Inversion of geophysical data by deep learning

Julio José Cárdenas Chapellín



Sorbonne Université

École doctorale Informatique, Télécommunications et Électronique (ED 130), *UMR 7606 LIP6*

Institut de Recherche pour le Développement (IRD)

Unité de Modelisation Mathématique et Informatique des Systèmes Complèxes, *UMI 2019 UMMISCO*

Université Cadi Ayyad (UCA)

Laboratoire d'Ingénierie des Systèmes Informatiques (LISI), *Faculté de Sciences Semlalia (FSSM)*

Inversion of geophysical data by deep learning

Julio José CÁRDENAS CHAPELLÍN

Doctoral thesis in Geophysics

Directed by Christophe DENIS and Hajar MOUSANNIF

Co-supervised by Christian CAMERLYNCK and Nicolas FLORSCH

Soutenue le 26 août 2022

Before the jury composed by :

M. Frédéric NGUYEN	Professeur ordinaire, Université de Liège, Belgium	Rapporteur
M. Francisco CHINESTA	Professeur des Universités, Arts et Métiers ParisTech, France	Rapporteur
M. Blaise HANCZAR	Professeur des Universités, Université Paris-Saclay, France	Examinateur
Mme Laetitia LE POURHIET	Professeure des Universités, Sorbonne Université, France	Examinatrice
M. Christophe DENIS	Maître de conférences (HDR), Sorbonne Université, France	Directeur de thèse
Mme Hajar MOUSANNIF	Professeure, Université Cadi Ayyad, Marrakech	Directrice de thèse
M. Christian CAMERLYNCK	Maître de conférences, Sorbonne Université, France	Encadrant
M. Nicolas FLORSCH	Professeur, Sorbonne Université, France	Invité

A mi familia y amigos,
A las coincidencias de la vida.

A mi abuela (donde quiera que estés..)

Fais de ta vie un rêve, et d'un rêve, une réalité.

Antoine de Saint-Exupéry

The measure of a person is not how much they have lived. It is in how they make use of what life has shown them.

Brandon Sanderson

Una búsqueda comienza siempre con la suerte del principiante y termina siempre con la prueba del conquistador.

Paulo Coelho

Abstract

This thesis presents the characterization of magnetic anomalies using convolutional neural networks, and the application of visualization tools to understand and validate their predictions. The developed approach allows the localization of magnetic dipoles, including counting the number of dipoles, their geographical position, and the prediction of their parameters (magnetic moment, depth, and declination). Our results suggest that the combination of two deep learning models, "YOLO" and "DenseNet", performs best in achieving our classification and regression goals. Additionally, we applied visualization tools to understand our model's predictions and its working principle. We found that the Grad-CAM tool improved prediction performance by identifying several layers that had no influence on the prediction and the t-SNE tool confirmed the good ability of our model to differentiate among different parameter combinations. Then, we tested our model with real data to establish its limitations and application domain. Results demonstrate that our model detects dipolar anomalies in a real magnetic map even after learning from a synthetic database with a lower complexity, which indicates a significant generalization capability. We also noticed that it is not able to identify dipole anomalies of shapes and sizes different from those considered for the creation of the synthetic database. Our current work consists in creating new databases by combining synthetic and real data to compare their potential influence in improving predictions. Finally, the perspectives of this work consist in validating the operational relevance and adaptability of our model under realistic conditions and in testing other applications with alternative geophysical methods.

Key words

Geophysics, Deep learning, Convolutional neural networks, Magnetic methods, Explainability, Unexploded ordnance.

Résumé

Cette thèse présente la caractérisation d'anomalies magnétiques par des réseaux de neurones convolutifs, et l'application d'outils de visualisation pour comprendre et valider leurs prédictions. L'approche développée permet la localisation de dipôles magnétiques, incluant le comptage du nombre de dipôles, leur position géographique, et la prédiction de leurs paramètres (moment magnétique, profondeur, et déclinaison). Nos résultats suggèrent que la combinaison de deux modèles d'apprentissage profond, "YOLO" et "DenseNet", est la plus performante pour atteindre nos objectifs de classification et de régression. De plus, nous avons appliqué des outils de visualisation pour comprendre les prédictions de notre modèle et son principe de fonctionnement. Nous avons constaté que l'outil Grad-CAM a amélioré les performances de prédiction en identifiant plusieurs couches qui n'avaient aucune influence sur la prédiction et l'outil t-SNE a confirmé la bonne capacité de notre modèle à différencier différentes combinaisons de paramètres. Ensuite, nous avons testé notre modèle avec des données réelles pour établir ses limites et son domaine d'application. Les résultats montrent que notre modèle détecte les anomalies dipolaires dans une carte magnétique réelle, même après avoir appris d'une base de données synthétique de moindre complexité, ce qui indique une capacité de généralisation significative. Nous avons également remarqué qu'il n'est pas capable d'identifier des anomalies dipolaires de formes et de tailles différentes de celles considérées pour la création de la base de données synthétique. Nos travaux actuels consistent à créer de nouvelles bases de données en combinant des données synthétiques et réelles afin de comparer leur influence potentielle dans l'amélioration des prédictions. Enfin, les perspectives de ce travail consistent à valider la pertinence opérationnelle et l'adaptabilité de notre modèle dans des conditions réalistes et à tester d'autres applications avec des méthodes géophysiques alternatives.

Mots clés

Géophysique, Apprentissage profond, Réseau de neurones convolutifs, Méthodes magnétiques, Explicabilité, Munitions non explosées.

Resumen

Esta tesis presenta la caracterización de anomalías magnéticas mediante redes neuronales convolucionales, y la aplicación de herramientas de visualización para entender y validar sus predicciones. El enfoque desarrollado permite la localización de dipolos magnéticos, incluyendo el recuento del número de dipolos, su posición geográfica y la predicción de sus parámetros (momento magnético, profundidad y declinación). Nuestros resultados sugieren que la combinación de dos modelos de aprendizaje profundo, "YOLO" y "DenseNet", es la que mejor se ajusta a nuestros objetivos de clasificación y regresión. Adicionalmente, aplicamos herramientas de visualización para entender las predicciones de nuestro modelo y su principio de funcionamiento. Descubrimos que la herramienta Grad-CAM mejoraba el rendimiento de la predicción al identificar varias capas que no influían en la predicción y la herramienta t-SNE confirmaba la buena capacidad de nuestro modelo para diferenciar entre distintas combinaciones de parámetros. Seguidamente, probamos nuestro modelo con datos reales para establecer sus limitaciones y su rango de aplicación. Los resultados demuestran que nuestro modelo detecta anomalías dipolares en un mapa magnético real incluso después de aprender de una base de datos sintética con una complejidad menor, lo que indica una capacidad de generalización significativa. También observamos que no es capaz de identificar anomalías dipolares de formas y tamaños diferentes a los considerados para la creación de la base de datos sintética. Nuestro trabajo actual consiste en crear nuevas bases de datos combinando datos sintéticos y reales para comparar su posible influencia en la mejora de las predicciones. Por último, las perspectivas de este trabajo consisten en validar la pertinencia operativa y la adaptabilidad de nuestro modelo en condiciones realistas y en probar otras aplicaciones con métodos geofísicos alternativos.

Palabras claves

Geofísica, Aprendizaje profundo, Redes neuronales convolucionales, Métodos magnéticos, Explicabilidad, Municiones no detonadas.

Acknowledgments

Tout d'abord, je tiens à remercier Nicolas Florsch et Christian Camerlynck de m'avoir fait confiance dès le premier jour pour développer ce sujet de recherche en tenant compte de l'expertise de deux disciplines différentes. Un merci particulier à Nicolas pour m'avoir prêté son violon, ce qui m'a permis de redécouvrir le plaisir de jouer ce bel instrument. À Christophe Denis pour son implication, son soutien et son rôle important durant la période du Covid. À Hajar Mousannif pour sa bienveillance et nos nombreux échanges scientifiques qui m'ont permis d'avancer dans mes recherches. Tous ces conseils m'ont aidé à évoluer professionnellement et personellement. Je vous serai toujours très reconnaissant.

Me gustaría agradecer a mis padres y a mi hermana su apoyo y el haber estado siempre a mi lado. Por enseñarme a no rendirme nunca y a aceptar cualquier reto que se me presente, por enseñarme que con una actitud positiva se pueden conseguir cosas buenas y por aconsejarme siempre en los momentos difíciles. Je remercie également Jean-Claude Hustache, Rolande Gallé, Maya Guevara, Latifa Abdali, et leur familles pour m'avoir aidé inconditionnellement dans les moments les plus éprouvants de ma vie.

Je suis reconnaissant aux membres du laboratoire METIS, de l'UMMISCO et de l'IRD de m'avoir bien accueilli, pour tous les moments que nous avons partagés et pour les amitiés qui dureront longtemps.

La vie, avec le temps, nous apprend qu'il y a des amitiés qui peuvent devenir une famille. Pour cela, je tiens à remercier tout particulièrement Fátima, Francesco, Aída et Youssef. Et en même temps Emma, Bertille, Audrey, Marie, Marco, Itzel, Flore, Quentin, Hans, Spyridon, Ahmad, Robert, Radegonde, Bassirou. Nous avons vécu de nombreuses aventures ensemble, des randonnées, des semi-marathons, des voyages, des matchs de ping-pong, des Noëls, des excursions, des sorties culturelles, des discussions scientifiques et personnelles, entre autres. Je serai toujours reconnaissant pour le temps partagé avec vous tous et je suis impatient de vivre les nouvelles aventures que la vie nous réserve.

Contents

List of Figures

Introduction

This doctoral thesis was performed in the framework of the PDI-MSC (International Doctoral Program in Complex Systems Modelling). The PDI-MSC is a transversal program for all the doctoral schools of the Sorbonne University which participate in the selection and ensure the registration and supervision of students according to their main discipline. It is the first North-South doctoral program of the Sorbonne University built on the commitment of recruiting young PhDs from all countries and more particularly from Southern countries where the IRD has developed numerous collaborations for over 60 years. It also relies on the network set up by the International Joint Unit UMMISCO of the Sorbonne Université (SU) and the Institut de Recherche pour le Développement (IRD), whose teams specialized in mathematical and computer modeling are spread over 5 countries (Morocco, Senegal, Cameroon, Vietnam and France). This program was created by the convention between UMMISCO (Unité Mixte de Modélisation mathématique et informatique de systèmes complexes) and Sorbonne Université (Paris, France). The approach developed in my thesis consists of the application of deep learning networks for the characterization of magnetic anomalies.

Background and State of the art

Geological interpretations based on observed data are generally far from revealing the complex characteristics of subsurface properties. Hence, to address geological interpretation needs, geophysical inversion methods are continually focused on reconstructing more accurate and detailed subsurface properties. The application of machine learning to subsurface geophysics should notably allow to overcome the weaknesses of linear or non-linear inversion algorithms. Finally, the physics underlying the direct problem, which often generates artifacts during the inversion, may be ignored, guaranteeing a relative independence from an unsatisfactory configuration of the parameters.

The first neural networks were developed in the 1950s. However, until the 2000s, there was a lack of sufficient computational power and efficient algorithms in order to take more complex neural net-

work topologies into account, thus enabling improvements in their prediction capabilities. The gradient backpropagation algorithm (Rumelhart et al., 1986) coupled with vastly improved computational power has led to spectacular results in the field of pattern recognition and perception.

Many scientific disciplines are directing their research activities towards deep machine learning. This is also the case in geophysics, where it is used to facilitate the automated processing of geophysical data and the treatment of inversion problems, e.g., the identification of wave trains or the filtering of seismic noise (Van der Baan and Jutten, 2000), or to obtain an estimate of the global thickness of the Earth's crust (Meier et al., 2007).

In the last decade, deep learning methods have shown great promise, especially in the field of seismic interpretation. For example, Zhao (2019), Wu et al. (2019), and An et al. (2021) use convolutional neural networks (CNNs) for automatic fault recognition from seismic data sets. Das et al. (2018) also used CNNs to predict an elastic subsurface model directly from recorded seismic data, Araya-Polo et al. (2018) to produce an accurate layered velocity model from shot gathers, and Heagy et al. (2020) to demonstrate the potential utility of machine learning for the classification of unexploded ordnance (UXO) from electromagnetic data. Moreover, deep learning methods have been applied to other geological and geophysical problems, particularly geophysical inversion (e.g., Laloy et al., 2019; Mosser et al., 2018), well log interpretation (e.g., Anemangely et al., 2019; Ali et al., 2021), and ground penetrating radar (e.g., Ishitsuka et al., 2018; Travassos et al., 2021).

Magnetic data characterization using deep machine learning is a topic that has been addressed only very recently. A key example of recent studies in this area is the work of Guo et al. (2021), who have used deep learning models to perform inversion of 3D magnetic structures. To do this, they generated a dataset containing over 3 millions of 3D geological models by using the python library *PyNoddy*. Through Python script, they added geological structures including folds, faults and tilts, and change the geological structure parameters within a certain sampling interval. It took 105 hours to generate all the data sets, and the required storage space was 1.2 TB. Then, this database was divided into 60% training, 20% validation and 20% test data. Subsequently, the hyperparameters of the models were determined with the "grid search" technique. Finally, their trained neural network models are used to classify and predict the parameters of a geological structure present on magnetic maps.

Furthermore, Liu et al. (2020) propose a deep learning application for the inversion of electrical resistivity data. They propose to build a correspondence between the apparent resistivity data (input) and the resistivity model (output) directly by using convolutional neural networks (CNNs). They provide an additional level feature map to the CNNs to help them become aware of the relationship between

the input and the output. Based on the current U-Net architecture, they design a network (ERSInvNet) that can be trained end-to-end and achieve real-time inference during testing. In their work, they collect a dataset with 36,214 pairs of different resistivity patterns and corresponding apparent resistivity data. They generate synthetic data by predefining a few anomalous bodies with different resistivity value and then integrating them at different positions of the homogeneous medium (500 Ω-m). The resistivity anomalous bodies consist of 5 subsets as follows: Type I: Single rectangular body, Type II: Two rectangular bodies, Type III: Three rectangular bodies, Type IV: Single declining bodies, and Type V: Two declining bodies. After multiple comparative experiments, their model called "ERSInvNet" was able to obtain accurate inversion results and suppress false anomalies, which demonstrates its promising inversion capability.

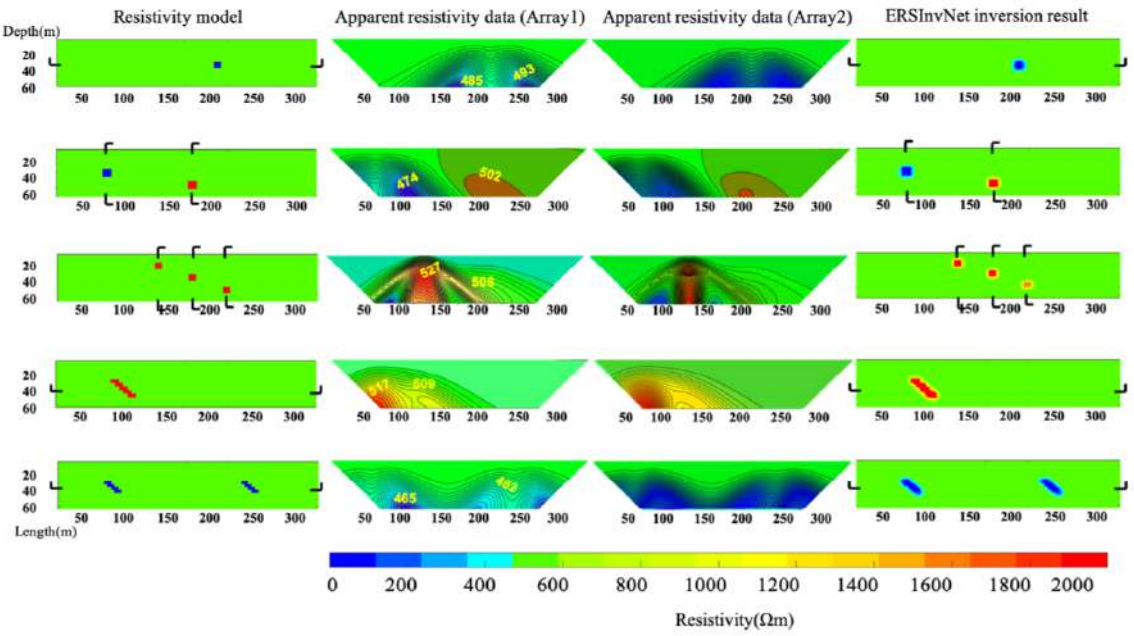


Figure 0.1 – Ground truth (the first column), inversion results of our ERSInvNet (the fourth column) and corresponding apparent resistivity data (the second and third column) on the test set. Vertical and horizontal profiles of resistivity models indicated by the truncation line are shown in the rightmost column for comparing the inverted resistivity values. Rows from top to bottom exhibit examples with the anomalous bodies from type I-V, respectively (Liu et al., 2020).

Structure of this thesis

This thesis is divided into four chapters. The first chapter consist of a theoretical framework that serves as the basis for the literature review, and most importantly, the methods and analysis. It presents several key concepts on geophysics and machine learning.

In chapter 2, I present an exhaustive explanation of the methodology developed in this research. It will start with the hypothesis proposed at the beginning of the research that allowed us to focus on

a machine learning architecture. Then, I will detail the evolution of the complexity of the synthetic data created for the learning phase. Lastly, I will describe and illustrate the evolution of our current architecture after multiple comparative experiments.

In Chapter 3, I discuss the results of the characterization of magnetic anomalies using deep learning models on synthetic and real data. I present these results in the form of a scientific paper (currently under first revision in the journal "Computer & Geosciences") together with the supplementary material of the article. In addition, I will highlight some advantages over traditional inversion softwares used nowadays

In chapter 4, I present the perspectives of this work. They consist of ongoing work that we plan to prepare for a second journal article and also other future applications that I believe are the next step in the application of deep learning models for geophysical inversion.

CHAPTER

1 Theoretical framework

This chapter intends to establish the theoretical background needed to understand the work I have done in my thesis. First, I present a background of machine learning where I put special emphasis from the comprehension of a neuron to the explanation of complex deep neural architectures. Finally, in the second part of this chapter, I describe a lot of concepts in geophysics, in particular the magnetic method, as it is a fundamental part of my research.

1.1 Machine Learning

Machine learning is a subfield of computer science that is concerned with building algorithms which, to be useful, rely on a collection of examples of some phenomenon. These examples can come from nature, be handcrafted by humans or generated by another algorithm. A machine learning system is *trained* rather than explicitly programmed. It is presented with many examples relevant to a task, and it finds statistical structures in these examples that eventually allows the system to come up with rules to automate the task (Burkov, 2019).

The main types of learning are supervised, semi-supervised, unsupervised and reinforcement. In this chapter, I will only focus on supervised learning because this is the type of learning I used throughout my research.

1.1.1 Supervised Learning

In supervised learning, the dataset is the collection of labeled examples $\{(\mathrm{x_i}, \mathrm{y_i})\}_{i=1}^{N}$ Each element x_i among N is called a feature vector. A feature vector is a vector in which each dimension j = 1, ..., D contains a value that describes the example somehow. That values is called a feature and is denoted as $x^{(j)}$. For intance, if each example x in our collection represents a person, the first feature, $x^{(1)}$,

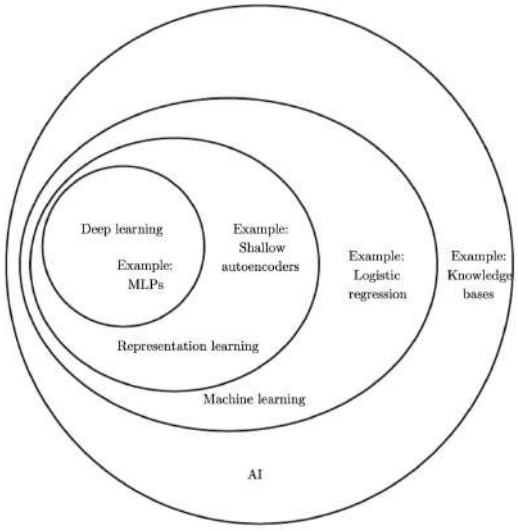


Figure 1.1 – A Venn diagram showing how deep learning is a kind of representation learning, which is in turn a kind of machine learning, which is used for many but not all approaches to AI. Each section of the Venn diagram includes an example of an AI technology (Goodfellow et al., 2016).

could contain height in cm, the second feature, $x^{(2)}$, could contain weight in kg, $x^{(3)}$ could contain gender, and so on. For all examples in the dataset, the feature at position *j* in the feature vector always contains the same kind of information. It means that if $\mathrm{x}_i^{(2)}$ contains weight in kg in some example x_i, then $\mathrm{x}_k^{(2)}$ will contain weight in kg in every example x_k, k = 1, ..., *N*. The label y_i can be either an element belonging to a finite set of classes 1, 2, ..., *C*, or a real number, or a more complex structure, like a vector, a matrix, a tree, or a graph.

The goal of a supervised learning is to use the dataset to produce a model that takes a feature vector x as input and outputs information that allows deducing the label for this feature vector. For instance, the model created using the dataset of people could take as input a feature vector describing a person and output a probability that the person has cancer (Burkov, 2019).

1.1.2 Parameters vs. Hyperparameters

A hyperparameter is a property of a learning algorithm, usually (but not always) having a numerical value. That value control the learning process and determine the values of model parameters that a learning algorithm ends up learning (Burkov, 2019). Hyperparameters are not learned by the algorithm itself from data. They have to be set by the data analyst before running the algorithm. Some common examples are : train-test split ratio, choice of optimization algorithm, learning rate in optimization algorithms, choice of activation function in a neural network, the choice of cost or loss function the model will use, etc.

Parameters on the other hand are internal to the model. That is, they are learned or estimated purely from the data during training as the algorithm used tries to learn the mapping between the input features and the labels or targets. Model training typically starts with parameters being initialized to some values (random values or set to zeros). As training/learning progresses the initial values are updated using an optimization algorithm (e.g., gradient descent). The learning algorithm is continuously updating the parameter values as learning progress. At the end of the learning process, model parameters are what constitute the model itself. Examples of parameters : The coefficients of linear and logistic regression models and weights and biases of a neural network.

1.1.3 Classification vs. Regression

In machine learning, the classification problem is solved by a classification learning algorithm that takes a collection of labeled examples as inputs and produces a model that can take an unlabeled example as input and either directly output a label or output a number that can be used by the analyst to deduce the label. An example of such a number is a probability. In a classification problem, a label is a member of a finite set of classes. If the size of the set of classes is two ("sick"/"healthy", "spam"/"not_spam"), we talk about binary classification (also called binomial in some sources). Multiclass classification (also called multinomial) is a classification problem with three or more classes.

