Y_c values in four different system setups. Setup A has no adjustments at all and we can clearly see how the values fluctuate around or close to the true values.

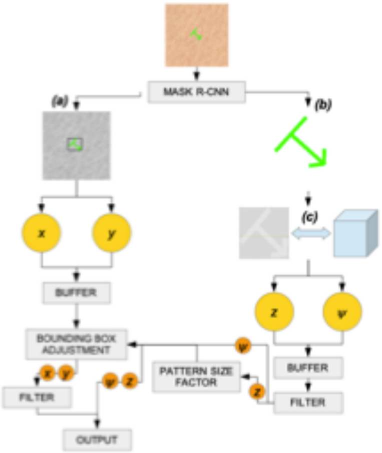


Fig. 3. The prediction process with object detection (a), semantic segmentation (b), the predicted mask and mask tensor comparison (c), and other adjustments.

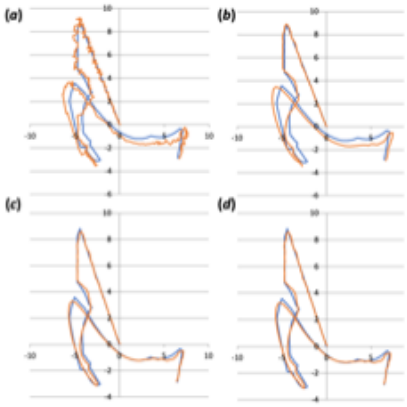


Fig. 4. Plot of X_c, Y_c and y values showing the trajectory of the pattern with the true values in blue and the prediction values for setups ***a***, ***b***, ***c***, and ***d*** in orange, with values in cm.

That behavior is also caused by fluctuations in the Z_c and yaw values, since they are used for the conversions from pixels to centimeters and from the camera coordinate system to the inertial frame of reference. In Setup B, we used the filter at the output of the system to smooth the values, getting a better approximation. The prediction, however, moves away from the true values in some parts of the simulation, especially when the marker is rotating and zooming. Setup C and D include the bounding box and the pattern size adjustments with the only difference that in Setup C the filter is applied to the yaw and Z_c values at the output, after adjustments have been made, while in Setup D they are applied before any adjustments. That difference should affect mainly the X_c and Y_c values, since they are dependent on the yaw and Z_c values for the bounding box adjustment and the conversion to the inertial frame of reference, although in our simulation both setups presented similar results. In Table 1, we can see how the filter improves the Root Mean Square Error (RMSE) of all values in Setup B, while the adjustments of the latter two setups promote an even bigger drop in error for the X_c, Y_c and Z_c values. Since the values obtained by the comparison of the predicted mask with the mask tensor were satisfactory, yaw values do not have any adjustments besides filtering and are not affected in those latter models.

Fig. 5 shows how the adjustments affect the X_c, Y_c and Z_c values. The Z_c values are affected by the pattern size factor adjustment, while X_c and Y_c values are affected by the

Table 1. RMSE for different prediction setups.

	x (cm)	y (cm)	z (cm)	yaw (°)
Setup A	0.368	0.318	3.667	1.577
Setup B	0.348	0.290	3.555	1.059
Setup C	0.157	0.091	1.159	1.059
Setup D	0.157	0.092	1.236	1.059

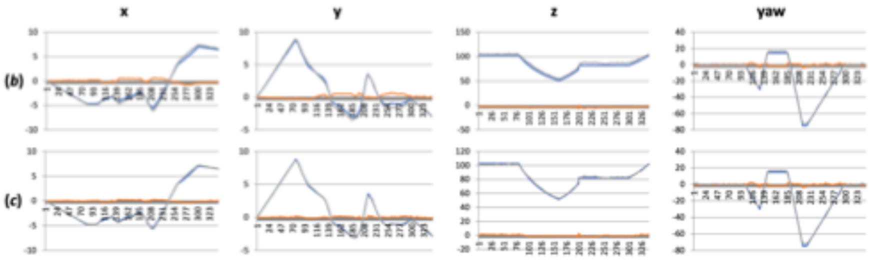


Fig. 5. Comparison between Setup B and C for each variable analysed. True values are shown in blue, prediction values are shown in grey, and the difference between them is shown in orange. The x, y, and z values are in cm, and the yaw values are in degrees.

improvement in Z_c values combined with the bounding box adjustment.