Regression is a problem of predicting a real-valued label (often called a target) given an unlabeled example. Estimating house price valuation based on house features, such as area, the number of bedrooms, location and so on is a famous example of regression. The regression problem is solved by a regression learning algorithm that takes a collection of labeled examples as inputs and produces a model that can take an unlabeled example as input and output a target (Burkov, 2019).

1.1.4 Generalization.

Generalization is the ability of a model, after being trained to digest new data and make accurate predictions. It is probably the most important element of an artificial intelligence project. To understand this, look at the plot in figure 1.2. If two classes are separable from one another by a decision boundary, then, obviously, examples that belong to each class are located in two different subspaces which the decision boundary creates. If the examples used for training were selected randomly, independently of one another, and following the same procedure, then, statistically, it is more likely that the new negative example will be located on the plot somewhere not too far from other negative examples. The same concerns the new positive example: it will likely come from the surroundings of other

positive examples. In such a case, the decision boundary will still, with high probability, separate well new positive and negative examples from one another. For other less likely situations, the model will make errors, but since these situations are less likely, the number of errors will probably be smaller than the number of correct predictions.

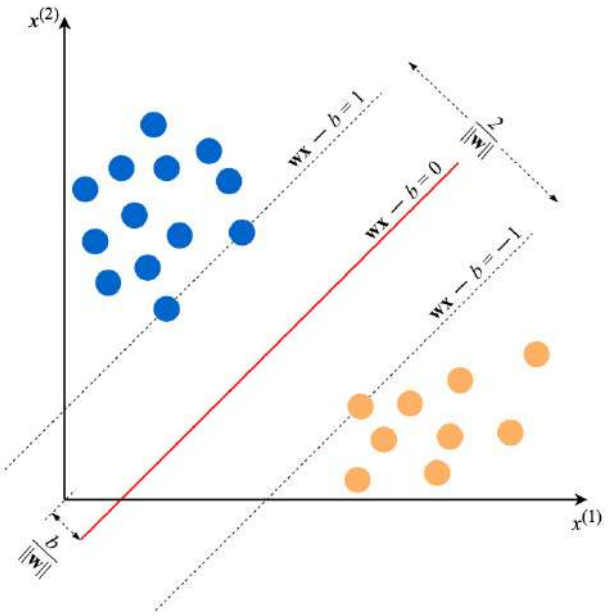


Figure 1.2 – An example of an Support Vector machine (SVM) model for two-dimensional feature vectors (Burkov, 2019).

Intuitively, the larger is the set of training examples, the more unlikely that the new examples will be dissimilar to (and lie on the plot far from) the examples used for training (Burkov, 2019).

1.1.5 Shallow vs. Deep Learning

A shallow learning algorithm learns the parameters of the model directly from the features of the training examples. Most supervised learning algorithms are shallow. The notorious exceptions are neural network learning algorithms, specifically those that build neural networks with more than one layer between input and output. Such neural networks are called deep neural networks. In deep neural network learning (or, simply, **deep learning**), contrary to shallow learning, most model parameters are learned not directly from the features of the training examples, but from the outputs of the preceding layers (Burkov, 2019).

1.1.6 Fundamental Algorithms

1.1.6.1 Linear Regression

We have a collection of labeled examples $\{(x_i, y_i)\}_{i=1}^{N}$, where N is the size of the collection, x_i is the D-dimensional feature vector of example i = 1, . . . , N, y_i is a real-valued target and every feature $x_i^{(j)}$, j = 1,...,D, is also a real number.

We want to build a model $f_{w,b}(x)$ as a linear combination of features of example x:

$$f_{w,b}(x) = wx + b \tag{1.1}$$

where w is a D-dimensional vector of parameters and b is a real number. The notation $f_{w,b}(x)$ means that the model f is parametrized by two values: w and b.

We will use the model to predict the unknown y for a given x like this: $y \leftarrow f_{w,b}(x)$. Two models parametrized by two different pairs (w, b) will likely produce two different predictions when applied to the same example. We want to find the optimal values (w^*,b^*). Obviously, the optimal values of parameters define the model that makes the most accurate predictions.

The hyperplane in linear regression is chosen to be as close to all training examples as possible. This requirement is essential by looking at the illustration in figure 1.3. It displays the regression line (in red) for one-dimensional examples (blue dots). We can use this line to predict the value of the target y_{new} for a new unlabeled input example x_{new}. If our examples are D-dimensional feature vectors (for D > 1), the only difference with the one-dimensional case is that the regression model is not a line but a plane (for two dimensions) or a hyperplane (for D > 2).

To get this requirement satisfied, the optimization procedure which we use to find the optimal values for w^* and b^* tries to minimize the following expression:

$$\frac{1}{N} \sum_{i=1..N} (f_{w,b}(x_i) - y_i)^2 \tag{1.2}$$

In mathematics, the expression we minimize or maximize is called an *objective function*, or, simply, an objective. The expression $(f_{w,b}(x_i) - y_i)^2$ in the above objective is called the **loss function**. It is a measure of penalty for misclassification of example i. This particular choice of the loss function is called **squared error loss**. All model-based learning algorithms have a loss function and what we

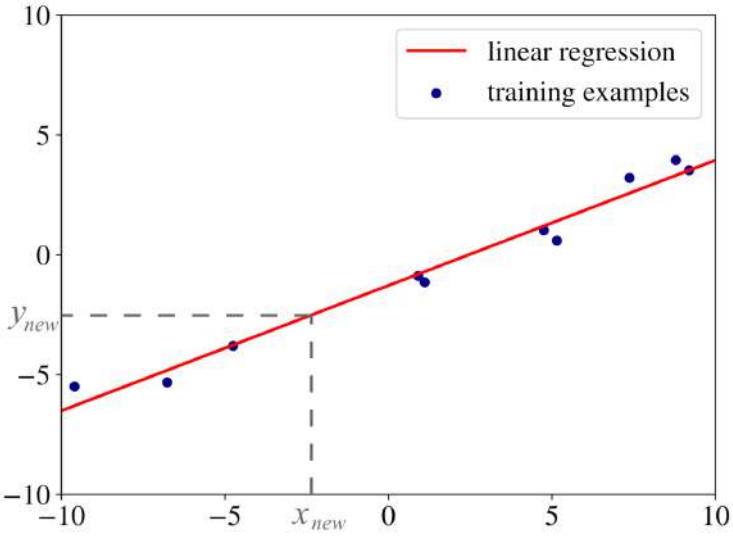


Figure 1.3 – Linear Regression for one-dimensional examples (Burkov, 2019).

do to find the best model is we try to minimize the objective known as the **cost function**. In linear regression, the cost function is given by the average loss, also called the *empirical risk*. The average loss, or empirical risk, for a model, is the average of all penalties obtained by applying the model to the training data.

One practical justification of the choice of the linear form for the model is that it rarely *overfit*. Overfitting is the property of a model such that the model predicts very well labels of the examples used during training but frequently makes errors when applied to examples that were not seen by the learning algorithm during training.

An example of overfitting in regression is shown in figure 1.4. The data used to build the red regression line is the same as in figure 1.3. The difference is that this time, this is the polynomial regression with a polynomial of degree 10. The regression line predicts almost perfectly the targets from almost all training examples, but will likely make significant errors on new data, as you can see for x_{new}.

1.1.7 Feature Engineering

The problem of transforming raw data into a dataset is called *feature engineering*. For most practical problems, *feature engineering* is a labor-intensive process that demands from the data analyst a lot of creativity and, preferably, domain knowledge. The role of the data analyst is to create informative features: those would allow the learning algorithm to build a model that does a good job of predicting labels of the data used for training. Highly informative features are also called features with high predictive power (Burkov, 2019).

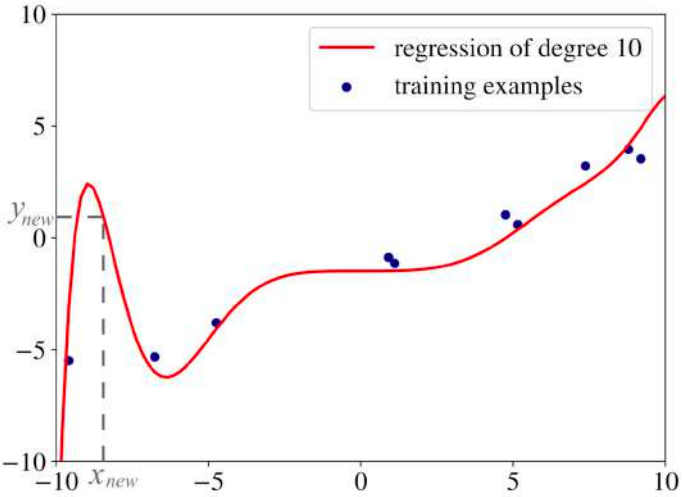


Figure 1.4 – Example of overfitting (Burkov, 2019)

1.1.7.1 Normalization

Normalization is the process of converting an actual range of values which a numerical feature can take, into a standard range of values, typically in the interval [-1, 1] or [0, 1]. More generally, the normalization formula looks like this:

$$\overline{x}^{(j)} = \frac{x^{(j)} - min^{(j)}}{max^{(j)} - min^{(j)}} \tag{1.3}$$

where $min^{(j)}$ and $max^{(j)}$ are, respectively, the minimum and the maximum value of the feature j in the dataset.

Normalizing the data is not a strict requirement. However, in practice, it can lead to an increased speed of learning. Imagine we have a two-dimensional feature vector. When you update the parameters of $w^{(1)}$ and $w^{(2)}$, you use partial derivatives of the mean squared error with respect to $w^{(1)}$ and $w^{(2)}$. If $x^{(1)}$ is in the range [0, 1000] and $x^{(2)}$ the range [0, 0.0001], then the derivative with respect to a larger feature will dominate the update during backpropagation (Burkov, 2019).

1.1.7.2 Standardization

Standardization is the procedure during which the feature values are rescaled so that they have the properties of a standard normal distribution with $\mu = 0$ and $\sigma = 0$, where μ is the mean (the average value of the feature, averaged over all examples in the dataset) and σ is the standard deviation from the mean (Burkov, 2019).

Standard scores of features are calculated as follows:

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{max^{(j)} - \sigma^{(j)}} \tag{1.4}$$

1.1.7.3 Splitting machine learning data

In practice, data analysts work with three distinct sets of labeled examples:

1. training set.
2. validation set.
3. test set.

Once the labeled dataset is prepared, the first thing to do is to shuffle the examples and split the dataset into three subsets: training, validation, and test. The training set is usually the biggest one; it is use to build the model. The validation and test sets are roughly the same sizes, much smaller than the size of the training set. The learning algorithm cannot use examples from the test set to build the model.

There is no optimal proportion to split the dataset into these three subsets. In the past, the rule of thumb was to use 70% of the dataset for training, 15% for validation and 15% for testing. However, in the age of big data, datasets often have millions of examples. In such cases, it could be reasonable to keep 95% for training and 2.5% for validation and 2.5% for testing.

The interest of splitting the data set is the following: when we build a model, what we do not want is for the model to only do well at predicting labels of examples the learning algorithm has already seen. A trivial algorithm that simply memorizes all training examples and then uses the memory to "predict" their labels will make no mistakes when asked to predict the labels of the training examples, but such an algorithm would be useless in practice. What we really want is a model that is good at predicting examples that the learning algorithm did not see: we want good performance on a holdout set (Burkov, 2019).

1.1.7.4 Underfitting and Overfitting

A model has a low bias if it predicts well the labels of the training data. If the model makes many mistakes on the training data, we say that the model has a high bias or that the model underfits. So,

underfitting is the inability of the model to predict well the labels of the data it was trained on. There could be several reasons for underfitting, the most important of which are:

1. the model is too simple for the data (for example a linear model can often underfit);
2. the features engineered are not informative enough.

The first reason is easy to illustrate in the case of one-dimensional regression: the dataset can resemble a curved line, but our model is a straight line. The second reason can be illustrated like this: let's say you want to predict whether a patient has cancer, and the features you have are height, blood pressure, and heart rate. These three features are clearly not good predictors for cancer so the model will not be able to learn a meaningful relationship between these features and the label.

The solution to the problem of underfitting is to try a more complex model or to engineer features with higher predictive power.

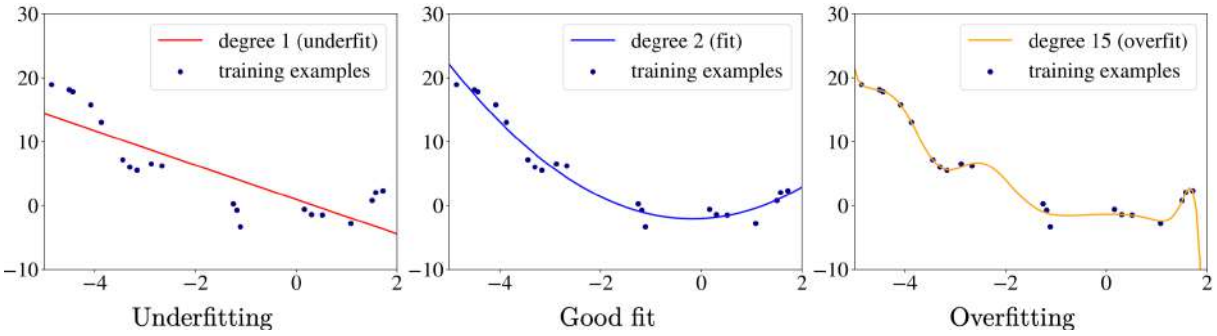


Figure 1.5 – Examples of underfitting (linear model), good fit (quadratic model), and overfitting (polynomial of degree 15) (Burkov, 2019).

Overfitting is another problem a model can exhibit. The model that overfits predicts very well the training data but poorly the data from at least one of the two holdout sets. Several reasons can lead to overfitting, the most important of which are:

1. The model is too complex for the data (e.g., a very deep or wide neural network often overfit);
2. A small number of training examples are considered for too many features .

Many solutions to the overfitting are possible:

1. Try a simpler model (e.g., a neural network with fewer layers/units).
2. Reduce the dimensionality of examples in the dataset.
3. Add more training data, if possible.
4. Regularize the model.

1.1.7.5 Regularization

Regularization is an umbrella term that encompasses methods that force the learning algorithm to build a less complex model. In practice, that often leads to slightly higher bias but significantly reduces the variance. This problem is known in the literature as the bias-variance tradeoff.

The two most widely used types of regularization are called *L1* and *L2* regularization. The idea is quite simple. To create a regularized model, the objective function is modified by adding a penalizing term whose value is higher when the model is more complex.

Recall the linear regression objective:

$$min_{w,b}\frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x_i)-y_i)^2 \tag{1.5}$$

An L1-regularized objective looks like this:

$$min_{w,b}\left[C|w|+\frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x_i)-y_i)^2\right] \tag{1.6}$$

where $|\mathbf{w}| = \sum_{j=1}^{D}\left|w^{(j)}\right|$ and C is a hyperparameter that controls the importance of regularization. If we set C to zero, the model becomes a standard non-regularized linear regression model. On the other hand, if we set to C to a high value, the learning algorithm will try to set most $w^{(j)}$ to a very small value or zero to minimize the objective, and the model will become very simple which can lead to underfitting.

An L2-regularized objective looks like this:

$$min_{\mathbf{w},b}\left[C||\mathbf{w}||^2+\frac{1}{N}\sum_{i=1}^{N}(f_{\mathbf{w},b}(x_i)-y_i)^2\right], where||\mathbf{w}||^2=\sum_{j=1}^{D}(w^{(j)})^2 \tag{1.7}$$

In practice, *L1* regularization produces a sparse model, a model that has most of its parameters (in case of linear models, most of $w^{(j)}$) equal to zero, provided the hyperparameter C is large enough. So *L1* performs feature selection by deciding which features are essential for prediction and which are not. That can be useful in case you want to increase model explainability. However, the goal is to maximize the performance of the model on the test data, then *L2* usually gives better results. *L2* also has the advantage of being differentiable, so gradient descent can be used for optimizing the objective function.

In the literature, the name ridge regularization can be found for L2 and lasso for L1.

Neural networks also benefit from two other regularization techniques: dropout and batch- normalization. There are also non-mathematical methods that have a regularization effect: data augmentation and early stopping (Burkov, 2019).

1.1.7.6 Model Performance Assessment

The test set contains the examples that the learning algorithm has never seen before, so if a model performs well on predicting the labels of the examples from the test set, we say that the model generalizes well or, simply, that it is good.

To be more rigorous, machine learning specialists use various formal metrics and tools to assess the model performance. For regression, the assessment of the model is quite simple. A well-fitting regression model results in predicted values close to the observed data values. If this is the case, the next step is to compare the performances of the model on the training and the test data. To do that, we compute the mean squared error (MSE) for the training, and, separately, for the test data. If the MSE of the model on the test data is substantially higher than the MSE obtained on the training data, this is a sign of overfitting. Regularization or a better hyperparameter tuning could solve the problem. The meaning of "substantially higher" depends on the problem at hand and has to be decided by the data analyst. For classification, the most widely used metrics and tools to assess the classification model are:

1. Confusion matrix,
2. Accuracy,
3. Precision/recall,
4. area under the ROC curve.

1.1.7.7 Hyperparameter Tuning

Hyperparameters are not optimized by the learning algorithm itself. The data analyst has to "tune" hyperparameters by experimentally finding the best combination of values, one per hyperparameter.

One typical way to do that, when a reasonable amount of data is available to obtain a decent validation set and the number of hyperparameters and their range are not too large, is to use *grid search*. Grid search is the most simple hyperparameter tuning technique. It is great for spot-checking combinations that are known to perform well generally. A well-known disadvantage is that testing all

combinations of hyperparameters, especially if there are more than two, can be very time consuming, especially for large data sets. There are more efficient techniques, such as random search and Bayesian hyperparameter optimization.

Random search differs from grid search in that a discrete set of values is not provide to explore for each hyperparameter; instead, a statistical distribution is provide for each hyperparameter from which values are randomly sampled. It is also possible to set the total number of combinations to try.

Bayesian techniques differ from random or grid search in that they use past evaluation results to choose the next values to evaluate. The idea is to limit the number of expensive optimizations of the objective function by choosing the next hyperparameter values based on those that have done well in the past (Burkov, 2019).

1.2 Neural Networks

A neural network (NN), just like a regression or an SVM model, is a mathematical function:

$$y = f_{NN}(x) \tag{1.8}$$

The function f_{NN} is a nested function. In literature, the term of neural network layers is employed for this particular form. So, for a 3-layer neural network that returns a scalar, f_{NN} looks like this:

$$y = f_{NN}(x) = f_3(f_2(f_{1(x)})) \tag{1.9}$$

In the above equation, f_1 and f_2 are vector functions of the following form:

$$f_1(z) = g_l(\mathbf{W}_l z + \mathbf{b}_l) \tag{1.10}$$

where l is called the layer index and can span from 1 to any number of layers. The function g_l is called an activation function. It is a fixed, usually nonlinear function chosen by the data analyst before the learning is started. The parameters $\mathbf{W}_l$ (a matrix) and $\mathbf{b}_l$ (a vector) for each layer are learned using the familiar gradient descent by optimizing, depending on the task, a particular cost function (such as MSE). The function f_3 is a scalar function for the regression task, but can also be a vector function depending on the problem.

1.2.1 Multilayer Perceptron

We have a closer look at one particular configuration of neural networks called feed-forward neural networks (FFNN), and more specifically the architecture called a multilayer perceptron (MLP). As an illustration, let's consider an MLP with three layers. The network takes a two-dimensional feature vector as input and outputs a number. This FFNN can be a regression or a classification model, depending on the activation function used in the third, output layer.

The MLP is depicted in figure 1.6. The neural network is represented graphically as a connected combination of units logically organized into one or more layers. Each unit is represented by either a circle or a rectangle. The inbound arrow represents an input of a unit and indicates where this input came from. The outbound arrow indicates the output of a unit. The output of each unit is the result of the mathematical operation written inside the rectangle. Circle units do not do anything with the input; they just send their input directly to the output.

The following happens in each rectangle unit. Firstly, all inputs of the unit are joined together to form an input vector. Then the unit applies a linear transformation to the input vector, exactly like linear regression model does with its input feature vector. Finally, the unit applies an activation function g to the result of the linear transformation and obtains the output value, a real number. In a FFNN, the output value of a unit of some layer becomes an input value of each of the units of the subsequent layer.

In figure 1.6 the activation function g_l has one index: l, the index of the layer the unit belongs to. Usually, all units of a layer use the same activation function, but it is not a rule. Each layer can have a different number of units. Each unit has its parameters $\mathbf{w}_{l,u}$ and $b_{l,u}$ where u is the index of the unit, and l is the index of the layer. The vector $\mathbf{y}_{l-1}$ in each unit is defined as $[\mathrm{y}^{(1)}_{l-1}, \mathrm{y}^{(2)}_{l-1}, \mathrm{y}^{(3)}_{l-1}, \mathrm{y}^{(4)}_{l-1}]$. The vector $\mathbf{x}$ in the first layer is defined as $[x^{(1)}, \ldots, x^{(D)}]$.

As you can see in figure 1.6, in multilayer perceptron all outputs of one layer are connected to each input of the succeeding layer. This architecture is called fully-connected. A neural network can contain fully-connected layers. Those are the layers whose units receive as inputs the outputs of each of the units of the previous layer.

1.2.2 Feed-Forward Neural Network Architecture

If we want to solve a regression or a classification problem discussed previously, the last (the rightmost) layer of a neural network usually contains only one unit. If the activation function g_{last} of the

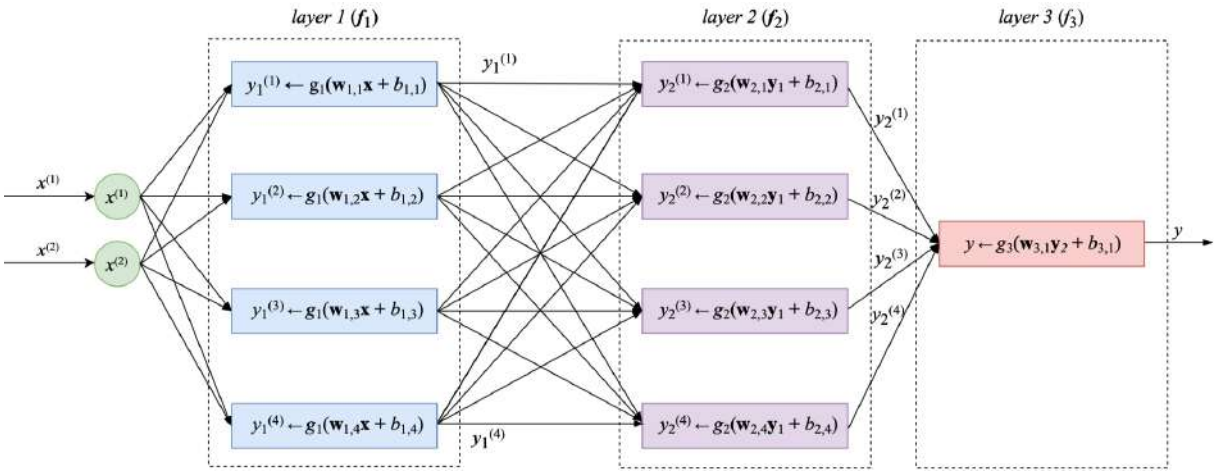


Figure 1.6 – A multilayer perceptron with two-dimensional input, two layers with four units and one output layer with one unit (Burkov, 2019).

last unit is linear, then the neural network is a regression model. If the g_{last} is a logistic function, the neural network is a binary classification model.