4. CONCLUSIONS AND FUTURE WORKS

This paper proposed a combination of different procedures to track the position and yaw rotation of a portable device based on a reference pattern, using only a monocular camera. The proposed system includes a CNN for object detection and semantic segmentation, a buffer for outlier detection, a mask comparison algorithm, adjustment algorithms and filtering. A computer simulation was done to test the system and the results obtained showed good RMSE values for the desired application. Further work should be done to test the system in real situations and to include roll and pitch values in the predictions. Data fusion should also be considered by comparing and possibly combining data from accelerometer and gyroscope sensors in order to obtain even better accuracy in the results providing reliable data to our final goal, which is to use a handheld high sensitivity magnetic sensor to find the position, angle of rotation and angle of inclination of a foreign body inside the human body.

ACKNOWLEDGMENTS

The authors thank for the financial support provided by the Brazilian funding agencies CNPq, CAPES, FINEP and FAPERJ. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] E. Costa Monteiro, C. R. H. Barbosa, E. A. Lima, P. Costa Ribeiro, P. Boechat, Locating steel needles in the human body using a SQUID magnetometer. *Physics in Medicine & Biology*, v. 45, n. 8, pp. 2389, 2000.

[2] E. Costa Monteiro, L. F. Leon, Metrological Reliability of Medical Devices. *Journal of Physics: Conference Series*, v. 588, pp. 012032, 2015.

[3] F. Pompéia, L. A. P. Gusmão, C. R. H. Barbosa, E. Costa Monteiro, L. A. P. Gonçalves, and F. L. A. Machado, "Ring shaped magnetic field transducer based on the GMI effect," *Meas. Sci. Technol.*, v. 19, pp. 025801, 2008.

[4] E. C. Silva, L. A. P. Gusmão, C. R. Hall Barbosa, E. Costa Monteiro, F. L. A. Machado, "High sensitivity giant magnetoimpedance (GMI) magnetic transducer: Magnitude versus phase sensing," Meas. Sci. Technol., v. 29, pp. 035106, 2011.

[5] E. Costa Silva, L. A. P. Gusmão, C. R. H. Barbosa, E. Costa Monteiro, "An enhanced electronic topology aimed at improving the phase sensitivity of GMI sensors," Meas. Sci. Technol., v. 25, pp. 115010, 2014.

[6] L. G. S. Fortaleza, C. R. H. Barbosa, E. Costa Monteiro, E. C. Silva, L. A. P. Gusmão, Detecting non-magnetic metallic foreign bodies by GMR sensors through the use of eddy currents. *XI International Congress on Electrical Metrology* (SEMETRO), pp.1-4, 2015.

[7] L. G. S. Fortaleza, E. Costa Monteiro, C. R. H. Barbosa, E. C. Silva, L. A. P. Gusmão, Biomedical comparison of magnetometers for non- ferromagnetic metallic foreign body detection. *Journal of Physics: Conference Series*, IOP Publishing. pp.012013, 2018.

[8] L. S. Benavides, E. C. Silva, E. Costa Monteiro, "Pressure transducer based on the phase characteristics of GMI effect for measuring the arterial pulse wave," in *Journal of Physics: Conference Series*, v. 1065, pp. 072039, 2018.

[9] B. R. C. Oliveira, D. R. Louzada, E. Costa Monteiro, C. R. Hall Barbosa, Automatic system for locating magnetic foreign bodies using GMI magnetometer. *Journal of Physics: Conference Series* , v. 1826 (in press).