The data analyst can choose any mathematical function as $g_{l,u}$ assuming it is differentiable. The latter property is essential for gradient descent used to find the values of the parameters $\mathbf{w}_{l,u}$ and $\mathbf{b}_{l,u}$ for all l and u. The primary purpose of having nonlinear components in the function f_{NN} is to allow the neural network to approximate nonlinear functions. Without nonlinearities, f_{NN} would be linear, no matter how many layers it has. The reason is that $\mathbf{W}_l\mathbf{z} + \mathbf{b}_l$ is a linear function and a linear function of a linear function is also linear.

Popular choices of activation functions are the logistic function, as well as **TanH** and **ReLU**. The former is the hyperbolic tangent function, similar to the logistic function but ranging from -1 to 1 (without reaching them). The latter is the rectified linear unit function, which equals to zero when its input z is negative and to z otherwise:

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \tag{1.11}$$

$$relu(z) = \begin{cases} 0 & if\ z < 0 \\ z & otherwise \end{cases} \tag{1.12}$$

As mentioned before, $\mathbf{W}_l$ in the expression $\mathbf{W}_l\mathbf{z} + \mathbf{b}_l$, is a matrix, while $\mathbf{b}_l$ is a vector. That looks different from linear regression's $\mathbf{wz} + b$. In matrix $\mathbf{W}_l$, each row u corresponds to a vector of parameters $\mathbf{w}_{l,u}$. The dimensionality of the vector $\mathbf{w}_{l,u}$ equals to the number of units in the layer *l-1*. The operation $\mathbf{W}_l\mathbf{z}$ results in a vector $\mathbf{a}_l = [\mathbf{w}_{l,1}\mathbf{z}, \mathbf{w}_{l,2}\mathbf{z}, ..., \mathbf{w}_{l,size_l}\mathbf{z}]$. Then the sum $\mathbf{a}_l + \mathbf{b}_l$ gives a

$size_l - dimensional$ vector $\mathbf{c}_l$. Finally, the function $\mathbf{g}_l(\mathbf{c}_l)$ produces the vector $\mathbf{y}_l = [y_l^{(1)}, y_l^{(2)}, ..., y_l^{(size_l)}]$ as output.

1.2.3 Deep Learning

Deep learning refers to training neural networks with more than two non-output layers. In the past, it became more difficult to train such networks as the number of layers grew. The two biggest challenges were referred to as the problems of exploding gradient and vanishing gradient as gradient descent was used to train the network parameters.

What is vanishing gradient and why does it arise? To update the values of the parameters in neural networks the algorithm called backpropagation is typically used. Backpropagation is an efficient algorithm for computing gradients on neural networks using the chain rule. During gradient descent, the neural network's parameters receive an update proportional to the partial derivative of the cost function with respect to the current parameter in each iteration of training. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing some parameters from changing their value. In the worst case, this may completely stop the neural network from further training.

Traditional activation functions, such as the hyperbolic tangent function, have gradients in the range (0, 1), and backpropagation computes gradients by the chain rule. That has the effect of multiplying n of these small numbers to compute gradients of the earlier (leftmost) layers in an n-layer network, meaning that the gradient decreases exponentially with n. That results in the effect that the earlier layers train very slowly, if at all.

However, the modern implementations of neural network learning algorithms allow you to effectively train very deep neural networks (up to hundreds of layers). This is due to several improvements combined together, including ReLU, LSTM, as well as techniques such as skip connections used in residual neural networks, as well as advanced modifications of the gradient descent algorithm.

Therefore, today, since the problems of vanishing and exploding gradient are mostly solved (or their effect diminished) to a great extent, the term "deep learning" refers to training neural networks using the modern algorithmic and mathematical toolkit independently of how deep the neural network is. In practice, many business problems can be solved with neural networks having 2-3 layers between the input and output layers. The layers that are neither input nor output are often called *hidden layers*.

1.2.4 Transfer learning

Humans have an inherent ability to transfer knowledge across tasks. What we acquire as knowledge while learning about one task, we utilize in the same way to solve related tasks. The more related the tasks, the easier it is for us to transfer, or cross-utilize, knowledge. Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones.

Let's understand the preceding definition with the help of an example. Assume that the task is to identify objects in images within a restricted domain of a restaurant. We mark this task in its scope defined as T_1. Given the dataset for this task, we train a model and tune it to perform well (generalize) on unseen data points from the same domain (restaurant).Traditional supervised ML algorithms break down when we do not have sufficient training examples for the required tasks in given domains. Suppose we now must detect objects from images in a park or a café (task T_2). Ideally, we should be able to apply the model trained for T_1, but in reality we face performance degradation and models that do not generalize well. This happens for a variety of reasons, which we can liberally and collectively term as the model's bias toward training data and domain. Transfer learning thus enables us to utilize knowledge from previously learned tasks and apply them to newer, related ones. If we have significantly more data for task T_1, we may utilize its learnings and generalize them for task T_2, (which has significantly less data). In the case of image classification, certain low-level features, such as edges, shapes, and lighting, can be shared across tasks and thus enable knowledge transfer among tasks (Sarkar et al., 2018).

Apart from providing capabilities to reuse already-built models, transfer learning may assist learning the target task in the following ways:

- **Improved baseline performance**: When we augment the knowledge of an isolated learner (also known as an ignorant learner) with knowledge from a source model, the baseline performance might improve due to this knowledge transfer.
- **Model-development time**: Utilizing knowledge from a source model might also help in fully learning the target task, as compared to a target model that learns from scratch. This, in turn, results in improvements in the overall time taken to develop/learn a model.
- **Improved final performance**: Higher final performance might be attained by leveraging transfer learning.

1.2.5 Autoencoders

Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent-space representation.

An autoencoder consists of 3 components: encoder, code and decoder. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code.

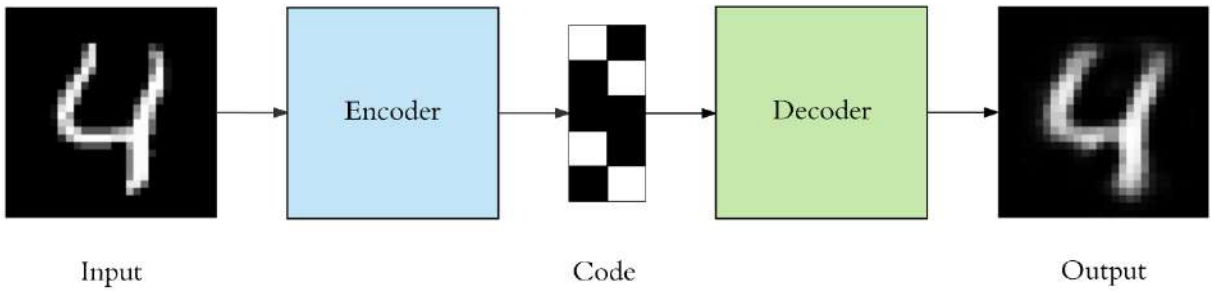


Figure 1.7 – Auto encoder example (Burkov, 2019).

Autoencoders are mainly a dimensionality reduction (or compression) algorithm with a couple of important properties:

1. Data-specific: Autoencoders are only able to meaningfully compress data similar to what they have been trained on. Since they learn features specific for the given training data, they are different than a standard data compression algorithm like gzip. So we can't expect an autoencoder trained on handwritten digits to compress landscape photos.
2. Lossy: The output of the autoencoder will not be exactly the same as the input, it will be a close but degraded representation. If you want lossless compression they are not the way to go.
3. Unsupervised: To train an autoencoder we do not need to do anything fancy, just throw the raw input data at it. Autoencoders are considered an unsupervised learning technique since they do not need explicit labels to train on. But to be more precise they are self-supervised because they generate their own labels from the training data.

First the input passes through the encoder, which is a fully-connected ANN, to produce the code. The decoder, which has the similar ANN structure, then produces the output only using the code. The goal is to get an output identical with the input. Note that the decoder architecture is the mirror image of the encoder. This is not a requirement but it is typically the case. The only requirement is the

dimensionality of the input and output needs to be the same. Anything in the middle can be played with.

1.2.6 Convolutional Neural Network

The number of parameters an MLP can have grows very fast as you make your network bigger. More specifically, as you add one layer, you add $(size_{l-1}+1).size_l$ parameters (our matrix $\mathbf{W}_l$ plus the vector $\mathbf{b}_l$). That means that if you add another 1000-unit layer to an existing neural network, then you add more than 1 million additional parameters to your model. Optimizing such big models is a very computationally intensive problem.

When the training examples are images, the input is very high-dimensional. If you want to learn to classify images using an MLP, the optimization problem is likely to become intractable.

A convolutional neural network (CNN) is a special kind of FFNN that significantly reduces the number of parameters in a deep neural network with many units without losing too much in the quality of the model. CNNs have found applications in image and text processing where they beat many previously established benchmarks.

You may have noticed that in images, pixels that are close to one another usually represent the same type of information: sky, water, leaves, fur, bricks, and so on. The exception from the rule are the edges: the parts of an image where two different objects "touch" one another.

If we can train the neural network to recognize regions of the same information as well as the edges, this knowledge would allow the neural network to predict the object represented in the image. For example, if the neural network detected multiple skin regions and edges that look like parts of an oval with skin-like tone on the inside and bluish tone on the outside, then it is likely that it is a face on the sky background. If our goal is to detect people on pictures, the neural network will most likely succeed in predicting a person in this picture.

Having in mind that the most important information in the image is local, we can split the image into square patches using a moving window approach. We can then train multiple smaller regression models at once, each small regression model receiving a square patch as input. The goal of each small regression model is to learn to detect a specific kind of pattern in the input patch. For example, one small regression model will learn to detect the sky; another one will detect the grass, the third one will detect edges of a building, and so on (Burkov, 2019).

In CNNs, a small regression model looks like the one in figure 1.6, but it only has the layer 1 and does

not have layers 2 and 3. To detect some pattern, a small regression model has to learn the parameters of a matrix **F** (for "filter") of size $p \times p$, where p is the size of a patch. Let's assume, for simplicity, that the input image is black and white, with 1 representing black and 0 representing white pixels. Assume also that the patches are 3 by 3 pixels ($p = 3$). Some patch could then look like the following matrix **P** (for "patch"):

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{1.13}$$

The above patch represents a pattern that looks like a cross. The small regression model that will detect such patterns (and only them) would need to learn a 3 by 3 parameter matrix **F** where parameters at positions corresponding to the 1s in the input patch would be positive numbers, while the parameters in positions corresponding to 0s would be close to zero. If we calculate the convolution of matrices **P** and **F**, the value we obtain is higher the more similar **F** is to **P** (Burkov, 2019). To illustrate the convolution of two matrices, assume that **F** looks like this:

$$\mathbf{F} = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 4 & 1 \\ 0 & 3 & 0 \end{bmatrix} \tag{1.14}$$

Then convolution operator is only defined for matrices that have the same number of rows and columns. For our matrices of **P** and **F** it's calculated as illustrated below:

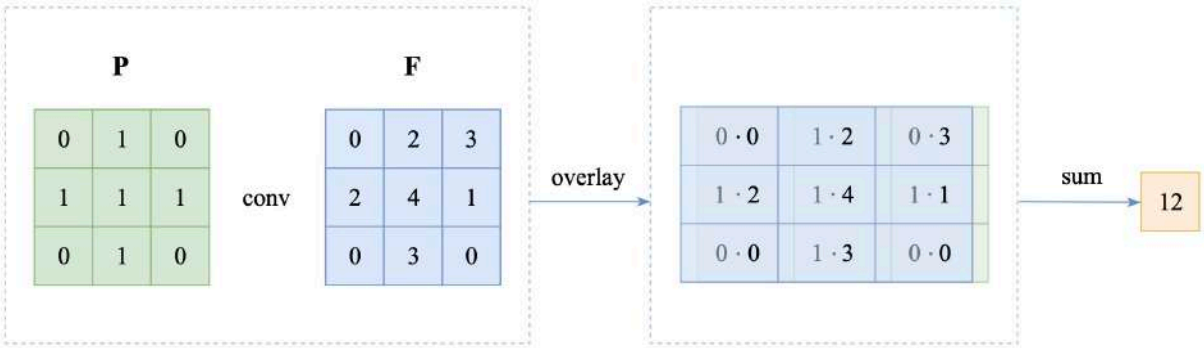


Figure 2: A convolution between two matrices.

Figure 1.8 – A convolution between two matrices (Burkov, 2019).

If the input patch **P** had a different pattern, for example, that of a letter L,

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{1.15}$$

then the convolution with **F** would give a lower result: 5. So, you can see the more the patch "looks" like the filter, the higher the value of the convolution operation is. For convenience, there's also a bias parameter b associated with each filter **F** which is added to the result of a convolution before applying the nonlinearity (activation function).

One layer of a CNN consists of multiple convolution filters (each with its own bias parameter), just like one layer in a vanilla FFNN consists of multiple units. Each filter of the first (leftmost) layer slides — or *convolves* — across the input image, left to right, top to bottom, and convolution is computed at each iteration.

An illustration of the process is given in Figure 1.9 where 6 steps of one filter convolving across an image are shown.

The filter matrix (one for each filter in each layer) and bias values are trainable parameters that are optimized using gradient descent with backpropagation. A nonlinearity is applied to the sum of the convolution and the bias term. Typically, the ReLU activation function is used in all hidden layers. The activation function of the output layer depends on the task. Since we can have $size_l$ filters in each layer l, the output of the convolution layer l would consist of $size_l$ matrices, one for each filter.

If the CNN has one convolution layer following another convolution layer, then the subsequent layer $l+1$ treats the output of the preceding layer l as a collection of $size_l$ image matrices. Such a collection is called a volume. The size of that collection is called the volume's depth. Each filter of layer $l+1$ convolves the whole volume. The convolution of a patch of a volume is simply the sum of convolutions of the corresponding patches of individual matrices the volume consists of.

An example of a convolution of a patch of a volume consisting of depth 3 is shown in Figure 1.10. The value of the convolution, -3, was obtained as $(-3\cdot 3+3\cdot 1+5\cdot 4+-1\cdot 1)+(-2\cdot 2+3\cdot(-1)+5\cdot(-3)+-1\cdot 1)+(-2\cdot 1+3\cdot(-1)+5\cdot 2+-1\cdot(-1))+(-2)$.

In computer vision, CNNs often get volumes as input, since an image is usually represented by three channels: R, G, and B, each channel being a monochrome picture (Burkov, 2019).

Two important properties of convolution are stride and padding. Stride is the step size of the moving window. In figure 1.9, the stride is 1, that is the filter slides to the right and to the bottom by one cell

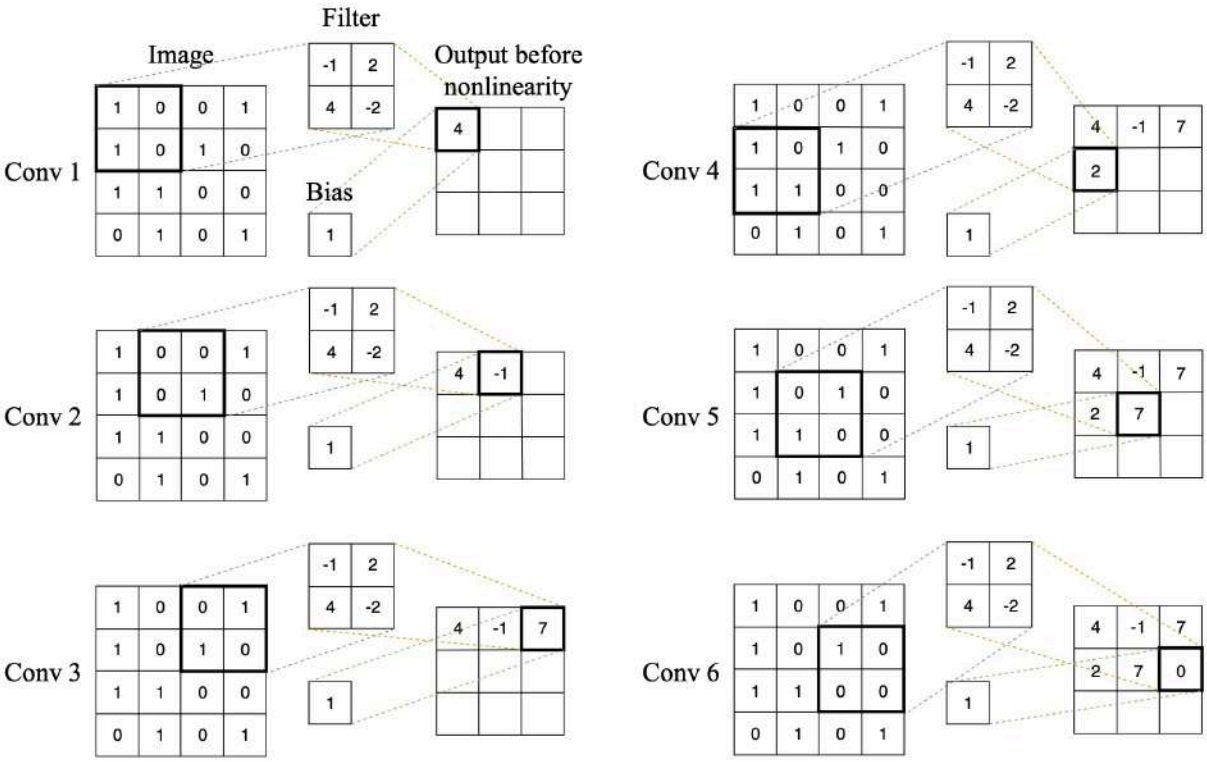


Figure 1.9 – A filter convolving across an image (Burkov, 2019).

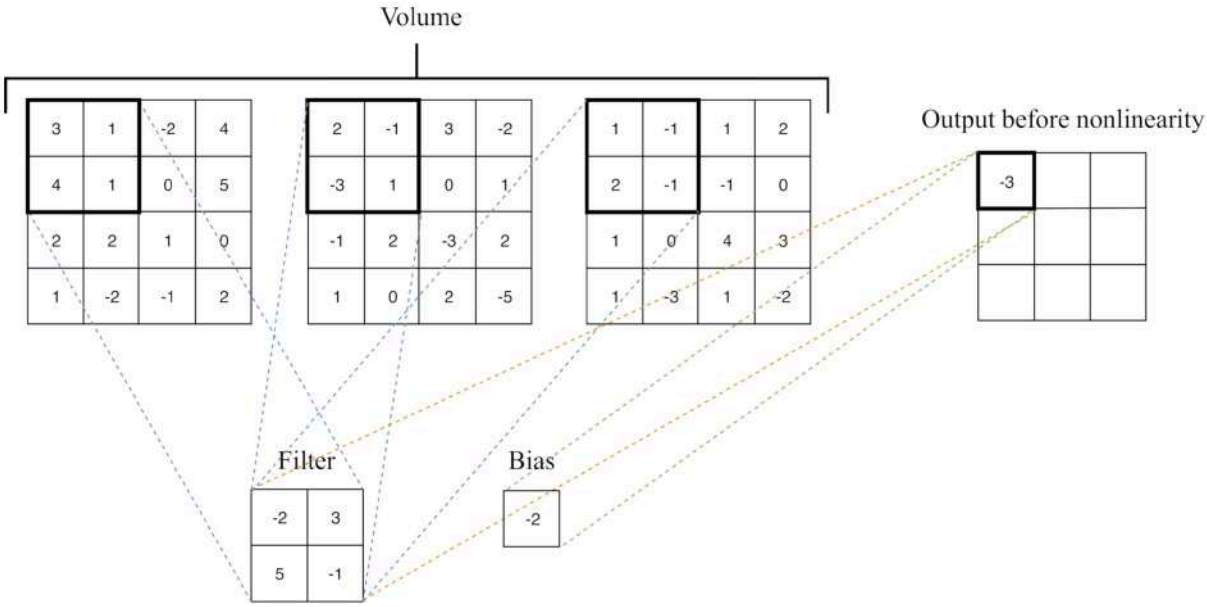


Figure 1.10 – Convolution of a volume consisting of three matrices (Burkov, 2019).

at a time. In figure 1.11, you can see a partial example of convolution with stride 2. You can see that the output matrix is smaller when stride is bigger.

Padding allows getting a larger output matrix; it is the width of the square of additional cells with which you surround the image (or volume) before you convolve it with the filter. The cells added by padding usually contain zeroes. In figure 1.9, the padding is 0, so no additional cells are added to the

image. In figure 1.12, on the other hand, the stride is 2 and padding is 1, so a square of width 1 of additional cells are added to the image. You can see that the output matrix is bigger when padding is bigger.

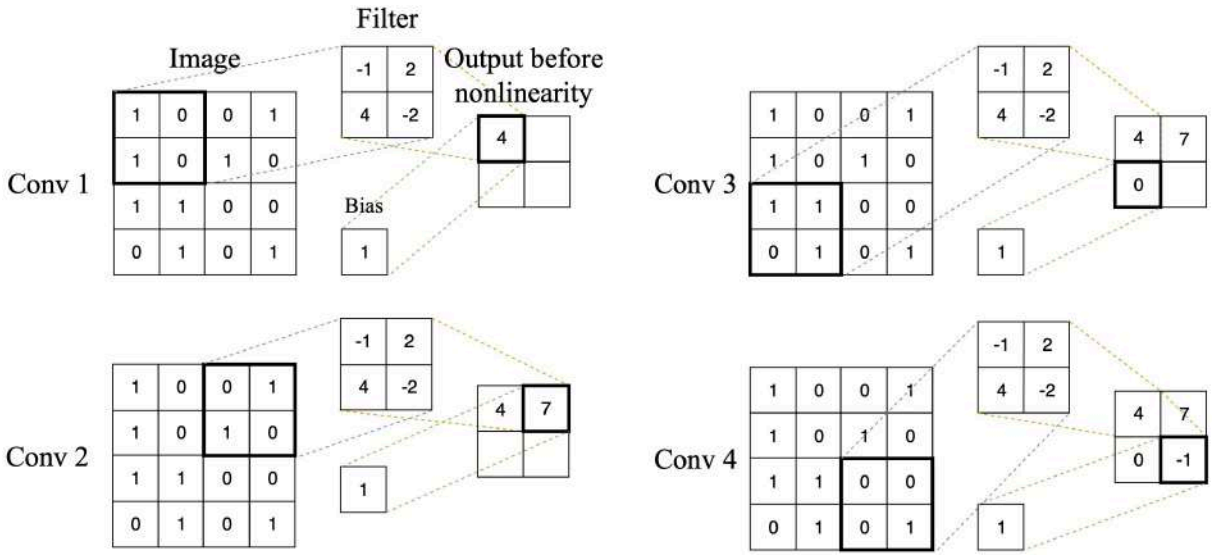


Figure 1.11 – Convolution with stride 2 (Burkov, 2019).

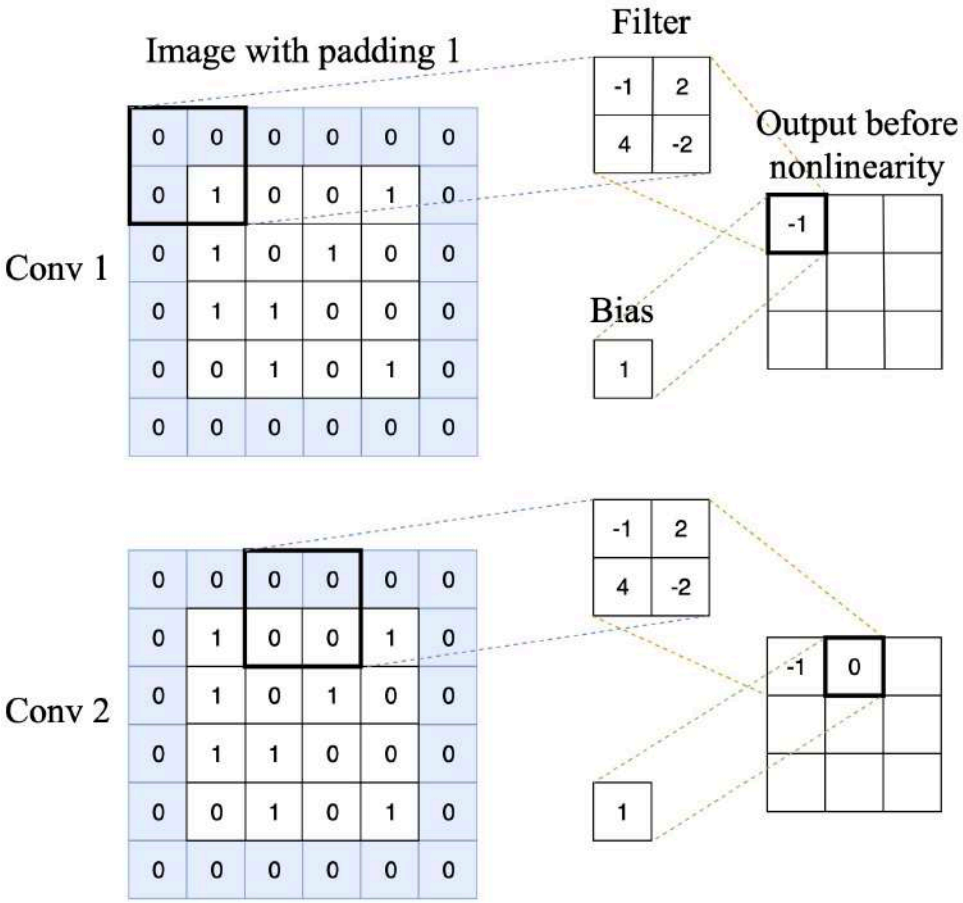


Figure 1.12 – Convolution with stride 2 and padding 1 (Burkov, 2019).

This section would not be complete without presenting pooling, a technique very often used in CNNs.

Pooling works in a way very similar to convolution, as a filter applied using a moving window approach. However, instead of applying a trainable filter to an input matrix or a volume, pooling layer applies a fixed operator, usually either max or average. Similarly to convolution, pooling has hyperparameters: the size of the filter and the stride.

Usually, a pooling layer follows a convolution layer, and it gets the output of convolution as input. When pooling is applied to a volume, each matrix in the volume is processed independently of others. Therefore, the output of the pooling layer applied to a volume is a volume of the same depth as the input.

As you can see, pooling only has hyperparameters and does not have parameters to learn. Typically, the filter of size 2 or 3 and stride 2 are used in practice. Max pooling is more popular than average and often gives better results (Burkov, 2019).

1.2.6.1 U-Net architecture

U-Net is an architecture for semantic segmentation (Figure 1.13). It consists of a contracting path and an expansive path. The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution (“up-convolution”) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes (Ronneberger et al., 2015).

The most commonly used evaluation parameter for image segmentation task is Dice score. The parameter is used in finding the similarity between the image samples and the predicted image by the architecture. The formula for Dice score is shown in Equation (1.16)

Dice score is defined as twice the area of overlap of the images (predicted and target image) divided by the total number of pixels in both the images. Since we are having binary output consisting of O and 1, the formula can also be written in another way, where area of overlap can be written as element-wise multiplication between the predicted and target image and sum of the total pixels divided by sum of the square of pixels in both the images (Kose et al., 2021).

$$Dice = \frac{2|P \cap T|}{|P| + |T|} = \frac{2|PT|}{|P|^2 + |T|^2} \tag{1.16}$$

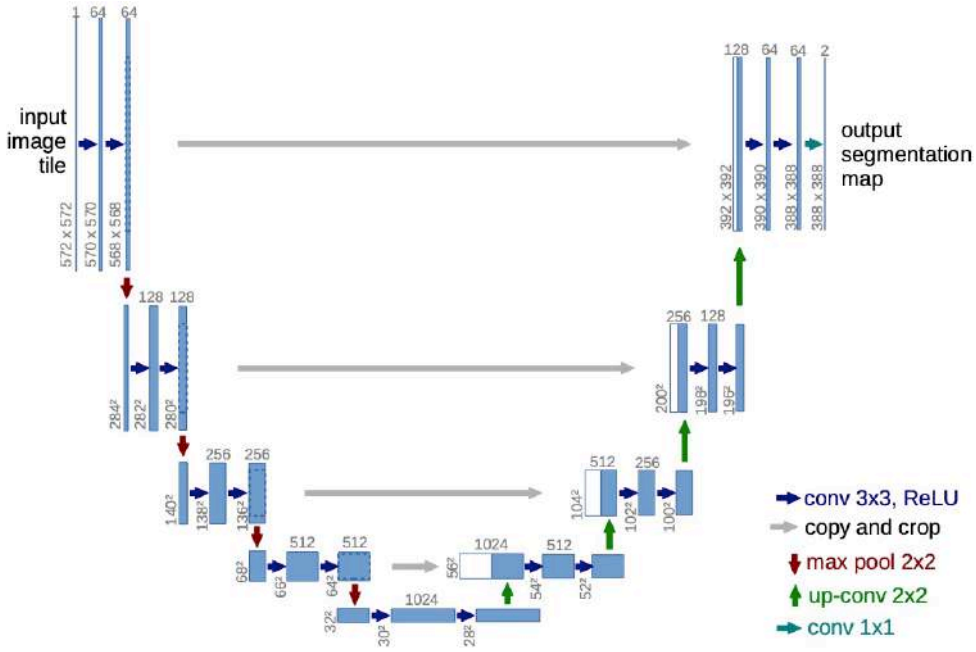


Figure 1.13 – Diagram of an U-Net architecture.

1.2.6.2 YOLO Architecture

YOLO (Redmon et al., 2016) is a new approach to object detection. Previous work on this approach reuses classifiers to perform detection. Instead, this approach views object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single convolutional network simultaneously predicts multiple bounding boxes and the class probabilities for those boxes. YOLO trains on complete images and directly optimizes detection performance.

This model divides the input image into a grid $S \times S$. If the center of an object is in a grid cell, that cell is responsible for detecting the object. Each grid cell predicts bounding boxes B and confidence scores for these boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurately it thinks the box predicts. Formally, YOLO defines confidence as $Pr(object) * IOU^{actual}_{predicts}$. Each bounding box consists of 5 predictions: x, y, w, h, and the confidence. The coordinates (x, y) represent the center of the box relative to the grid cell boundaries. The width and height are predicted relative to the entire image. Finally, the confidence prediction represents the IOU between the predicted box and any ground truth box (Figure 1.14).

This architecture takes an input image and resizes it to 448*448 keeping the same aspect ratio and performing a padding. This image is then passed through the CNN network. This model has 24

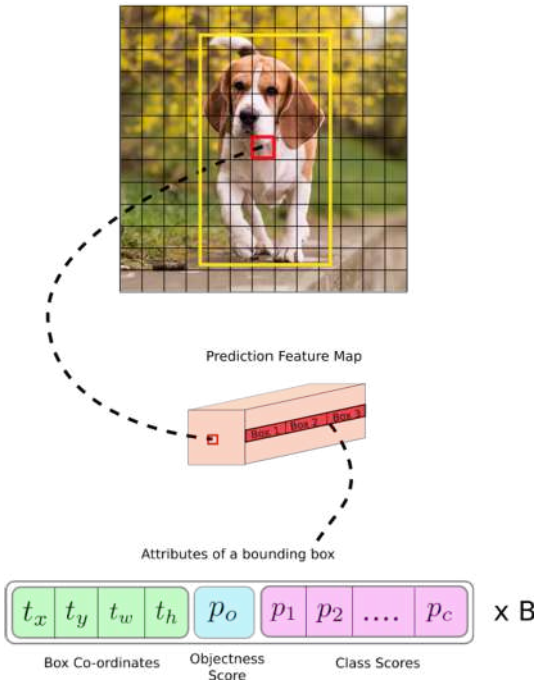


Figure 1.14 – Diagram of a bounding box.

convolution layers, 4 max-pooling layers followed by 2 fully connected layers (Figure 1.15). This architecture uses Leaky ReLU as the activation function throughout the architecture, except in the layer where it uses a linear activation function. Batch normalization is also used to regularize the model. The model is trained for a week and achieves an accuracy of 88% in the top 5 of the 2012 ImageNet validation, which is comparable to GoogLeNet (2014 ILSVRC winner), the leading model at that time. There is also a Fast YOLO version that uses fewer layers (9 instead of 24) and fewer filters.

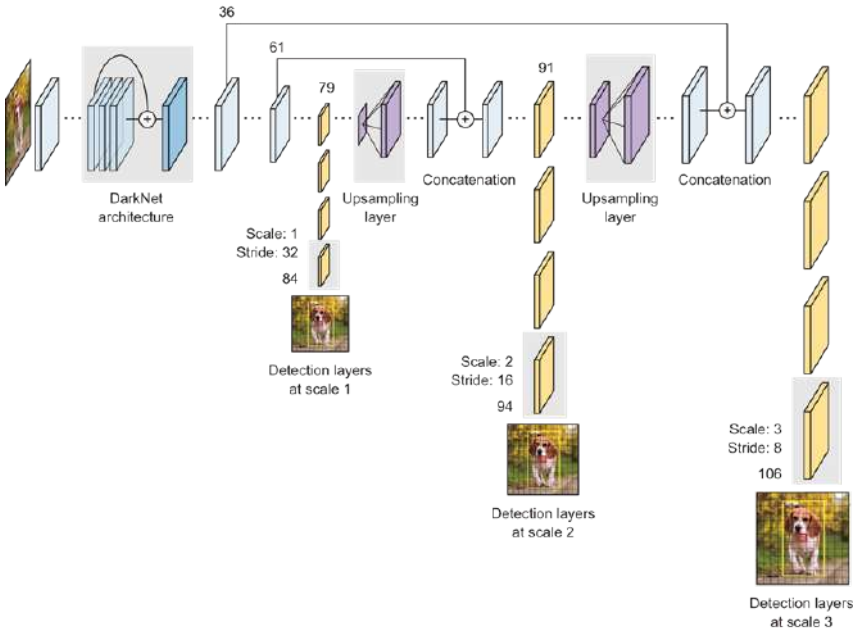


Figure 1.15 – Diagram of the YOLO architecture.

1.2.6.3 DenseNet architecture

Convolutional networks can be significantly deeper, more accurate, and more efficient to train if they contain shorter connections between layers near the input and those near the output. The dense convolutional network ("DenseNet") (Huang et al., 2018), connects each layer to all other layers in a feed-forward logic (Figure 1.16). While traditional L-layer convolutional networks have L connections - one between each layer and its subsequent layer - DenseNet has $\frac{L(L+1)}{2)}$ direct connections. For each layer, the feature maps of all previous layers are used as inputs, and its own feature maps are used as inputs in all subsequent layers (Figure 1.17).

Dense networks have several important advantages: they mitigate the vanishing gradient problem, enhance feature propagation, encourage feature reuse, and significantly reduce the number of parameters.

After several experiments, the DenseNet models tend to produce a consistent improvement in accuracy with an increasing number of parameters, with no signs of performance degradation or overfitting. Under several parameters, it achieved state-of-the-art results on several highly competitive data sets.

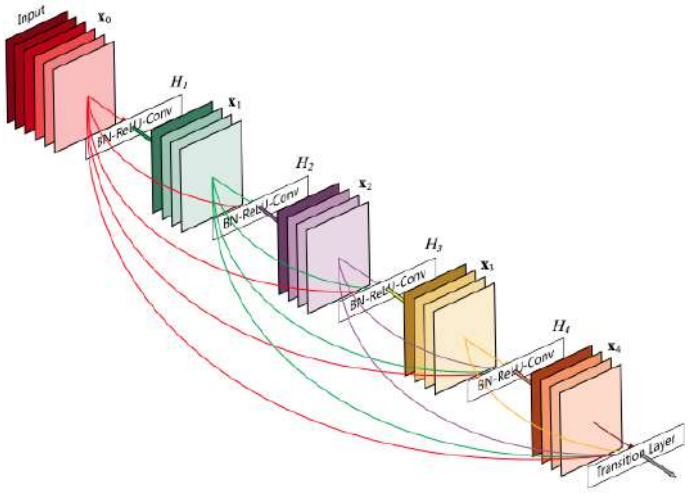


Figure 1.16 – Diagram of a DenseNet architecture (Huang et al., 2018).

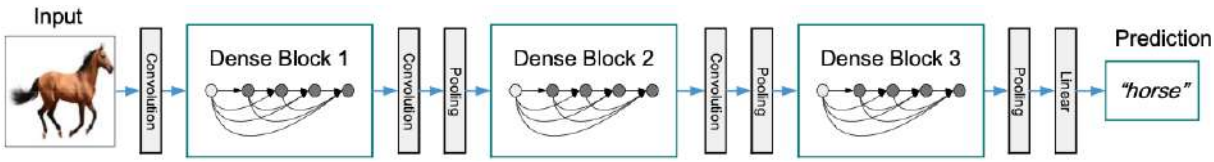


Figure 1.17 – Example of a deep DenseNet with three dense blocks. The layers between two adjacent blocks are called transition layers and change the size of the feature maps by convolution and pooling (Huang et al., 2018)

1.2.7 Visualization tools

1.2.7.1 Grad-CAM

Grad-CAM is a class-discriminative localization technique that generates visual explanations for any CNN-based network without requiring architectural changes or re-training. It uses the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest. Although this technique is fairly general in that it can be used to explain activations in any layer of a deep network (Selvaraju et al., 2020).

Grad-CAM is used to visualize the regions of the image that provide support for a particular prediction. The results reported in Figure 1.18 correspond to the VGG-16 (Simonyan and Zisserman, 2015) network trained on ImageNet.

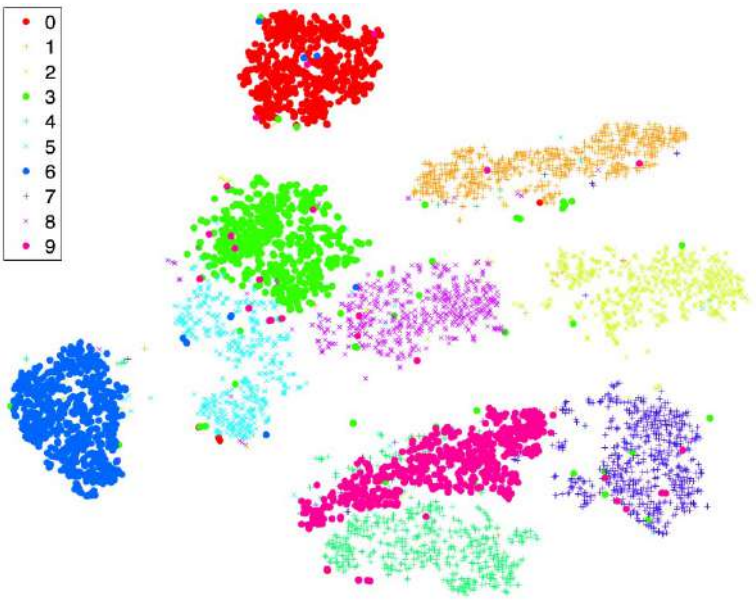
Figure 1.18 – Visualizations for randomly sampled images from the COCO validation dataset. Predicted classes are mentioned at the top of each column (Selvaraju et al., 2020).

1.2.7.2 t-SNE

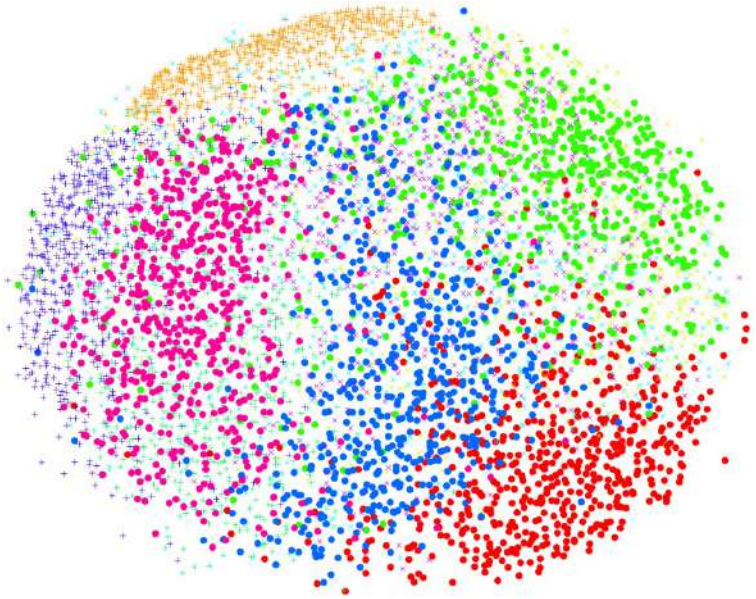
This technique called "t-SNE" visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize, and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map. t-SNE is better than existing techniques at creating a single map that reveals structure at many different scales (Van der Maaten and Geoffrey, 2008).

Van der Maaten and Geoffrey (2008) show, in figure 1.19a and figure 1.19b, the results of their experiments with t-SNE and Sammon mapping on the MNIST data set. The results reveal the strong performance of t-SNE compared to the other techniques. In particular, Sammon mapping constructs a

"ball" in which only three classes (representing the digits 0, 1, and 7) are somewhat separated from the other classes. In contrast, t-SNE constructs a map in which the separation between the digit classes is almost perfect. Moreover, detailed inspection of the t-SNE map reveals that much of the local structure of the data (such as the orientation of the ones) is captured as well. The map produced by t-SNE contains some points that are clustered with the wrong class, but most of these points correspond to distorted digits many of which are difficult to identify.



(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.

Figure 1.19 – Visualizations of 6,000 handwritten digits from the MNIST data set (Van der Maaten and Geoffrey, 2008).

1.3 Geophysics

The use of physics to study the interior of the Earth, from land surface to the inner core, is known as *solid earth geophysics*. This can be subdivided further into *global geophysics*, or alternatively *pure geophysics*, which is the study of the whole or substantial parts of the planet, and *applied geophysics*, which is concerned with investigating the Earth's crust and near-surface to achieve a practical and, more often than not, an economic aim.

'Applied geophysics' covers everything from experiments to determine the thickness of the crust (which is important in hydrocarbon exploration) to studies of shallow structures for engineering site investigations, exploring from groundwater and for minerals and other economic resources, to trying to locate narrow mine shafts or other forms of buried cavities, or the mapping of archaeological remains, or locating buried pipes and cables - but where in general the total depth of investigation is usually less than 100 m (Reynolds, 2011).

Depending on the physical property involved, applied geophysics can be classified into six different methods. Gravimetry and magnetism, often called passive methods, are based on the study of the variations of an earth field inherent to the medium, respectively the gravitational field and the magnetic field. On the other hand, seismic, electrical, electromagnetic and geo-radar methods, considered as active methods, consist in the study of the reaction of the medium to a stimulus, respectively acoustic waves, direct current, low and high frequency alternating current.

Passive methods use naturally occurring fields (such as Earth's magnetic field), over which the observer has no control, and detect variations caused by geology or man-made objects. Interpretation is usually non-unique, relying a great deal on the experience of the interpreter. Active methods involve generating signals in order to induce a measurable response associated with a target. The observer can control the level of energy input to the ground and also measure variations in energy transmissibility over distance and time. Interpretation of this type of data can be more quantitative. Depth discrimination is often better than with passive methods, but ease of interpretation is not guaranteed (Milsom and Eriksen, 2011).

1.3.1 Magnetic methods

The magnetic method is the oldest and one of the most widely used geophysical techniques for exploring the Earth's subsurface. It is a relatively easy and inexpensive tool to apply to a wide variety of subsurface exploration problems involving horizontal magnetic property variations from near the

base of the Earth's crust to within the uppermost meter of soil. These variations cause anomalies in the Earth's normal magnetic field that are mapped by the magnetic method.

The geomagnetic field is caused by electrical currents associated with convective movements in the electrically conducting outer core of the Earth. Additional significant components of the field are derived from variations in the magnetic properties of the lithosphere and electrical currents in the ionosphere as well as in the Earth. For most exploration purposes the magnetic force per unit pole, or magnetic field strength, is measured either in a directional mode where the sensor is spatially oriented or in a non-directional mode where the sensor obtains total field measurements that are presumed to be collinear with the Earth's magnetic field.

Successful applications of the magnetic method require an in-depth understanding of its basic principles, and careful data collection, reduction, and interpretation. The results of magnetic investigations are seldom sufficiently diagnostic to provide the complete answer to problems concerning the subsurface and, as is the case of other potential field techniques such as the gravity method, the interpretations are not unique. Accordingly, the magnetic method is normally employed in conjunction with other geophysical methods and direct subsurface information. The relative ease and simplicity of applying magnetics makes it a desirable first-choice methodology wherever it can solve or contribute to the solution of subsurface studies (Hinze et al., 2013).

The magnetic method is typically used to :

— Locate abandoned steel well casing, buried tanks, pipes and metallic debris.
— Map old waste sites and landfill boundaries.
— Map basement faults and basic igneous intrusives.
— Investigate archaeological sites.