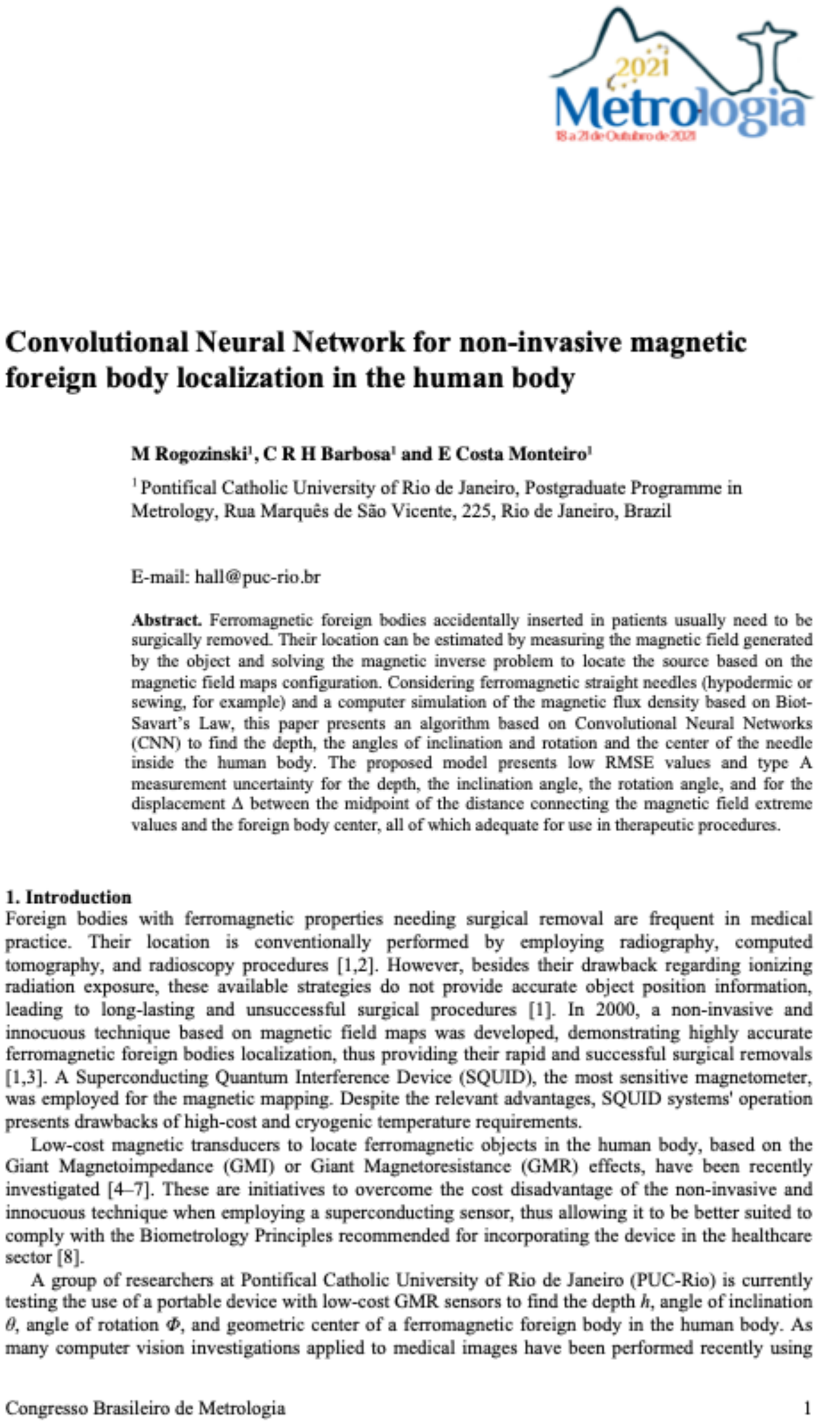
[10] K. He, G. Gkioxari, P. Dollar, R. Girshick, (2017, October). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). 2017 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2017.322