1.3.2 Magnetic force

Magnetism is associated with the motion of electric charges. At the most fundamental level, the magnetic properties of matter originate in a dipole moment caused by spin and orbital motion of electrons around the nucleus of atoms, and coupling of spins between particular adjacent atoms.

Magnetism involves fields which exert a force on other magnetic bodies and electrically conducting materials. Unlike the monopolar phenomenon of gravitation, magnetism is dipolar with each magnetic component consisting of two poles that attract each other.Thus, for example, the magnetization of a linear body magnetized along its length (e.g. a bar magnet) is represented by two magnetic poles

where the magnetization is concentrated near the opposite ends of the body.

Magnetization or the magnetic moment per unit volume is the operative physical property for the magnetic method of geophysical exploration. It is the vector sum of an induced magnetization given by the product of magnetic susceptibility (the ease with which an object is magnetized in the ambient magnetic field) and the intensity of the terrestrial field, as well any permanent or remanent magnetizations imprinted on the object by previous magnetic fields.

The magnetic force between two magnetic poles is proportional to the product of their strengths (Figure 1.20). If the poles are of opposite sign, the force is attractive, and thus operating to move them towards each other, whereas if the poles are the same sign, the force is repulsive and operating to push them apart (Lowrie and Fichtner, 2020)

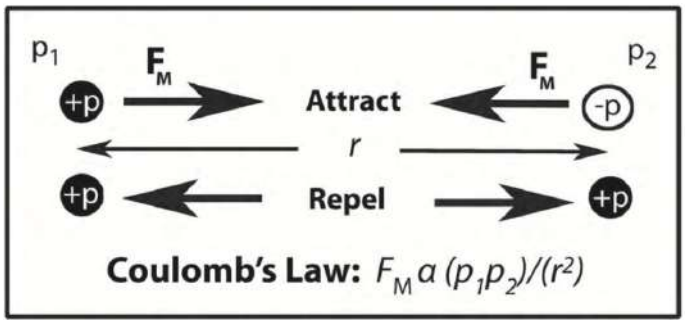


Figure 1.20 – Schematic illustration of the attractive and repulsive magnetic forces (FM) generated between two magnetic poles by Coulomb's law. The unit magnetic dipole (top) consists of two fictitious point poles of equal strengths (p), but opposite signs and separated by an infinitesimal distance (r) (Lowrie and Fichtner, 2020).

1.3.3 Dipoles

The most important type of magnetic field – and also the dominant component of the geomagnetic field – is that of a magnetic dipole (Figure 1.21a). This is the field of two magnetic poles of opposite sense that are infinitesimally close to each other. The geometry of the field lines shows the paths along which a free magnetic pole would move in the vicinity of the dipole. A tiny current loop (Figure 1.21b) and a uniformly magnetized sphere (Figure 1.21c) also have dipole-type magnetic fields around them. Although magnetic poles do not exist physically, many problems that arise in geophysical situations can be readily solved in terms of surface distributions of poles or dipoles (Lowrie and Fichtner, 2020).

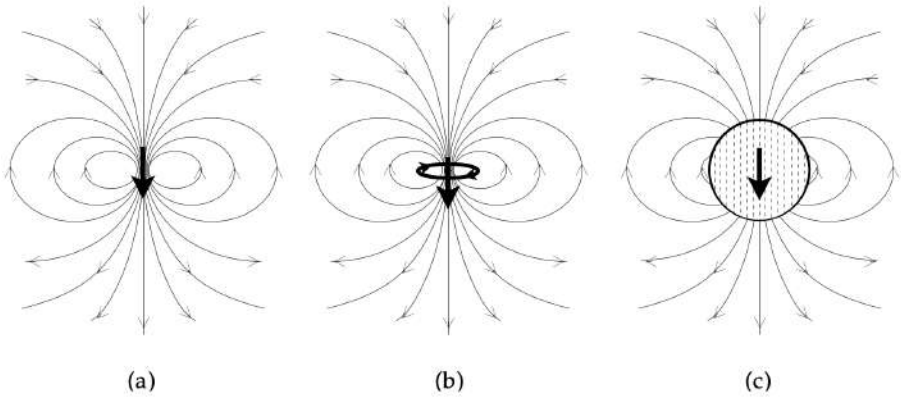


Figure 1.21 – The characteristic field lines of a magnetic dipole are found around (a) a short bar magnet, (b) a small loop carrying an electric current, and (c) a uniformly magnetized sphere (Lowrie and Fichtner, 2020).

1.3.4 The magnetic properties of materials

Magnetic property refers to the response of a material to an applied magnetic field. The macroscopic magnetic properties of a material are a consequence of interactions between an external magnetic field and the magnetic dipole moments of the constituent atoms. Different materials react to the application of magnetic field differently. The most familiar effects occur in ferromagnetic materials, which are strongly attracted by magnetic fields and can be magnetized to become permanent magnets, producing magnetic fields themselves. Only a few substances are ferromagnetic. The most common ones are iron, cobalt and nickel and their alloys. Five basic types of magnetism have been observed and classified on the basis of the magnetic behavior of materials in response to magnetic fields at different temperatures (Figure 1.22).

1.3.4.1 Diamagnetism

All magnetic materials show a diamagnetic reaction in a magnetic field (Figure 1.22a). The diamagnetism is often masked by stronger paramagnetic or ferromagnetic properties. It is characteristically observable in materials in which all electron spins are paired (Lowrie and Fichtner, 2020). Diamagnetism results from changes in electron orbital motion that are induced by an external field. Diamagnetic materials include water, wood, most organic compounds such as petroleum and some plastics, and many metals including copper.

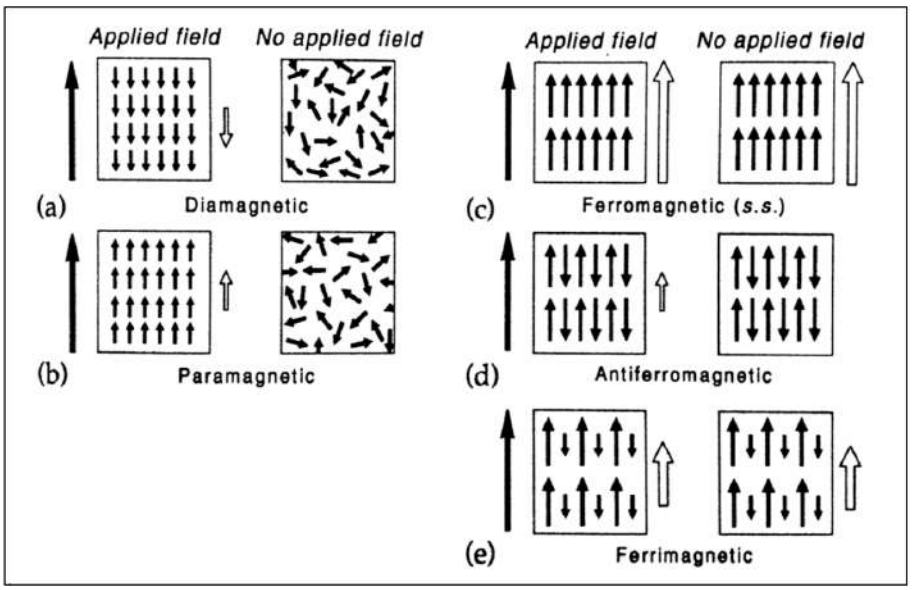


Figure 1.22 – An illustration of the alignment of electron spins and the associated magnetization of different classes of magnetic materials (a-e), both in the presence and absence of an applied field. Black arrows to the left of boxes show the orientation of the applied field, and white arrows to the right of boxes show the orientation of the resulting magnetization (Ives, 2016).

1.3.4.2 Paramagnetism

Paramagnetic materials are those having permanent atomic dipoles, which are acted on individually and aligned in the direction of an external field (Figure 1.22b). This is ans statistical phenomenon. when one or more electron spins are unpaired, the net magnetic moment of an atom or ion is no longer zero. The resultant magnetic moment can align with a magnetic field. The alignment is opposed by thermal energy which favors chaotic orientations of the spin magnetic moments. An important characteristic is that the susceptibility k varies inversely with temperature (Lowrie and Fichtner, 2020). Paramagnetic materials have no remanent magnetization and have a low positive susceptibility which decreases with the increase of the temperature.

1.3.4.3 Ferromagnetism

In paramagnetic and diamagnetic materials the interactions between individual atomic magnetic moments are small and often negligible. However, in some metals, the atoms occupy lattice positions that are close enough to allow the exchange of electrons between neighboring atoms. The exchange is a quantum-mechanical effect that involves a large amount of energy, called the *exchange energy* of the metal. The exchange interaction produces a very strong molecular field within the metal, which aligns the atomic magnetic moments exactly parallel and produces a spontaneous magnetization.

The magnetic moments react in unison to a magnetic field, giving rise to a class of strong magnetic behavior known as *ferromagnetism* (Figure 1.22c) (Lowrie and Fichtner, 2020).

Every ferromagnetic material has its own individual temperature, called the Curie temperature, or Curie point, above which it loses its ferromagnetic properties. This is because the thermal tendency to disorder overwhelms the energy-lowering due to ferromagnetic order. Only a few substances are ferromagnetic. The common ones are iron, cobalt, nickel and most of their alloys, and some compounds of rare earth metals.

1.3.4.4 Antiferromagnetism

In certain minerals, interaction between magnetic spins becomes possible by the exchange of electrons from one metal ion to another through the electron "cloud" of the oxygen ion. This indirect exchange (or superexchange) process results in anti-parallel directions of adjacent atomic magnetic moments (Figure 1.22d), giving two sub-lattices with equal and opposite intrinsic magnetic moments. As a result, the susceptibility of an anti-ferromagnetic crystal is weak and positive, and remanent magnetization is not possible (Lowrie and Fichtner, 2020).

1.3.4.5 Ferrimagnetic

When the indirect exchange process involves anti-parallel and unequal magnetizations of the sublattices (Figure 1.22e), resulting in a net spontaneous magnetization, the phenomenon is called ferrimagnetism. Ferrimagnetic materials (called ferrites) exhibit magnetic hysteresis and retain a remanent magnetization when they are removed from a magnetizing field (Lowrie and Fichtner, 2020).

1.3.5 The earth's magnetic field

1.3.5.1 Origin

William Gilbert, in his *De Magnete*, first identified the Earth's magnetic field as having its origin inside the Earth. He believed that its origin lay in permanent magnetization (lodestone) at the center of the Earth. Later in the first half of the seventeenth century Rene Descartes developed a large following for his ideas on the origin of the Earth's magnetic field (Mattis, 1965). Descartes believed that the Earth's magnetism was associated with "threaded parts" that were channeled in one-way ducts through the Earth whose main entrances and exits were the North and South Poles. In more modem times Ein-

stein, shortly after writing his special relativity paper in 1905, described the problem of the origin of the Earth's magnetic field as being one of the most important unsolved problems in physics. Blackett (1947) proposed that large astronomical bodies might, because of some new as-yet- unexplained law of physics, have dipole moments that are directly proportional to their angular momentum. To test this hypothesis, Blackett (1952) attempted to measure the weak magnetic field predicted from a pure gold sphere as it rotated with the Earth. In spite of using a sensitive astatic magnetometer specially developed for this purpose, he obtained negative results and he concluded his hypothesis was therefore in error. The sensitive astatic magnetometer he produced was subsequently used to measure the weak magnetizations of sedimentary rocks and was an important instrument in the early days of paleomagnetism (Merrill et al., 1996).

1.3.5.2 The Present Geomagnetic Field

Direct measurements of the Earth's magnetic field (usually quoted in the magnetic induction units of nanotesla) are made continuously at magnetic observatories, and are obtained from various oceanographic, land, aircraft, and satellite surveys. A common way of describing the magnetic field is simply to plot different magnetic elements, such as total intensity or inclination. The various elements used in such plots are defined in (Figure 1.23).

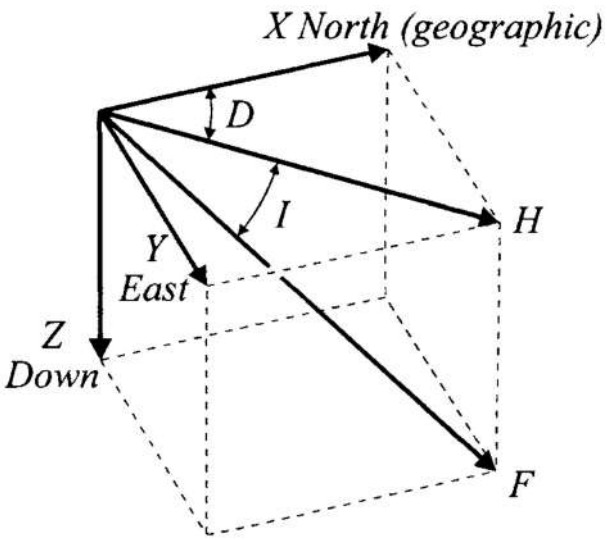


Figure 1.23 – The main elements of the geomagnetic field. The deviation, *D*, of a compass needle from true north is referred to as the *declination* . The total field *F*, which is at an angle *I*, termed the inclination (or dip), to the horizontal. The horizontal *(H)* and vertical *(Z)* components of *F* are given by *H=FcosI* and *Z=FsinI*, respectively (Merrill et al., 1996).

Following the work of Gilbert in the seventeenth century, geomagnetists have recognized that the magnetic field at the surface of the Earth can be well approximated by a magnetic dipole placed at the Earth's center and tilted about 11° with respect to the axis of rotation (Merrill et al., 1996).

The intersections between the dipole axis and the surface are defined as the magnetic poles. These should not be confused with the geographical poles which correspond to the intersections between the axis of rotation and the surface. The magnetic and geographic poles are therefore offset. However, the first ones get their quality of north and south from the last ones by pure convention, so that the magnetic poles are reversed compared to those of the terrestrial dipole (Figure 1.24).

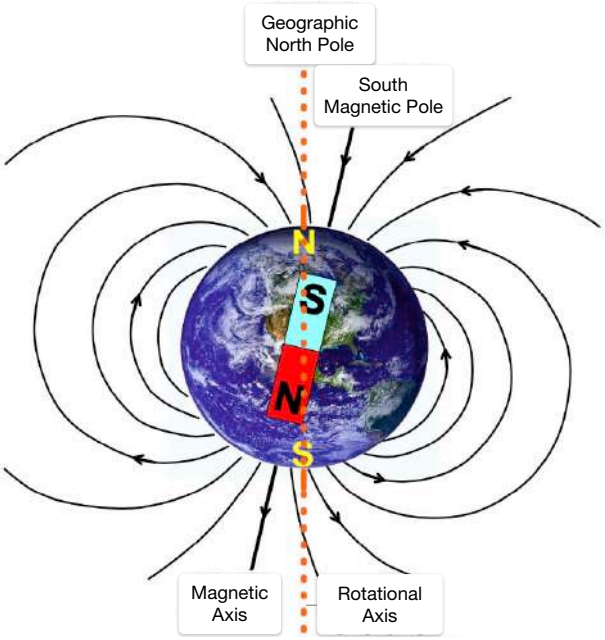


Figure 1.24 – Representation of the earth's dipolar magnetic field (modified from Florsch et al. (2019)).

A considerable amount of information concerning the Earth's magnetic field and its variation in time is contained in the spherical harmonic coefficients. Examples of various magnetic charts that can be derived from use of the spherical harmonic coefficients, giving a pictorial representation of the Earth's magnetic field and its secular variation, are shown in figure 1.25, figure 1.26, and figure 1.27 (Merrill et al., 1996).

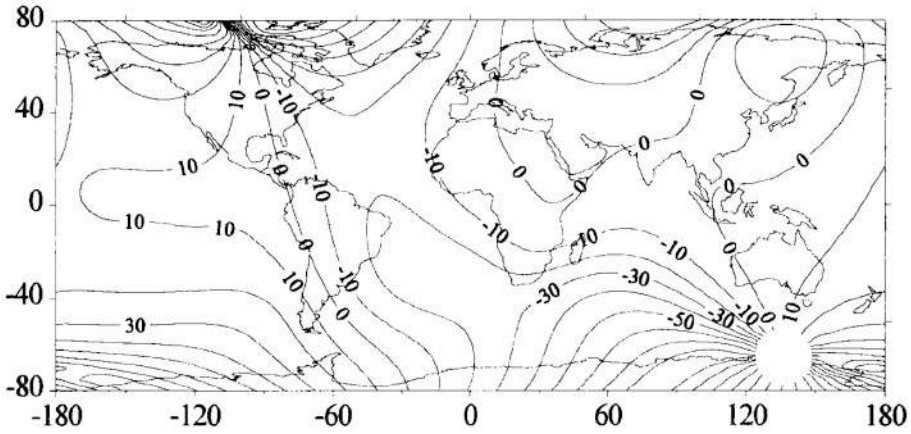


Figure 1.25 – Isogenic chart for 1990 showing the variation in declination in degrees over the Earth's surface (Merrill et al., 1996).

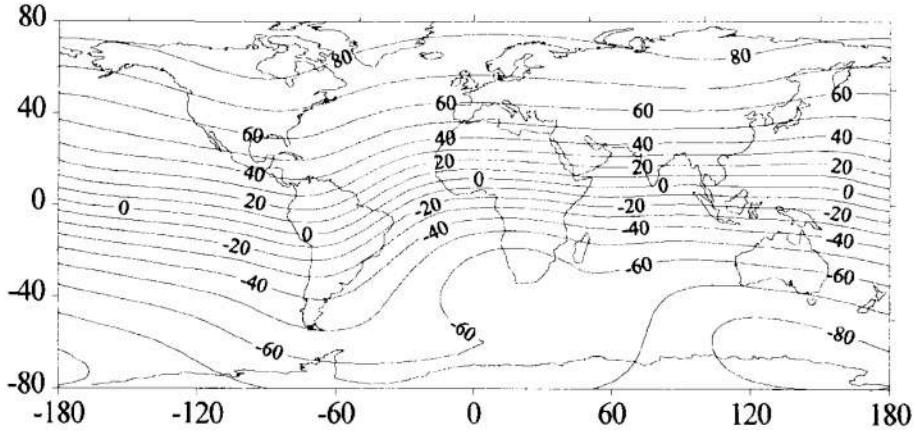


Figure 1.26 – Isoclinic chart for 1990 showing the variation of inclination in degrees over the Earth's surface (Merrill et al., 1996).

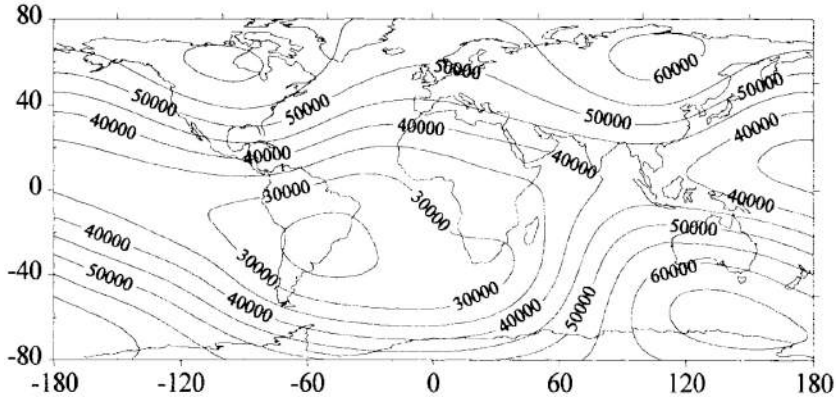


Figure 1.27 – Isodynamic chart for 1990 showing the variation of total intensity over the Earth's surface. Contours are labeled in nT (Merrill et al., 1996).

1.3.5.3 Induce magnetization

Magnetic anomalies in the earth's magnetic field are caused by two different kinds of magnetism: induced and remanent (permanent) magnetization. Induced magnetization refers to the action of the field on the material wherein the ambient field is enhanced and the material itself acts as a magnet. The magnetization of such material is directly proportional to the intensity of the ambient field and to the ability of the material to enhance the local field -a property called *magnetic susceptibility.* The induced magnetization is equal to the product of the volume magnetic susceptibility, k, and the earth's or ambient field intensity, F, or :

$$I_i = kF \tag{1.17}$$

Where I_i is the induced magnetization per unit volume in cgs electromagnetic units, and F is the field intensity in gauss. For most materials, k is much less than 1 and, in fact, is usually +10-6 cgs or smaller.

If k is this small and positive, the material is said to be *paramagnetic* and, when negative, *diamagnetic* (Breiner, 1973).

1.3.5.4 Remanent or permanent magnetization

The remanent of permanent magnetization, I_i, (the former ascribed to rocks, the latter to metals) is often the predominant magnetization (relative to the inductive magnetization) in many igneous rocks and irons alloys. Permanent magnetization depends upon the metallurgical properties and the thermal, mechanical and magnetic history of the specimen, and is independent of the field in which it is measured.

The remanent magnetization of a rock or object may or may not be in the same direction as the present earth's field for the object may have been reoriented and because the earth's s field is known to have changed its orientation in geologic an even historic time. Rocks are frequently reversely magnetized so that the measurement of this remanent magnetization is a useful aid to interpretation if the rocks which produce and observed anomaly are, indeed, accessible (Breiner, 1973).

1.3.6 Magnetic data acquisition

The acquisition of magnetic data is relatively simple, rapid, and less complex than are the observations of data of most geophysical methods. Most observations are made of the scalar, total intensity of the field, with alkali-vapor (resonance) magnetometers which readily achieve a sensitivity of better than a nanotesla with rates of several observations per second from a moving platform.

Prior to the late 1940s, magnetic surveying was conducted primarily with ground-based instrumentation which measured a vector component of the field to an accuracy approaching a few nanotesla with tripod-mounted mechanical balance magnetometers. Since then the availability of more precise and accurate electronic instruments which are insensitive to motion and orientation has led to rapid and efficient surveying on mobile platforms, especially from aircraft, with accuracies of the order of 1 nT. More recently an accuracy of 0.1 nT is sometimes targeted in high-resolution surveys. These levels of accuracy are readily achieved by available instrumentation and can be approached in data reduction depending on the noise envelope of the observations (Hinze et al., 2013).

1.3.6.1 Instrumentation

Since World War II, a variety of techniques have been used to measure the angular relationships and magnitudes of the vector components of the Earth's magnetic field both absolutely and relatively. Some examples of instrumentation are : resonance magnetometers, SQUID magnetometers, and flux-gate magnetometers; which are addressed in this thesis.

In specialized applications of the magnetic method, it may be desirable to obtain the directional attributes as well as the total magnitude of the magnetic field. The scalar intensity of the field is determined by resonance-type magnetometers, but flux-gate magnetometers are used to measure vector components of the field. The flux-gate instrument, which is no longer regularly used for measuring the total field variations in air- borne magnetometry, measures only the relative change, not the absolute value, of the geomagnetic field and is directionally dependent.

Construction details vary among flux-gate instruments, but, in the classical design, this magnetometer consists of a pair of identical, but oppositely wound inductive coils with cores of the same high magnetic permeability material along their axes. The cores are magnetized to saturation by the induced fields from an alternating current passed through the windings of the coils in opposite directions.

Three mutually perpendicular elements appropriately oriented can measure the individual vector components of the field, and thus their angular relationships and by calculation the total field intensity. Sensitivities of fluxgate magnetometers are usually about 1 nT up to the order of 0.1 nT, in specially constructed instruments. A wide variety of alternative designs of the fluxgate magnetometer have been developed including the cylindrical core flux-gate and the ring-core design which has found considerable use in space measurements because of its low mass and simplicity (Hinze et al., 2013).

1.3.6.2 The different modes of acquisition

Before the mid-1940s, essentially all exploration magnetic observations were vector components, primarily the vertical magnetic anomaly field. However, since the advent of the airborne flux-gate and resonance magnetometers, essentially all exploration magnetic observations have been made of the scalar field. It was realized that the total magnetic anomaly field is in error because the amplitude of the total geomagnetic field is subtracted in a scalar sense from the observed field to determine the anomaly regardless of the orientation of these components. Nonetheless, the error is minor for most

crustal magnetic anomalies and only approaches 10% as the anomalies exceed 10.000 nT. As a result the scalar total magnetic intensity has been and continues to be the acceptable magnetic anomaly component for exploration purposes (Hinze et al., 2013).

Acquisition modes in magnetic measurements can be divided into : total field measurements (made with one magnetometer) and gradient measurements (made with two magnetometers).

Gradient measurement : it is of some interest in exploration to measure various gradients, particularly the vertical gradient using a portable magnetometer. The average horizontal gradient along the traverse can easily be computed from the profile, whereas the vertical gradient from typical, widely-spaced ground traverses cannot be accurately computed. A *gradiometer* is so named because it measures the gradient of the total field. The gradiometer is defined as a differential magnetometer where the spacing between sensors is fixed and small with respect to the distance to sources whose gradients are to be measured. The difference in intensity divided by the distance between sensors is then the gradient measured at the midpoint of the sensor spacing. As an additional condition, it is relatively important in any ground gradient applications that there be no significant surface magnetic noise, for gradient anomalies tend to greatly enhance such shallow noise sources which would be detrimental for most objectives.

The vertical gradient or any gradient for that matter has several properties of interest in exploration. First, gradient anomalies tend to resolve composite or complex anomalies into their individual constituents and on the same basis automatically remove the regional magnetic gradient to better define the shallower anomalies assumed to be of interest. Also, the magnetic time variations including the effects of magnetic storms are effectively removed. Third useful attribute of the gradients is that they can be used very quantitatively or for their vector properties (gradient of the scalar) in ascertaining anomaly depth, magnetic moment, shape and location (Breiner, 1973).

Total field measurement : in total field measurement only one sensor is used, which makes the device lighter and more compact than a gradiometer. It is possible to use scalar magnetometers, which measure directly the magnetic field intensity, or vector magnetometers, which allow to calculate intensity as well as direction from the three components of space. The advantage of this last type of sensor is the possibility of modeling and correcting known signals disturbing the measurement, like those generated by the equipment itself.

Compared to the gradiometric measurement, the total field measurement has the advantage of recording the signal in its entirety without emphasizing specific wavelengths. It is therefore possible to image deeper structures or those with a lower magnetization contrast with the surrounding material. On

the other hand, the signal includes both temporal and spatial variations of the and spatial variations of the magnetic field.

1.3.7 Magnetic data interpretation

Dipolar magnetic anomalies are found most frequently at historic sites, where iron artifacts have been lost and buried. However, these simplest of anomalies are also apparent at archaeological sites of any age, where they may be caused by a refilled pit, a fire hearth, or a magnetic stone. While a magnetic map may reveal these objects as small and circular patterns, an analysis or interpretation of those patterns is required for estimating the location, depth, and perhaps the mass of the objects that are underground.

Location can be approximated by the peak of the anomaly, but depth is not immediately apparent. Depth is valuable for it can suggest age (shallow objects may be modern); it will also reveal the amount of excavation that is needed to expose the object.

A good estimate of depth can be made by measuring the diameter of an anomaly. More accurate results may be determined with a computer program that calculates the field of an ideal magnetic dipole and keeps changing its position until the calculated pattern matches the measurements. A very different interpretation can be done with a program that examines each measurement in a magnetic map for the magnitude of the field and how rapidly it changes with position. With this basic information, the possible location of nearby dipoles can be calculated; if several calculations yield the same location, that location is probably correct (Bevan, 2017).

Two examples of geophysical programs that interpret magnetic anomalies are MagPick (Tchernychev, 2009) and Potent (Almond, 2006).

1.3.7.1 MagPick

MagPick is a semi-automatic program that analyzes one or a few anomalies at a time until all have been studied. A summary of the steps for analyzing all of the dipoles in a magnetic map are:

1. Prepare a map of all of the measurements.
2. Zoom in to a single anomaly or cluster of anomalies. Draw a line between the high and low pair of each anomaly.
3. Another entry in the list.

4. Start the inversion of the dipole parameters.

5. Delete the zoomed map.

6. Return to the full map.

Then, it is necessary to repeat this cycle, starting at step 1, for all interesting anomalies in order to end with a map showing the picked lines for all analyzed dipoles.

Since the MagPick program was first written for the study of marine magnetic surveys, many parts of the program emphasize the mapping and analysis of surveys that were done with lines of traverse that deviate from straight and which do not have a uniform spacing. The MagPick program allows the interpretation of only total-field magnetic maps, but it is also possible to analyze maps that have been measured with gradiometers by applying a correction to the findings of the program (Figure 1.28).

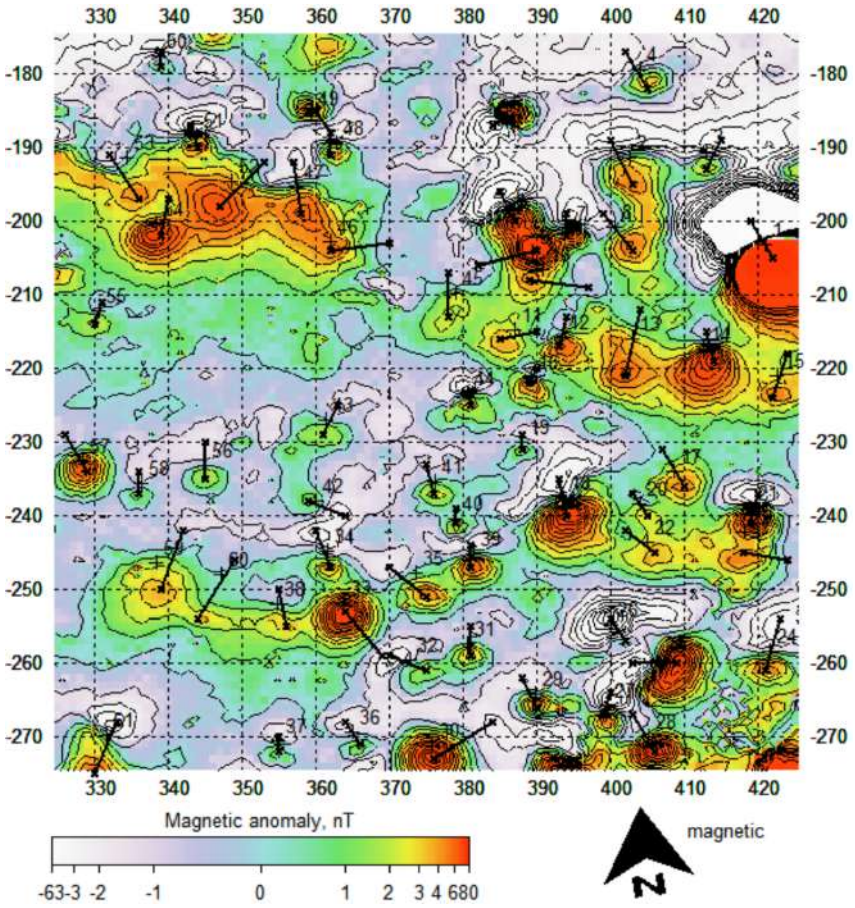


Figure 1.28 – The analysis of the entire bombproof map with the MagPick Program. The highs and lows of 61 dipolar anomalies have been marked here with short line segments. This entire collection of anomalies was not analyzed all at once. Instead, small clusters of nearby anomalies were analyzed as a group until all anomalies had been studied (Bevan, 2017).

1.3.7.2 Potent

With the Potent program, one can select polygonal areas for analysis. By including only the anomalies of interest, the analysis will not be affected by unwanted patterns; this careful selection of the areas for analysis is not possible with MagPick, for the areas of analysis there must be rectangles. The magnetic models are created from all of the measurements in the area for analysis; if some of these measurements are anomalies from nearby objects, the parameters that are determined can have errors.

The procedure used by the program consists of generating the best magnetic models for the anomalies of a magnetic map; it can be called a model-based analysis. It is sometimes called a magnetic map inversion. The correct or best values for the parameters can be determined by automatically changing them until the calculation of the magnetic field is similar to the measurements of an anomaly. An example of dipolar models that were calculated by the Potent program are plotted in Figure 1.29; their location, depth, and magnetic moment are revealed.

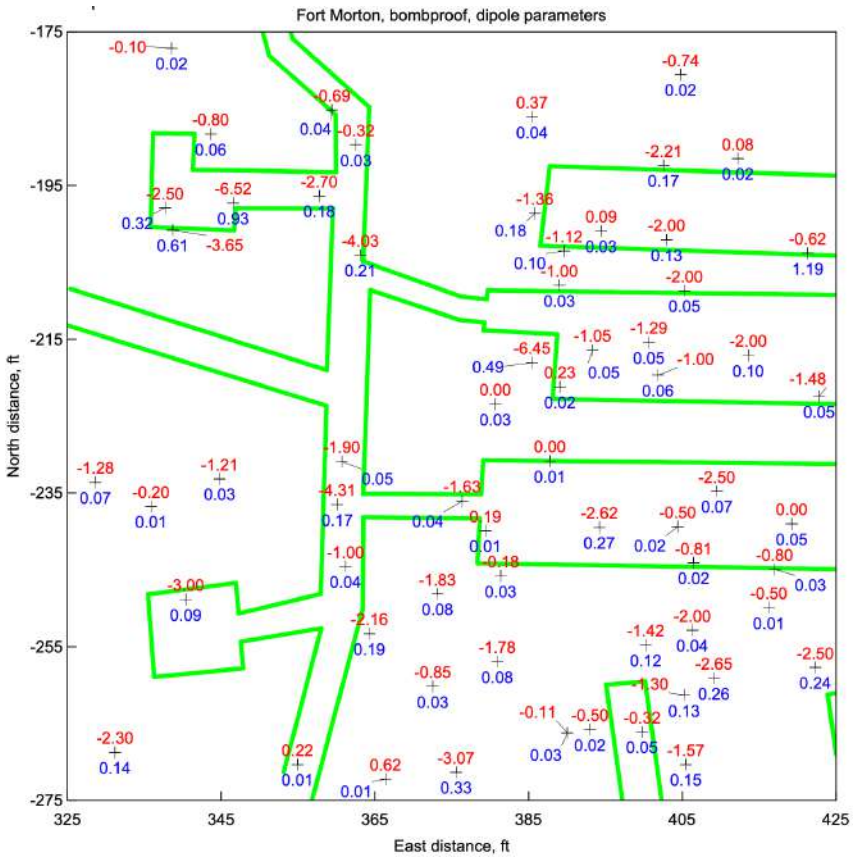


Figure 1.29 – The interpretation from the Potent program. Plus symbols + locate the objects, while their depths are listed in red, and their magnetic moments in blue. Like the MagPick program, this program also calculates the total magnetization and its direction for each dipolar anomaly (Bevan, 2017).

There are advantages to this model-based analysis. First, all of the parameters of the source are determined with one process. Second, the model-based analysis allows a valuable confirmation of the quality of the solution: This confirmation is supplied by a map that shows the difference between the measured map and the calculated map. If this residual map is not blank, the patterns that remain are anomalies that have not been analyzed completely (Bevan, 2017).

CHAPTER

2 Methodology

In this chapter, I will explain and illustrate step by step the methodology developed throughout my research (Figure 2.1). It will first focus on the definition of a hypothesis on the nature of the geophysical data. This is important to have an idea of the type of neural network architecture to use. Second, the creation of a database in order to have enough examples available for the learning phase. The goal is to achieve stability in learning and to avoid overfitting at the end of it. Third, the selection of a deep neural network architecture according to the tasks defined in our objectives. Finally, I will specify some techniques that we plan to implement to verify, statistically and visually, the accuracy of our model.

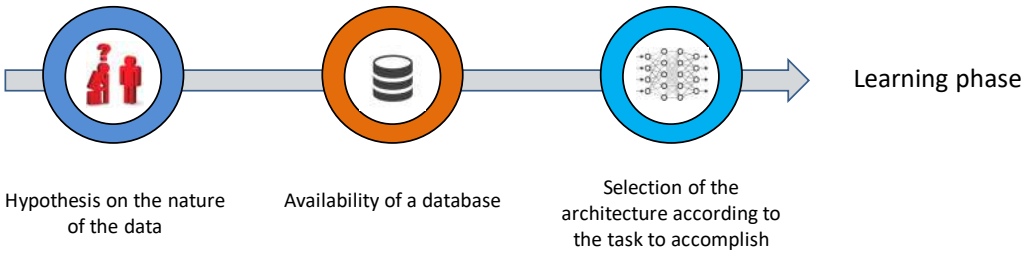


Figure 2.1 – Diagram of the methodology adopted for this research.

2.1 Hypothesis on the nature of the database

In deep learning, There are a multitude of models and model implementations that may or may not be suitable for a specific task. To select one, it is important to understand our problem domain and the nature of our data (e.g., text, audio, image, etc). As a starting point, we decided to work with geophysical magnetic data due to the simplicity of generating synthetic data. In the absence of specific bibliography on our magnetic application, I decided to use convolutional neural networks ("CNN"), often applied to images, because of the similarity between the classical representation of magnetic data (as a colored map) and black/white images. We consider that the magnetic maps contain in each pixel a magnetic induction value similar to the information per pixel in a gray image (Figure 2.2).

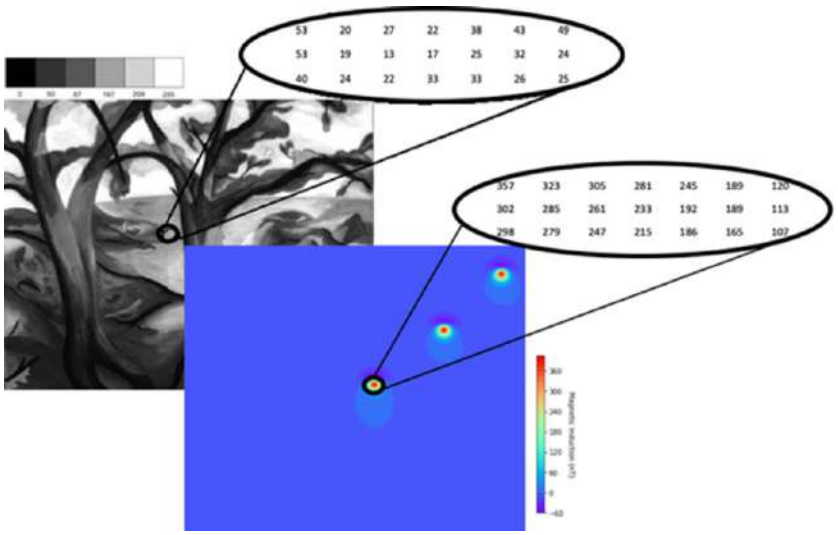


Figure 2.2 – Similarity between a magnetic map and a gray image.

2.2 Synthetic data creation

In deep learning, the accuracy of the predictions made by neural networks depends to a large extent on the complexity of the architecture and the volume of data available for learning. Contrary to many other domains in which neural networks have been highly successful, such as image classification (e.g., Krizhevsky et al., 2017), the availability of geophysical data is often a limiting factor, as the number of available real maps is low (a thousand examples at the most).

To generate a synthetic database, I first used a Matlab code provided by my co-supervisors (Nicolas Florsch and Christian Camerlynck) which calculates dipolar anomalies. This script considers the equation 8.17 proposed by Scollar (1990) (see Equation 2.1) that calculates the vertical component of the total field intensity. We adapted this equation to consider the pseudo-gradient measurement (in nT/m) provided by a fluxgate magnetometer. This specific consideration will be discussed in the results and discussion section.

$$\Delta F_{x,y} = \frac{\mu_o}{4\pi(H_{high}-H_{bottom})} M \left\{ \begin{array}{c} \frac{1}{(\sqrt{x^2+y^2+(h+H_{bottom})^2})^5}[(2(h+H_{bottom})^2-x^2-y^2)sin(I)-3yzcos(I)] \\ -\frac{1}{(\sqrt{x^2+y^2+(h+H_{top})^2})^5}[(2(h+H_{high})^2-x^2-y^2)sin(I)-3yzcos(I)] \end{array} \right\} \tag{2.1}$$

With:

— $\Delta F_{x,y}$ = dipole magnetic anomaly located at position (0,0) (nT/m);

— μ_o = magnetic permeability of vacuum (H/m);

— M = magnetic moment of the induced dipole ($A.m^2$);

— h = depth of the dipole to the ground (m);

— H_{bottom}, H_{high} = depth of the dipole to the low sensor and the high sensor respectively (m);

— I = Inclination of the dipole (radians).

Then, I improved our initial script and I converted it to Python format to automatize the process of synthetic data creation by considering multiple configurations of the parameters of magnetic bodies. I chose this programming language because most deep learning frameworks (e.g., tensorflow (Martín Abadi et al., 2015) and keras) and some of geophysical simulation and inversion tools (Cockett et al., 2015) are written in Python. Figure 2.3 shows an example of a dipolar anomaly generated by a sphere of magnetic moment 52 A.m^2, buried at 1.4 m depth, and in the presence of a horizontal primary field.

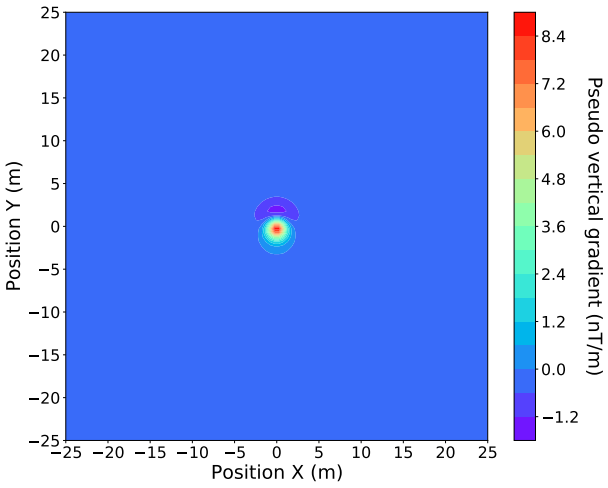


Figure 2.3 – Example of a synthetically generated magnetic anomaly map.

After obtaining promising results with our first database, we decided to increase the complexity of our magnetic maps. First I chose a range of values for the parameters of depth, radius, and magnetic declination. It is important to point out that the declination parameter does not appear in Equation 2.1 because Scollar (1990) ignored it by considering magnetic north to coincide with geographic north. I used data augmentation strategies to rotate each anomaly to simulate the effect we would observe if the declination parameter was considered (Figure 2.4). Secondly, one dipolar anomaly was generated for each combination of parameters by randomly varying their positions (x,y) on the map. Then, I considered 4 different latitude values to check if the neural network is able to differentiate between dipolar anomalies of different shapes (Figure 2.5). Lastly, in order to increase the number of anomalies, I summed the generated maps with the same parameter distribution, respecting a min-

imum distance of 2 meters between dipoles to avoid any coalescence of anomalies (Figure 2.6a and Figure 2.6b). The choice of using the same distributions to sum the maps avoids hiding anomalies whose amplitudes would be too different.

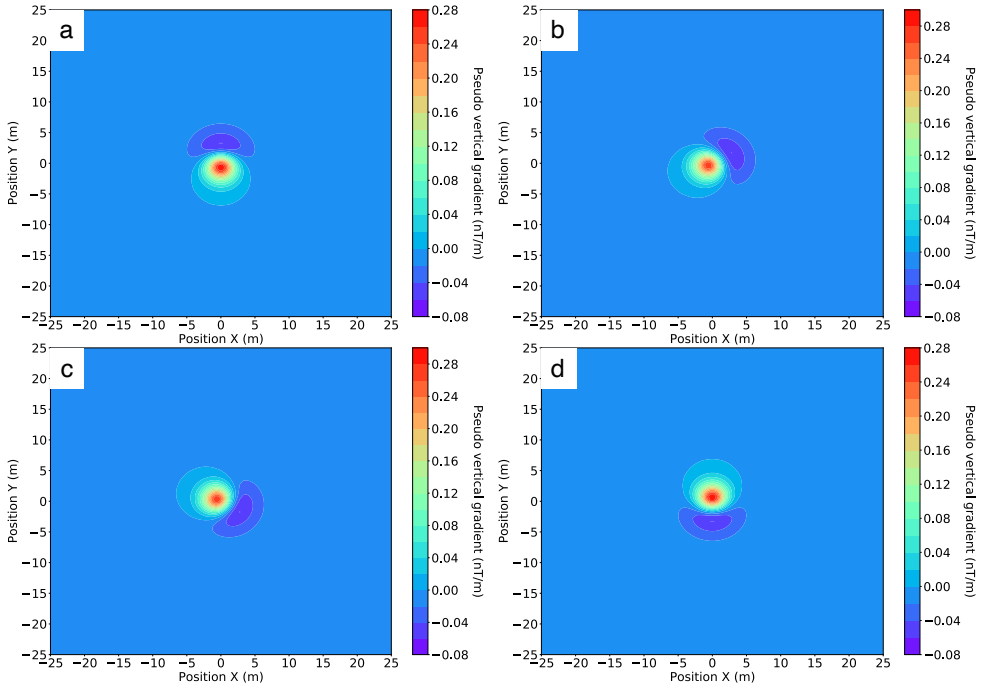


Figure 2.4 – Induced magnetic anomaly maps considering declination values of 0°(a), 60°(b), 120°(c), and 180°(d).