[11] K. He, X. Zhang, S. Ren, J. Sun, (2016, June). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, (2014). Microsoft COCO: Common Objects in Context. In Computer Vision – ECCV. Springer International Publishing. pp. 740–755, 2014 https://doi.org/10.1007/978-3-319-10602-1_48

[13] A. Savitzky, M. J. E. Golay, Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Analytical Chemistry, 36(8), 1627–1639. 1964 https://doi.org/10.1021/ac60214a047

Appendix C: Convolutional Neural Network for non-invasive magnetic foreign body localization in the human body



2021 Metrologia
18 a 21 de Outubro de 2021

Convolutional Neural Network for non-invasive magnetic foreign body localization in the human body

M Rogozinski[1], C R H Barbosa[1] and E Costa Monteiro[1]

[1] Pontifical Catholic University of Rio de Janeiro, Postgraduate Programme in Metrology, Rua Marquês de São Vicente, 225, Rio de Janeiro, Brazil

E-mail: hall@puc-rio.br

Abstract. Ferromagnetic foreign bodies accidentally inserted in patients usually need to be surgically removed. Their location can be estimated by measuring the magnetic field generated by the object and solving the magnetic inverse problem to locate the source based on the magnetic field maps configuration. Considering ferromagnetic straight needles (hypodermic or sewing, for example) and a computer simulation of the magnetic flux density based on Biot-Savart's Law, this paper presents an algorithm based on Convolutional Neural Networks (CNN) to find the depth, the angles of inclination and rotation and the center of the needle inside the human body. The proposed model presents low RMSE values and type A measurement uncertainty for the depth, the inclination angle, the rotation angle, and for the displacement Δ between the midpoint of the distance connecting the magnetic field extreme values and the foreign body center, all of which adequate for use in therapeutic procedures.

1. Introduction

Foreign bodies with ferromagnetic properties needing surgical removal are frequent in medical practice. Their location is conventionally performed by employing radiography, computed tomography, and radioscopy procedures [1,2]. However, besides their drawback regarding ionizing radiation exposure, these available strategies do not provide accurate object position information, leading to long-lasting and unsuccessful surgical procedures [1]. In 2000, a non-invasive and innocuous technique based on magnetic field maps was developed, demonstrating highly accurate ferromagnetic foreign bodies localization, thus providing their rapid and successful surgical removals [1,3]. A Superconducting Quantum Interference Device (SQUID), the most sensitive magnetometer, was employed for the magnetic mapping. Despite the relevant advantages, SQUID systems' operation presents drawbacks of high-cost and cryogenic temperature requirements.

Low-cost magnetic transducers to locate ferromagnetic objects in the human body, based on the Giant Magnetoimpedance (GMI) or Giant Magnetoresistance (GMR) effects, have been recently investigated [4–7]. These are initiatives to overcome the cost disadvantage of the non-invasive and innocuous technique when employing a superconducting sensor, thus allowing it to be better suited to comply with the Biometrology Principles recommended for incorporating the device in the healthcare sector [8].

A group of researchers at Pontifical Catholic University of Rio de Janeiro (PUC-Rio) is currently testing the use of a portable device with low-cost GMR sensors to find the depth h, angle of inclination θ, angle of rotation Φ, and geometric center of a ferromagnetic foreign body in the human body. As many computer vision investigations applied to medical images have been performed recently using convolutional neural networks (CNN) [9,10], incipient initiatives are being implemented for applying CNN architectures for estimating the location parameters of this new technique being developed [11]. This paper addresses a more advanced step of this research by developing a CNN architecture able to solve the magnetic inverse problem and accurately provide these positioning parameters, based on a simulation of the magnetic field map generated by the foreign body, as would be detected by a magnetometer integrated into a portable device.