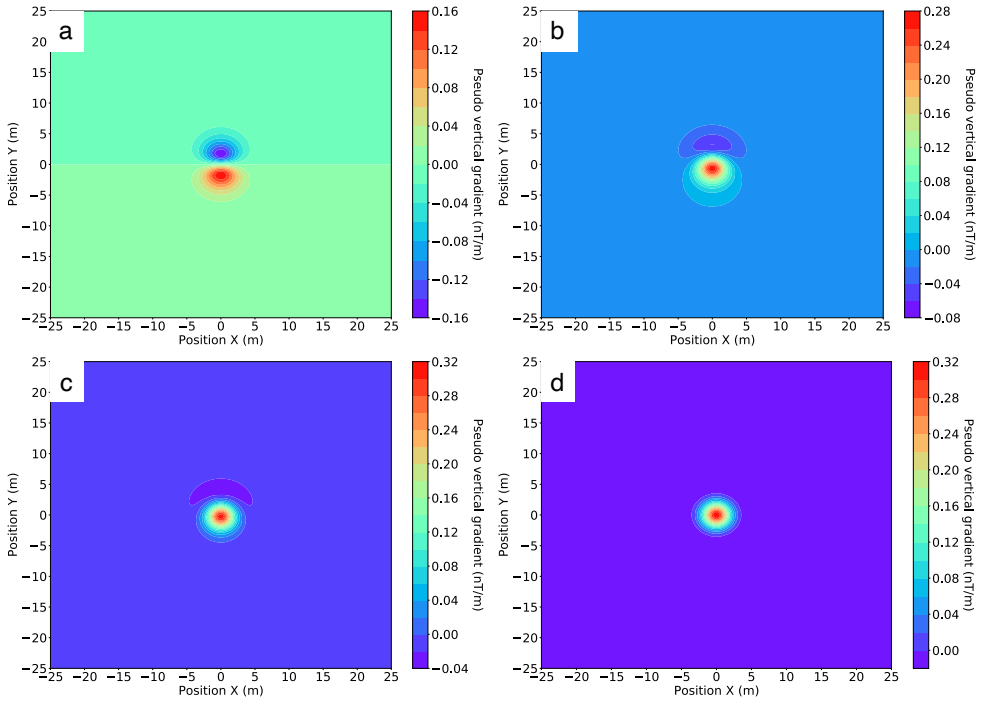


Figure 2.5 – Induced magnetic anomaly maps considering inclination values of 0°(a), 30°(b), 60°(c), and 90°(d).

At this level of complexity, we observed that our database was getting closer to what we would observe

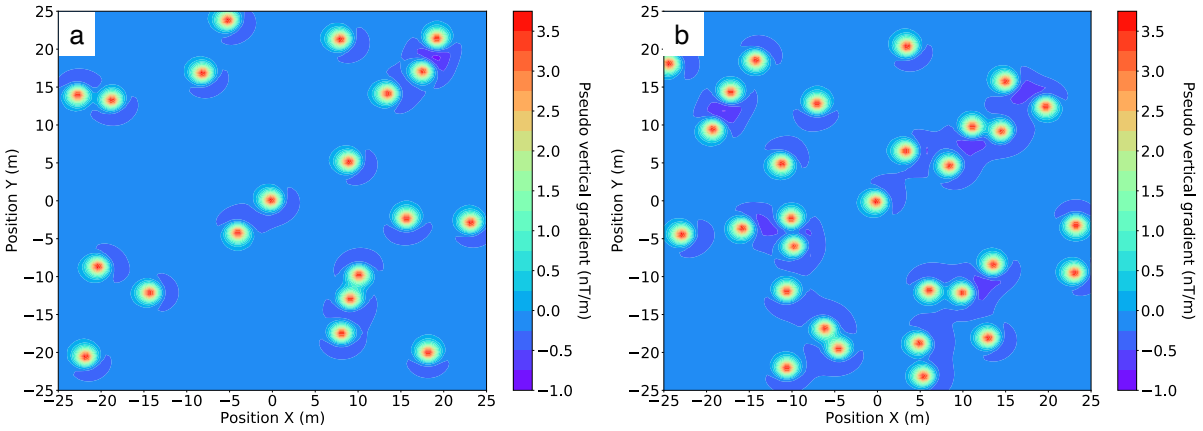


Figure 2.6 – Induced magnetic maps displaying 20(a) and 30(b) dipolar anomalies.

in a real case study. To go further, we took into account the presence of noise (Figure 2.7), a characteristic often observed during data acquisition. Magnetic measurements contain many additional unwanted components that become embedded in the data, originating from the use of the instruments, the soil conditions, or from external disturbances. These unwanted components are usually referred to as 'noise'. The challenge is to extract meaningful information from such composite data and differentiate between the sought-after 'signal' and the undesirable 'noise' (Schmidt et al., 2020). Heuristically, we might expect that the noise will 'smear out' each data point and make it difficult for a neural network to fit individual data points precisely, and hence will reduce over-fitting. In practice, it has been demonstrated that training with noise can indeed lead to improvements in network generalization (Bishop, 1996).

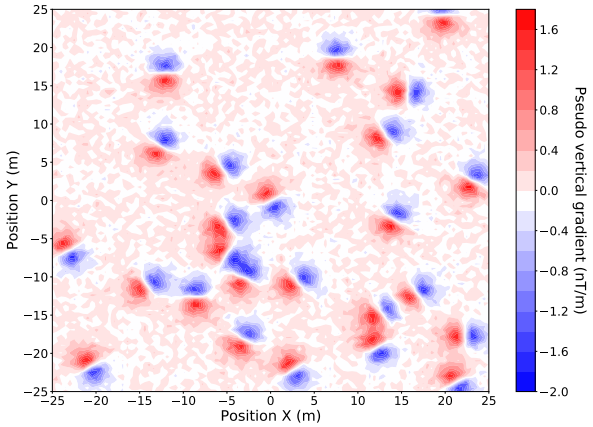


Figure 2.7 – Example of a Gaussian noise applied to a magnetic anomaly map.

Figure 2.8 is a summary of the evolution of our database complexity since the beginning of our research.

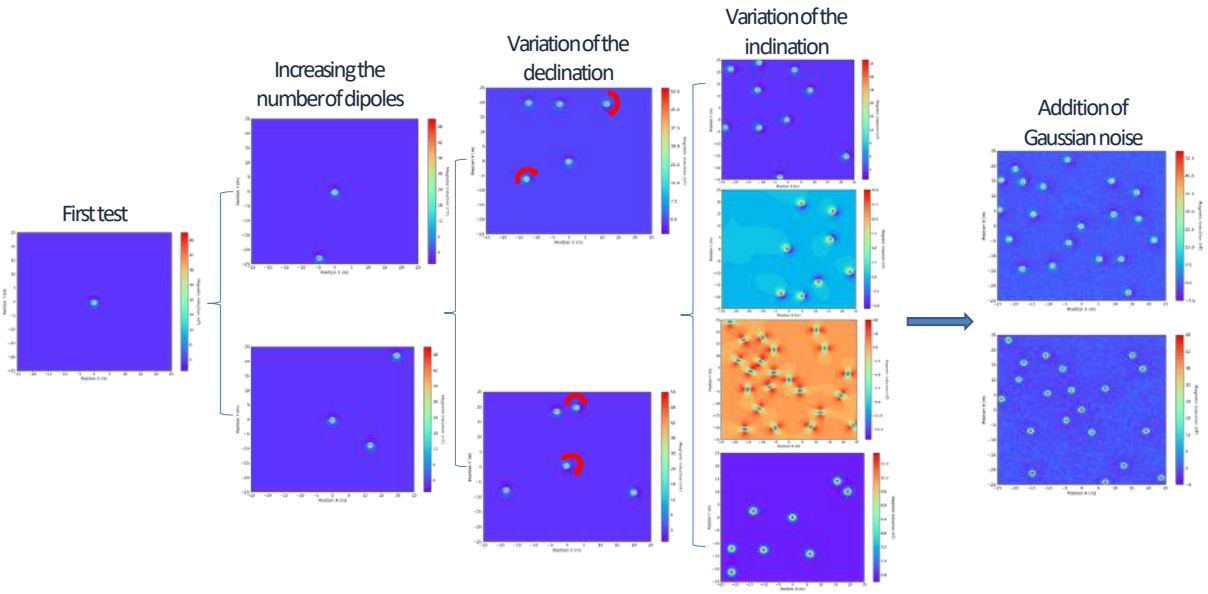


Figure 2.8 – Evolution of the complexity of our synthetic database. It covers from the simplest dipolar anomaly up to the incorporation of Gaussian noise in some magnetic maps.

2.3 Architecture selection

As presented before, I decided to use convolutional neural networks according to our proposed hypothesis. I evaluated several combinations of CNN models before finding the one that best fits our objectives.

I proposed to use the architecture called "autoencoder" (Figure 2.9) to verify that the network manages to identify the characteristics of the input models by reproducing the same model in output. This is an important step to validate the possible use of this type of architecture with our database. Our first results demonstrated that the network was able to reconstruct the shape of the dipolar anomaly we used as input data (Figure A1-1). It means that the convolutional layers were able to detect edges, shapes or even pixel intensities that we observed in the magnetic map.

Then, I decided to use the VGG16 architecture (Simonyan and Zisserman, 2015), which is a well-known CNN architecture that achieved state-of-the-art results in 2014. This architecture is composed of multiple convolutional layers, which I tested before. One of its advantages is that it can be used for classification and regression tasks.

The first version of this architecture that I created (Figure 2.10-1st version) was able to accurately predict the parameters associated with one dipolar anomaly. However, in most cases, there are multiple magnetic anomalies in a real magnetic map. The second version corrected this inconvenient by multiplying the number of parameters in the output layer by the number of dipole anomalies presented

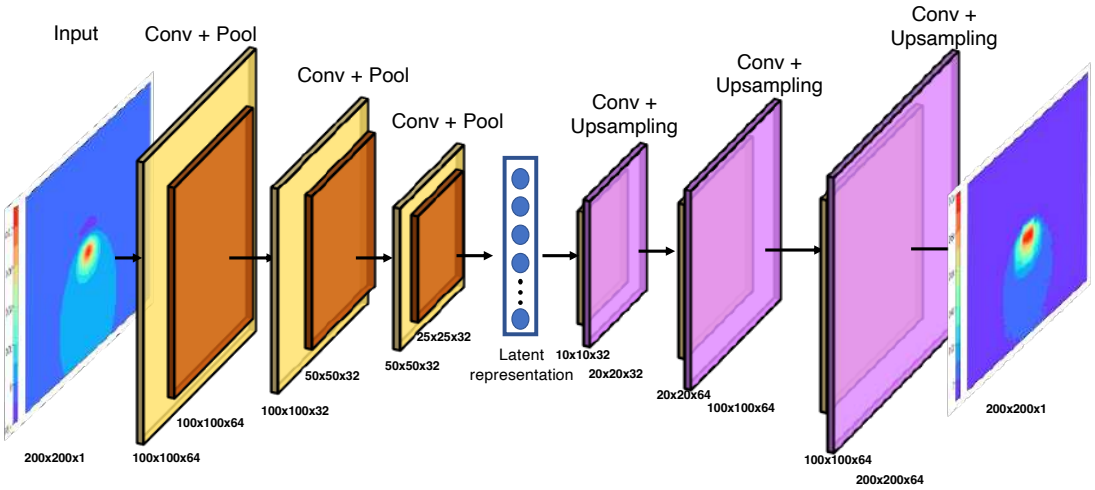


Figure 2.9 – Schema of the autoencoder architecture used. The output size of each layer is specified at the bottom of each rectangle.

in the magnetic map (Figure 2.10-2nd version). This version achieved accurate predictions but its output size was limited by the maximum number of dipolar anomalies considered in the database. In addition, the learning phase time was getting too long with the increase of the output size. Lastly, I tested a third version of the same architecture, even if I already observed some disadvantages in the previous versions, I wanted to verify if it was able to predict not only the parameters of each anomaly but also its location on the map (Figure 2.10-3rd version). As expected, the model was not able to predict the location of the anomalies and the accuracy of the parameter prediction was decreasing.

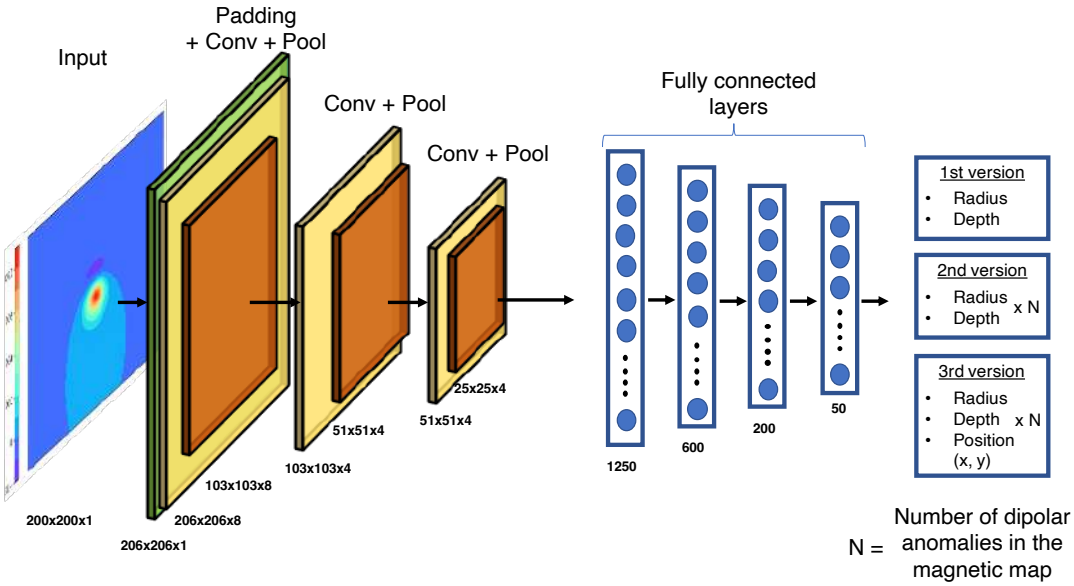


Figure 2.10 – Schema of the VGG16 architecture used. The different versions of the output data are shown on the left of the image. Each version is composed by 3 convolutional layer and 4 fully connected layers.

Up to this point, I created an architecture that was able to accurately predict the parameters of a

limited number of dipoles but was not able to locate them or even count them on a magnetic map. We believe that these latter tasks would be genuinely useful in a real-time magnetic survey compared to the way magnetic data are measured and interpreted nowadays.

Eventually, I decided to use the U-Net architecture (Figure 2.11) that was invented to deal with biomedical images where the target is not only to classify whether there is an infection or not but also to identify the area of infection. The advantage of this architecture is that it performs segmentation tasks, also known as pixel-based classification. It classifies each pixel of an image as belonging to a particular class. Besides medical images, another popular databases are seismic images and GIS objects.

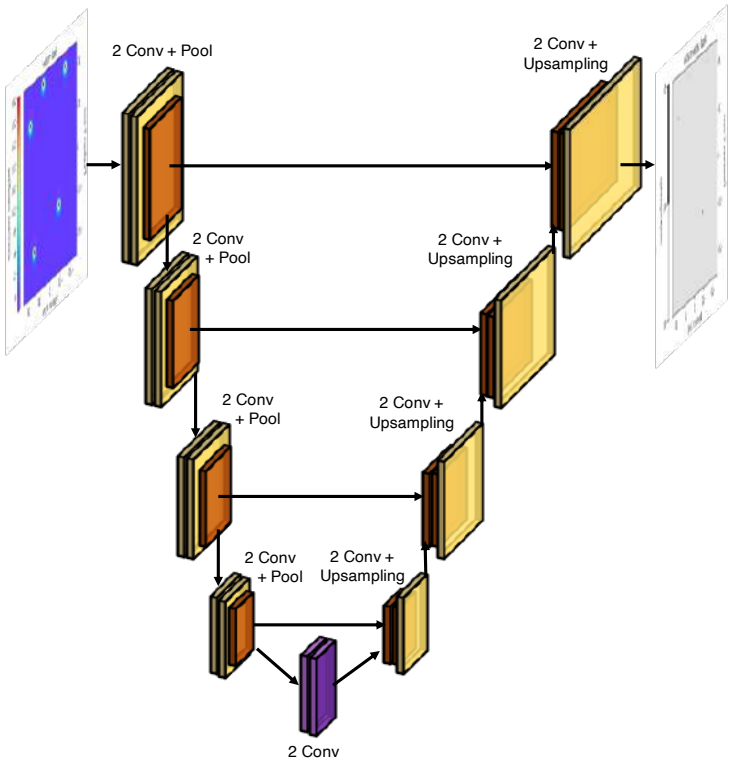


Figure 2.11 – Schema of the U-Net architecture used. The input is a magnetic map and the output is a segmentation of the same image. The network is composed by an encoder network (left-side) and and a decoder network (right-side).

The first results applying the U-Net were outstanding. The model was able to accurately identify the pixels were the center of each anomaly was located (Figure A1-2). This means that it could predict the number of anomalies on a magnetic map and locate each of them. Nevertheless, the model was still limited by the number of dipoles considered in the database. This remains a disadvantage because it is pointless to create an example in the database for every potential number of dipoles that we could measure in a survey. On the other hand, the performance observed in each of the tests leads us to consider U-Net for future geophysical applications which I will address in the perspectives section.

After finding two architectures that could accomplish the desired tasks, and learning about their ad-

vantages and limitations, I continued to search for other architectures that would improve our actual results and lead to a future real-time implementation. First, I discovered the DenseNet architecture which, like VGG16, can be used for regression tasks (Figure 2.12). The application of this architecture reduced the number of trainable parameters from 4 million to 1.5 million. This means that the time required for the learning phase has been considerably reduced, even considering that the network was learning from scratch. Subsequently, I performed some tests with the YOLO architecture, considered as the state-of-the-art in the field of object recognition. I found a version of this network that has already been trained on "ImageNet" (Deng et al., 2009), one of the largest visual database available on internet. This version allowed me to use transfer learning in order to reduce the time spent during the learning phase, since that this version of YOLO has approximately 60 millions trainable parameters. The results showed us that once retrained, the model reaches extraordinary detection and counting capabilities. It was able to accurately detect over 30 dipole anomalies, even though I only considered up to 10 in the database. Finally, after several experiments, I observed that the combination of two models, "YOLO" and "DenseNet" (Figure 2.13), performs best to achieve our classification and regression goals. I first used the "YOLO" architecture to locate and classify each dipolar anomaly. Then, I implemented the "DenseNet" architecture to individually predict the parameters of each dipole detected by the YOLO model.

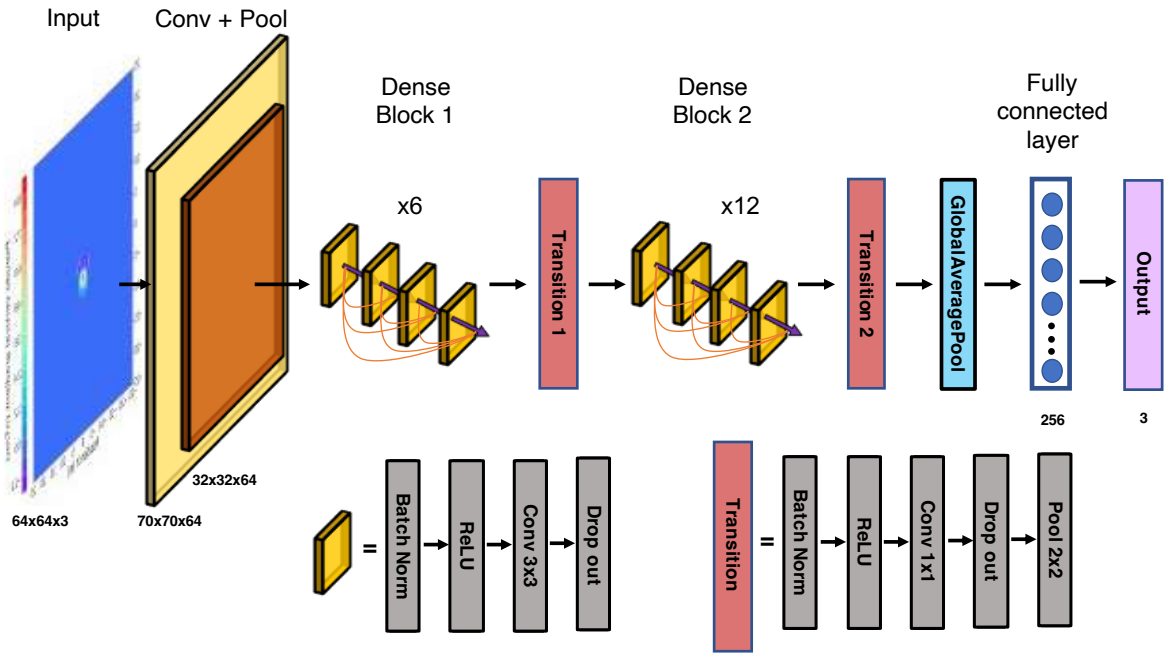


Figure 2.12 – DenseNet network diagram. It is composed of two DenseBlocks with a variable number of layers, respectively [6 and 12] layers.

To summarize, Table 2.1 shows a summary of the architectures tested by order with comments on the advantages and limitations observed.

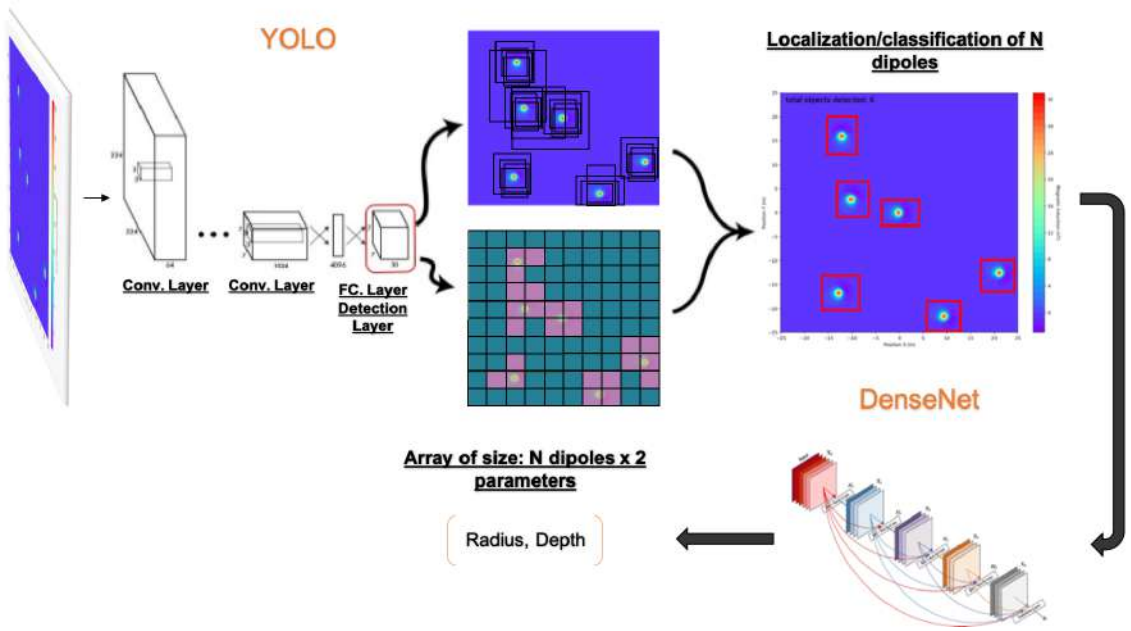


Figure 2.13 – Diagram of our CNN architecture combining two distinct methods (YOLO and DenseNet).