The rest of the paper is organized as follows. Section 2 presents the materials and methods, including the proposed CNN architecture, the dataset used in our experiments, the hyperparameter configuration, and the experimental protocol. Section 3 shows the results obtained, while Section 4 summarizes the main conclusions.

2. Materials and Methods

2.1. Ferromagnetic Foreign Body

Figure 1 illustrates a needle with the parameters used in this study. The inclination angle θ, rotation angle ϕ, and depth h of the needle are direct outputs of the network. The geometric center of the needle, represented in figure 1 by p, is obtained by calculating the center between magnetic field extreme values and adding to it a displacement value Δ, also provided by the CNN. Figure 2 shows an example image of the magnetic field generated by the ferromagnetic foreign body, which serves as the input for our CNN, with added markings to highlight some parameters. The white markings are the minimum and maximum values of the magnetic field, and the black marking is the middle point between those extremes, while the red marking is the geometric center of the needle. Our CNN returns the value Δ, which corresponds to the difference between the center of the magnetic field extreme values and the needle's geometric center.

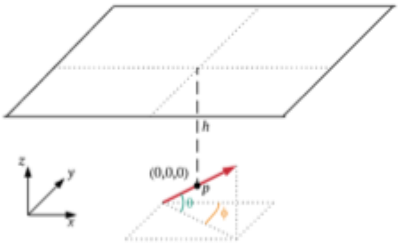


Figure 1. Visualization of a needle position (in red) with (0,0,0) as its center point p, showing the needle inclination angle θ, rotation angle ϕ, and depth h.

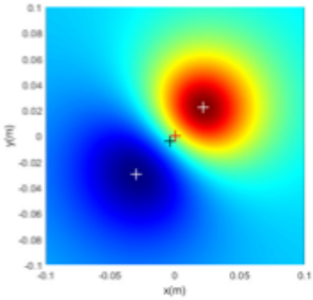


Figure 2. Colormap image representing the magnetic field with white markings on its extreme values, a black marking at the center between their positions, and a red marking at the geometric center of the ferromagnetic object that generated the map.

2.2. Dataset

The dataset was built by simulating the magnetic field data that would be read by our portable device during a complete scan. Each scan corresponds to a path taken by the device in a zig-zag pattern over a 20 cm x 20 cm area centered on a known reference, consisting of a marker affixed to the patient skin. The output values obtained from the CNN, along with the reference point marked on the patient's skin, will be used by the medical team to guide the surgical strategy.

The magnetic field data was generated based on the Biot-Savart law for a magnetic dipole, integrated over the length of the needle (50 mm). The scans were created by applying random variations to a reference path in the positions *x*, *y* and *z*, and in the z-axis' rotation (yaw). Rotations in the *x* and *y* axes (roll and pitch, respectively) were kept constant at zero.

Four different irregular paths were used and, for each one of them, 2 376 images were created representing the magnetic field generated by the dipole. These images encompass a variation in depth h from 50 mm to 150 mm, in 10 mm steps, a variation in inclination angle θ from -60° to 60°, in 15° steps, and a variation of rotation angle ϕ from 0° to 360°, in 15° steps. The delta Δ value was calculated from the magnetic field values for each entry according to the procedure presented in [3]. The final dataset comprises 9 504 images with corresponding h, θ, ϕ, and Δ values. Figure 3 shows in (a) an undisturbed scan path, with a perfect magnetic field map generated and, from (b) to (e), the four paths used in our dataset, with their corresponding generated magnetic field maps for the parameters h = 60 mm, θ = 15°, ϕ = 45° and Δ = 0.56 mm.

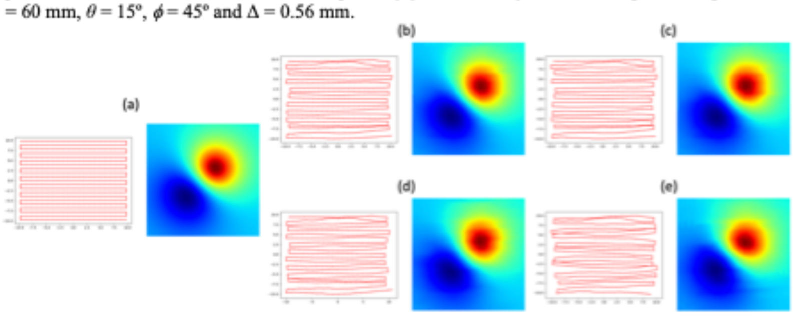


Figure 3. A perfect scan and magnetic field map (a) and the tracking variations used in our dataset (b, c, d, e) with the corresponding magnetic field map, for h = 60 mm, θ = 15°, ϕ = 45° and Δ = 0.56 mm.

Most images with imperfect scan paths or not completely centered have missing pixels near their edges. Since interpolation cannot be applied in those situations, we have created a procedure to fill in those missing pixels. The procedure checks each image pixel, from the image's center to its edges, in a spiral movement. If any pixel is missing, it fills it with the average value of the available surrounding pixels. Figure 4 shows the same image as figure 3e after a scan simulation without (figure 4a) and with (figure 4b) our procedure to fill in missing pixels.

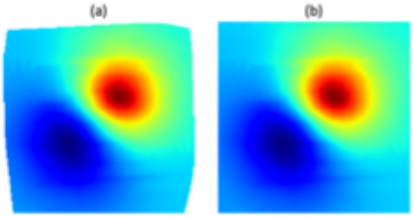


Figure 4. An image of the magnetic field map without (a) and with (b) the method to fill in missed values from the scan procedure.

2.3. Convolutional Neural Network Model

The proposed network is based on a shorter version of a VGG architecture [12] setup for regression, with branches for each output value to be found (h, θ, ϕ and Δ). The first layers of the network are shared, and the last layers are task-specific. We expected that this approach provides good generalization while keeping most of the learnable parameters specific for each task.

At its input, the network expects an image of the magnetic field values with 80 x 80 pixels with three color channels in RGB format for each pixel (red, green, and blue values). The network uses a linear activation function in the last layer with a mean squared error loss function in order to produce real numbers as the output for each task. Figure 5 shows the proposed network with the types of layers used and the number of filters for each layer.

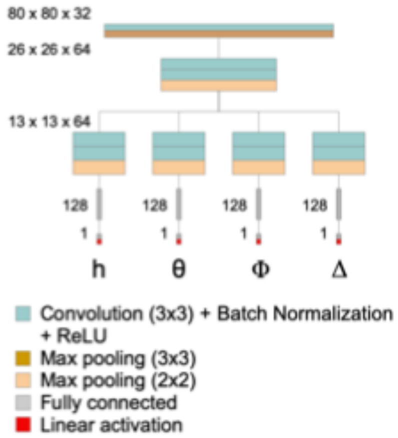


Figure 5. Proposed network showing the type of layers, the number of filters per layer, and the output for each task: the depth h, the inclination angle θ, the rotation angle ϕ, and the displacement Δ between the position of the center of the foreign body and the midpoint between the positions of the extreme magnetic field values.

2.4. Hyperparameter tuning

The model was tested with different numbers of layers and different numbers of filters per layer in search of the best results considering the size of our dataset. The final configuration, shown in figure 5, has a total of 1 534 788 learnable parameters, with 56 960 of those being shared between h, θ, ϕ and Δ. Since the rotation can also be inferred from the magnetic field values directly, we also tested an equivalent model without the rotation branch with a total of 1 165 251 learnable parameters. Both models, with and without the rotation branch, were also tested with absolute Δ values only and considering negative and positive Δ values.

2.5. Experimental protocol

The dataset was randomly separated into disjoint sets, 60 % for training, 20 % for validation, and 20 % for testing. After experimental analysis, we used the Adam optimizer with a learning rate of 0.0001, and dropout was used to help prevent overfitting. Early stopping with a patience of 10 and model checkpoint callbacks were employed to control the training time and save only the best model during training. For the code, we used the Keras framework with Tensorflow backend. Figure 6 shows the evolution of the training and validation data for all values associated with the parameters of the magnetic source location (h, θ, ϕ and Δ).

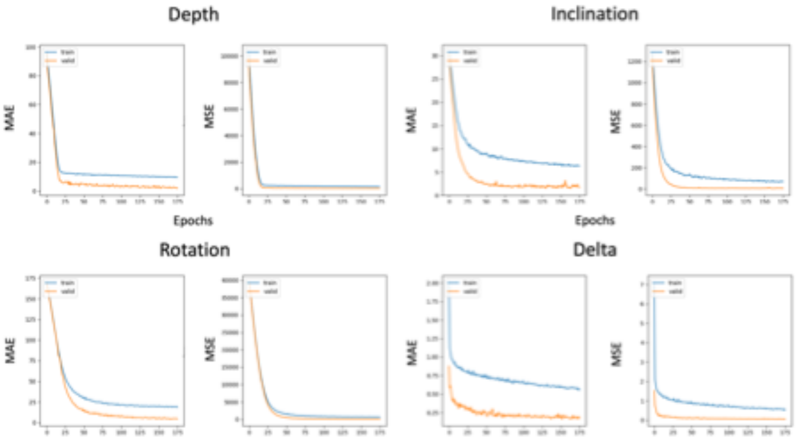


Figure 6. Evolution of Mean Absolute Error (MAE) and Mean Square Error (MSE) for the training (blue) and validation (orange) data over the epochs for the depth *h*, inclination *θ*, rotation *Φ*, and delta Δ values.

3. Results

Table 1 shows the results obtained for the model presented in figure 5 for each output variable. The model was trained using only absolute Δ values because it presented better results than using negative and positive Δ values. The model with the rotation branch also provided better results than the model with only the depth, inclination, and Δ branches. The final application should consider the absolute Δ values and invert the signal when the rotation is between 90° and 270°, with negative inclination, or when the rotation is between 270° and 90°, with positive inclination.

Table 1. Root Mean Square Error (RMSE), maximum error, and expanded type A uncertainty for each output variable for the test data.

Output variable	RMSE	Max. Error	Expanded Type A uncertainty
depth h	2.8 mm	12.6 mm (@ 130 mm)	5.6 mm
inclination θ	2.2°	7.0° (@ 60°)	4.4°
rotation ϕ	5.5°	19.9° (@ 345°)	11.1°
displacement Δ	0.2 mm	1.1 mm (@ 8.6 mm)	0.4 mm

Figure 7 presents the scatter plots between predicted and true values, and figure 8 presents the histograms of the prediction errors, both for the test data. The proposed model shows an improvement in maximum error for Δ compared to [3] while providing similar results for the other variables in terms of expanded type A uncertainty as those obtained in [11].

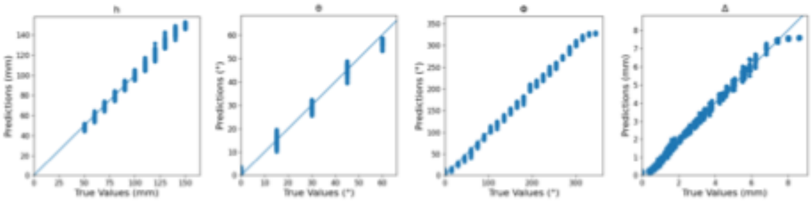


Figure 7. Scatter plot of the test data between predicted and true values for depth (h), inclination (θ), rotation (ϕ), and displacement (Δ) values.

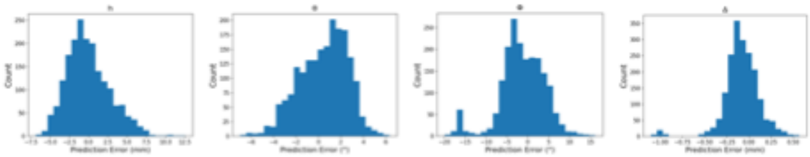


Figure 8. Histogram of the predicted error in the test data for depth (h), inclination (θ), rotation (ϕ), and displacement (Δ) values.

4. Conclusion and future work

This paper proposed a novel CNN architecture to locate foreign objects in the human body based on their magnetic field map image. Our best model achieved an RMSE value in the test data of 2.8 mm for the depth h, 2.2º for the inclination angle θ, 5.5º for the rotation angle ϕ, and 0.21 mm for the displacement Δ between the midpoint of the distance connecting the magnetic field extreme values and the foreign body center.

The obtained low RMSE and type A uncertainty values, also providing information regarding Δ values, a crucial localization parameter, with better performance than [3], point toward the promising feature of the proposed CNN to contribute to the accuracy of the non-invasive localization of ferromagnetic foreign bodies using a low-cost and portable transducer, thus providing the essential information for a safe, quick and successful surgical procedure for object removal [8]. Future steps in the study involve implementing the developed algorithm to solve the inverse problem from experimental magnetic maps.

Acknowledgments

We thank the financial support by the Brazilian funding agencies CNPq, FINEP and FAPERJ. This work was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

[1] Costa Monteiro E, Barbosa C H, Lima E A, Ribeiro P C and Boechat P R M 2000 Locating steel needles in the human body using a SQUID magnetometer

[2] Aras M H, Miloglu O, Barutcugil C, Kantarci M, Ozcan E and Harorli A 2010 Comparison of the sensitivity for detecting foreign bodies among conventional plain radiography, computed tomography and ultrasonography *Dentomaxillofac Radiol* **39** 72–8

[3] Barbosa C H, Costa Monteiro E, Lima E A, Santos S F, Cavalcanti E G and Ribeiro P C 2001 Improvement of a technique for localization of steel needles in humans using a SQUID magnetometer *IEEE Transactions on Applied Superconductivity* **11** 677–80

[4] Pompéia F, Gusmão L A P, Barbosa C R H, Costa Monteiro E, Gonçalves L A P and Machado F L A 2008 Ring shaped magnetic field transducer based on the GMI effect *Meas. Sci. Technol.* **19** 025801

[5] Fortaleza L G e S, Barbosa C R H, Costa Monteiro E, Silva E C da and Gusmão L A P de 2015 Detecting non-magnetic metallic foreign bodies by GMR sensors through the use of eddy currents *XI International Congress on Electrical Metrology (SEMETRO)* vol 1 pp 1–4

[6] Fortaleza L G S, Costa Monteiro E, Barbosa C R H, Silva E C and Gusmão L A P 2018 Biomedical comparison of magnetometers for non-ferromagnetic metallic foreign body detection *J. Phys.: Conf. Ser.* **1044** 012013

[7] Oliveira B R, Louzada D R, Costa Monteiro E and Barbosa C R H 2020 Automatic system for locating magnetic foreign bodies using GMI magnetometer *Journal of Physics*

[8] Costa Monteiro E and Leon L F 2015 Metrological Reliability of Medical Devices *J. Phys.: Conf. Ser.* **588** 012032

[9] Suzuki K 2017 Overview of deep learning in medical imaging *Radiol Phys Technol* **10** 257–73

[10] Giger M L 2018 Machine Learning in Medical Imaging *J Am Coll Radiol* **15** 512–20

[11] Rogozinski M, Barbosa C and Costa Monteiro E 2021 Localization of magnetic foreign bodies using CNN and GMI magnetometer IMEKO XXIII World Congress (Yokohama, Japan)

[12] Simonyan K and Zisserman A 2015 Very Deep Convolutional Networks for Large-Scale Image Recognition *arXiv:1409.1556 [cs]*