Architecture (predictions)	Experience feedback
1) VGG16 (Parameters)	- Prediction limited to one object. - No position information.
2) VGG16 multi-objects (parameters)	- Slowing down of learning process due to the increase of the number of neurons in output. - No position information.
3) VGG16 multi-objects (parameters + position)	- Slowing down of learning process due to the increase of the number of neurons in output. - Decrease in accuracy due to the complexity of predictions.
4) VGG16 multi-objects (parameters) + U-Net	- Position information (x, y) but inaccurate measurement of the width and length of the objects.
5) YOLO + DenseNet multi-objects (parameters)	- Position information (x, y) and dipolar anomaly detection. - Improvement of learning time and parameter prediction.

Table 2.1 – Overview of all tested combinations among VGG16 (Simonyan and Zisserman, 2015), U-Net (Ronneberger et al., 2015), DenseNet, and YOLO.

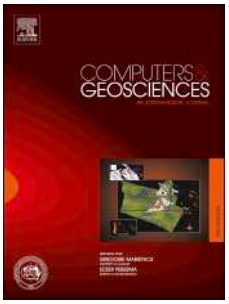
CHAPTER

3 Magnetic Anomalies Characterization

3.1 Introduction to the article in the context of this thesis

The main objectives of this thesis are the non-classical inversion of geophysical measurements, the identification of geological structures and automatic recognition of anomalies. In this chapter, I present the article submitted to the journal "Computer & Geosciences", which is published and available at the following DOI link: https://doi.org/10.1016/j.cageo.2022.105227. Our article presents the characterization of magnetic anomalies using convolutional neural networks, and the application of visualization tools to understand and validate their predictions. The developed approach allows the localization of magnetic dipoles, including counting the number of dipoles, their geographical position, and the prediction of their parameters (magnetic moment, depth, and declination). Our results suggest that the combination of two deep learning models, "YOLO" and "DenseNet", performs best in achieving our classification and regression goals. Additionally, we applied visualization tools to understand our model's predictions and its working principle. We found that the *Grad-CAM* tool improved prediction performance by identifying several layers that had no influence on the prediction and the *t-SNE* tool confirmed the good ability of our model to differentiate among different parameter combinations. Then, we tested our model with real data to establish its limitations and application domain. Results demonstrate that our model detects dipolar anomalies in a real magnetic map even after learning from a synthetic database with a lower complexity, which indicates a significant generalization capability. We also noticed that it is not able to identify dipole anomalies of shapes and sizes different from those considered for the creation of the synthetic database.

Journal Pre-proof



Magnetic anomalies characterization: Deep learning and explainability

J. Cárdenas, C. Denis, H. Mousannif, C. Camerlynck, N. Florsch

PII: S0098-3004(22)00176-5

DOI: https://doi.org/10.1016/j.cageo.2022.105227

Reference: CAGEO 105227

To appear in: *Computers and Geosciences*

Received Date: 12 January 2022

Revised Date: 2 June 2022

Accepted Date: 27 August 2022

Please cite this article as: Cárdenas, J., Denis, C., Mousannif, H., Camerlynck, C., Florsch, N., Magnetic anomalies characterization: Deep learning and explainability, *Computers and Geosciences* (2022), doi: https://doi.org/10.1016/j.cageo.2022.105227.

Magnetic Anomalies Characterization : Deep Learning and Explainability

J. Cárdenas[a,d], C. Denis[a,b], H. Mousannif[a,c], C. Camerlynck[d] and N. Florsch[a,d]

[a]*Sorbonne Université, IRD, Unité de Modélisation Mathématique et Informatique des Systèmes Complexes, UMMISCO, F-93143, Bondy, France.*
[b]*LIP6, Sorbonne Université, 4 place Jussieu, 75252 Paris Cedex, France.*
[c]*LISI Laboratory, FSSM, Cadi Ayyad University, Bd. Prince My Abdellah, B.P. 2390, 40000 Marrakech, Morocco.*
[d]*UMR 7619 METIS, Sorbonne Université, 4 place Jussieu, 75252 Paris Cedex, France.*

ARTICLE INFO

Keywords:
Geophysics
Deep learning
Convolutional neural networks
Magnetic Methods
Explainability
Unexploded ordnance

ABSTRACT

In recent years, deep learning methods have shown great promise in the field of geophysics, especially for seismic interpretation. However, there is very little information with regard to its application in the field of magnetic methods. Our research introduces the use of convolutional neural networks for the characterization of magnetic anomalies. The models developed allow the localization of magnetic dipoles, including counting the number of dipoles, their geographical position, and the prediction of their parameters (magnetic moment, depth, and declination). To go even further, we applied visualization tools to understand our model's predictions and its working principle. The *Grad-CAM* tool improved prediction performance by identifying several layers that had no influence on the prediction and the *t-SNE* tool confirmed the strong capacity of our model to differentiate between different parameter combinations. Then, we tested our modelwith real data to establish its limitations and application domain. Results demonstrate that our model detects dipolar anomalies in a real magnetic map even after learning from a synthetic database with a lower complexity, which indicates a significant generalization capability. We also noticed that it is unable to identify dipole anomalies of shapes and sizes different from thoseconsidered for the creation of the synthetic database. Finally, the perspectives for this work consist of creating a more complex database to approach the complexity traditionally observedin magnetic maps, using real data from multiple acquisition campaigns, and other applications with alternative geophysical methods.

1. Introduction

The first neural networks were developed in the 1950s. However, until the 2000s, there was a lack of sufficient computational power and efficient algorithms in order to take more complex neural network topologies into account, thus enabling improvements in their prediction capabilities. The gradient backpropagation algorithm (Rumelhart et al., 1986) coupled with vastly improved computational power has led to spectacular results in the field of pattern recognition and perception.

Many scientific disciplines are directing their research activities towards deep machine learning. This is also the case in geophysics, where it is used to facilitate the automated processing of geophysical data and the treatment of inversion problems, e.g., the identification of wave trains or the filtering of seismic noise (Van der Baan and Jutten, 2000), or to obtain an estimate of the global thickness of the Earth's crust (Meier et al., 2007).

Abbreviations: CNNs, Convolutional Neural Network; YOLO, You Only Look Once; MSE, Mean Square Error; R&D, Research and Development; IoU, Interception over Union; UXO, Unexploded Ordnance; EMI, Electromagnetic Induction; TDEM, Time Domain Electromagnetic.

julio.cardenas_chapellin@sorbonne-universite.fr (J. Cárdenas); christophe.denis@lip6.fr (C. Denis); mousannif@uca.ma (H. Mousannif); christian.camerlynck@sorbonne-universite.fr (C. Camerlynck); nicolas.florsch@sorbonne-universite.fr (N. Florsch)

In the last decade, deep learning methods have shown great promise, especially in the field of seismic interpretation. For example, Zhao (2019), Wu et al. (2019), and An et al. (2021) use convolutional neural networks (CNNs) forautomatic fault recognition from seismic data sets. Das et al. (2018) also used CNNs to predict an elastic subsurface model directly from recorded seismic data, and Araya-Polo et al. (2018) to produce an accurate layered velocity model from shot gathers. Moreover, deep learning methods have been applied to other geological and geophysical problems,particularly geophysical inversion (e.g., Laloy et al., 2019; Mosser et al., 2018), well log interpretation (e.g., Anemangely et al., 2019; Ali et al., 2021), ground penetrating radar (e.g., Ishitsuka et al., 2018; Travassos et al., 2021), and electrical resistivity data inversion (e.g., Liu et al., 2020).

Magnetic data inversion using deep machine learning is a topic that has been addressed only very recently. A key example of recent studies in this area is the work of Guo et al. (2021), who have used deep learning models to perform inversion of 3D magnetic structures. To do this, a data set containing millions of annotated geological models for each geological structure were generated to feed convolutional neural networks. Their neural network models are used to classify and predict the parameters of a geological structure present in the magnetic maps. However, this approach is limited to the analysis of a single configuration. Furthermore, by analyzing all anomalies instead of each one separately, it multiplies the number of cases to process and thus the prediction time.

Our scientific approach characterizes the magnetic data by counting the number of dipole magnetic anomalies present in a magnetic induction map, and the subsequent prediction of their respective positions (x,y) and associated parameters (magnetic moment, depth and declination).

1.1. Traditional dynamics in magnetic methods in geophysics and perspectives on the application of deep learning

In magnetic methods in geophysics, the physical properties of interest are magnetic susceptibility (induced magnetization) and magnetization density (remanent case). Here we are interested in the induced case, the most frequent. Induced magnetization enables the characterization of the quantity of magnetized material in a primary magnetic field.The magnetized material creates a secondary magnetic field (often called an induced field - but strictly speaking it should be called magnetic induction), with the data measured during a survey representing the superposition of the primary earth field and the secondary induced fields. These data are sometimes interpreted in terms of real geologicalunits or structures (such as faults or magmatic intrusions), or inverted to obtain the distribution of induced magnetic moments below the surface (Figure 1).

In order to interpret magnetic data, geophysicists perform corrections, providing the processed data as a final product. These corrections, performed in part by traditional inversion software, isolate the anomalous field caused by objects of interest buried underground to interpret their magnetic characteristics and depth. Sometimes the situation is

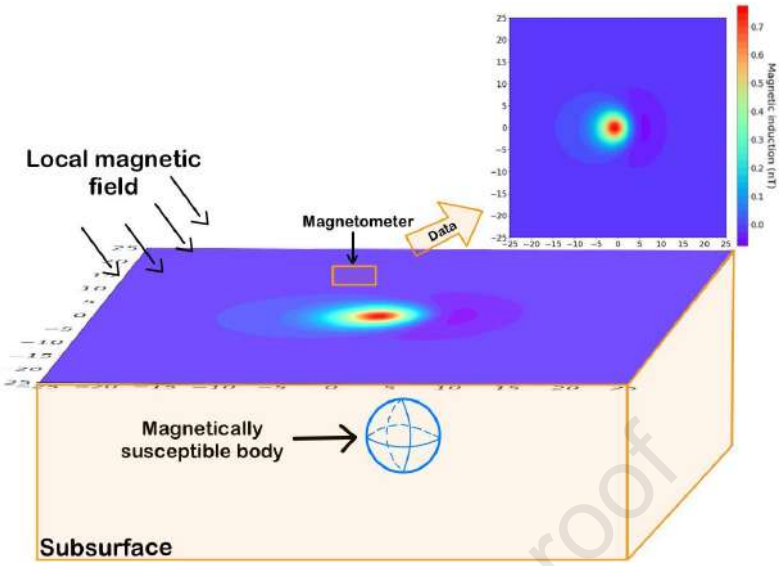


Figure 1: Elements of a magnetic survey: a local magnetic field, a magnetizable body buried in the ground, a measuring instrument (magnetometer) and maps generated after data processing.

complicated for magnetic maps, particularly in instances where the responsible objects have complicated shapes and variable dimensions. Frequently, this process involves a degree of subjectivity and may vary depending on the expertise of the geophysicist. Our approach to the application of deep learning (Figure 2) should overcome the weaknesses of traditional inversion algorithms, and also better handle the various noises affecting the data, which often require fine-tuning of the damping parameters.

Traditional approach

Magnetic prospection

↳ **Collection of observed data**
(magnetic field)

↳ **Data inversion**

↳ **Geophysical interpretation**
(geological structure parameters)

New approach

Synthetic generation of :
- Geological structure parameters (output data)
- Collection of observed data (input data)

↳ **Learning phase of the model**

Real data

} **Geophysical interpretation**

Figure 2: Diagram of the traditional approach in geophysics, and the new approach proposed in this study.

2. Methodology

As we mentioned before, obtaining a map illustrating the distribution of magnetic susceptibility is the result of a field survey. The maps obtained are often imperfect due to multiple factors, such as: instrumental noise, ground regularity conditions, access difficulties, etc. As a consequence, the quantity and quality of geophysical data are limited. In deep learning, the accuracy of the predictions made by neural networks depends to a large extent on the complexity of the architecture and the volume of data available for learning. Contrary to many other domains in which neural networks have been highly successful, such as image classification (e.g., Krizhevsky et al., 2017) where the amount of available data is often more than 1 million examples (e.g., Deng et al., 2009), the availability of geophysical data is a limiting factor, as the number of available real maps is low (a thousand examples at the most). To overcome this difficulty, simulated data is used to train and validate the convolutional neural network. For example, Figure 3 shows the magnetic anomaly generated by a sphere of magnetic moment 52 $A.m^2$, which is located at 1.4 m depth and in the presence of a horizontal primary field.

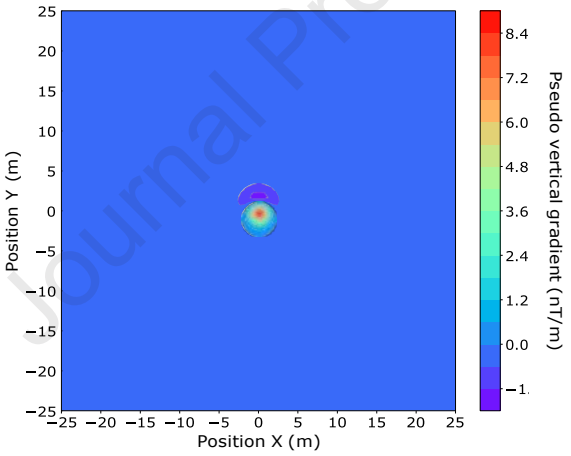


Figure 3: Example of a synthetically generated magnetic anomaly map.

2.1. Generation of synthetic models

By using Equation 8.17 proposed by Scollar (1990) (see Equation 1), which calculates the anomaly created by a dipole buried at depth h for a magnetometer with two superimposed sensors, we were able to generate maps of vertical magnetic pseudo-gradient anomalies for different configurations of magnetic bodies, which are here assimilated into magnetic dipoles. The declination parameter does not appear in Equation 1 because Scollar (1990) ignored it by considering magnetic north to coincide with geographic north. We used data augmentation strategies to rotate each

anomaly to simulate the effect we would observe if the declination parameter was considered. This strategy allowed us to obtain labeled data and to monitor the angle predictions of our model on all the rotations considered.

$$\Delta F_{x,y} = \frac{\mu_o}{4\pi} M \begin{Bmatrix} \frac{1}{(\sqrt{x^2+y^2+(h+H_{bottom})^2})^5}\left[(2(h+H_{bottom})^2 - x^2 - y^2)sin(I) - 3yzcos(I)\right] \\ \frac{1}{(\sqrt{x^2+y^2+(h+H_{top})^2})^5}\left[(2(h+H_{top})^2 - x^2 - y^2)sin(I) - 3yzcos(I)\right] \end{Bmatrix} \quad (1)$$

With:)

- $\Delta F_{x,y}$ = dipole magnetic anomaly located at position (0,0) (nT/m);
- μ_o = magnetic permeability of vacuum (H/m);
- M = magnetic moment of the induced dipole ($A.m^2$);
- h = depth of the dipole to the ground;
- H_{bottom}, H_{top} = depth of the dipole to the low sensor and the high sensor respectively;
- I = Inclination of the dipole (radians).

To generate the synthetic models, we first chose a range of values for the depth and magnetic amplitude parameters. Then, every magnetic anomaly was calculated by considering a point P at position (0,0) on our magnetic map. In order to increase the number of dipoles, we shifted and summed the maps generated with the same distribution of parameters (Figures 5a and 5b), respecting a minimum distance of 2 meters between the dipoles to avoid any coalescence of anomalies. The choice of using the same distributions to sum the maps avoids the occultation of anomalies whose amplitudes would vary too widely. This case, which is often observed in real examples, will be the subject of a later study.

The physical and magnetic parameters of each dipole were used as labeling data for our algorithm. Our data set, consisting of about 10 000 examples, was divided as follows: 60% learning, 20% cross-validation and 20% testing. This distribution is considered as optimal by Chollet (2018), when the data set is limited (number of examples lower than 10^6). To choose the size of the database, we trained our model with different sizes, finding that the best stability of the model is observed from 10^4 examples. We chose this amount of data based on the evolution of the loss and the accuracy values during the training phase. We observed that 10 400 examples were sufficient to satisfy the stability of the model during this phase. The characteristics of this database are described in Table 1.

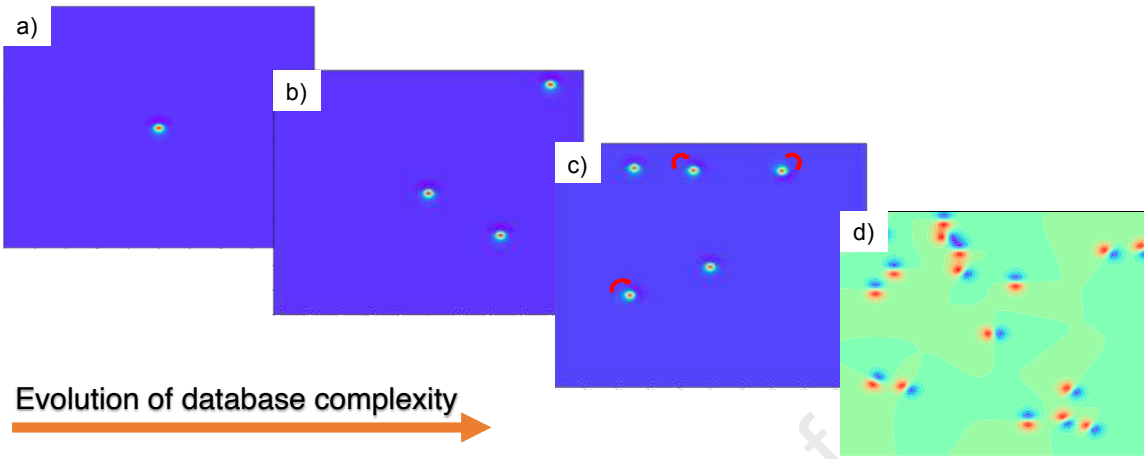


Figure 4: Evolution of the complexity of our database. From left to right, we started with: a) the presence of a single dipole anomaly, b) an increase in the number of anomalies and variations of their positions (x, y), c) variation of the magnetic declination (highlighted with red arrows) and, lastly, d) variation of the magnetic inclination.

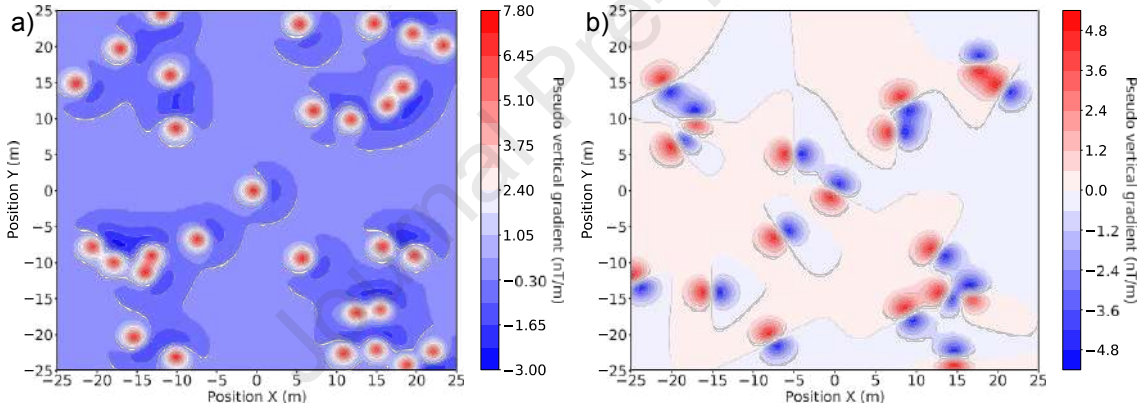


Figure 5: Induced magnetic anomaly maps included in our synthetic data set used for training. The dipole type anomalies correspond to variable declination values and an inclination of 30°(a) and 0°(*b*).

2.2. Architecture selection

At the beginning of our research, in the absence of a specific bibliography on our magnetic application, we decided to use convolutional neural networks ("CNNs") because of the similarity between the classical representation of magnetic data (in the form of a colored map) and color or black/white images. We consider that the images contain a magnetic induction value in each pixel similar to the information of one of the RGB channels of a color image.

This assumption allowed us to consider and evaluate several combinations of CNN architectures, often applied to images, before finding the one that best fits our goals. Table 2 provides a summary of the architectures used in order with comments on the advantages and limitations observed.

Characteristics of the data set		
Magnetic case	**Remanent magnetization**	
Dipole	Not	
Grid		
Size	**Spacing**	
100 m	0.5 m	
Physical characteristics		
Depth	**Radius**	**Sensor height**
[1.0 – 1.6] m	[0.10 – 0.16] m	0.5 m
Magnetic characteristics		
Local field	**Inclination**	**Declination**
47 000 nT	[0, 30, 60, 90] degrees	[0-150] degrees

Table 1
Characteristics used to generate the simulated database.

Architecture (predictions)	Experience feedback
1) VGG16 (Parameters)	- Prediction limited to one object. - No position information.
2) VGG16 multi-objects (parameters)	- Slowing down of learning process due to the increase of the number of neurons in output. - No position information.
3) VGG16 multi-objects (parameters + position)	- Slowing down of learning process due to the increase of the number of neurons in output. - Decrease in accuracy due to the complexity of predictions.
4) VGG16 multi-objects (parameters) + U-Net	- Position information (x, y) but inaccurate measurement of the width and length of the objects.
5) YOLO + DenseNet multi-objects (parameters)	- Position information (x, y) and dipolar anomaly detection. - Improvement of learning time and parameter prediction.

Table 2
Overview of all tested combinations among VGG16 (Simonyan and Zisserman, 2015), U-Net (Ronneberger et al., 2015), DenseNet, and YOLO.

After testing numerous combinations of deep learning models according to the evolution of our database (Figure 4), we observed that the combination of two models, "YOLO" (Redmon and Farhadi, 2018) and "DenseNet" (Huang et al., 2018) (Figure 6), performs best in achieving our classification and regression goals. We first used the "YOLO" architecture to locate and classify each dipole. To perform parameter regression, we then implemented the "DenseNet" architecture to individually predict the parameters of each dipole detected by the YOLO model.

"YOLO" is an efficient object recognition architecture capable of identifying the presence of objects in images. It splits the image into regions and predicts rectangles and probabilities for each of them. The model structure is composed of CSPDarknet-53, Spatial Pyramid Pooling in Deep Convolutional networks (SPPnet), Path Aggregation Network (PANet), and three YOLO heads (YOLOv3). The network contains 53 convolution layers with sizes of 1 × 1 and 3 × 3, with each convolution layer connected with a batch normalization (BN) layer and a Mish activation layer (Li et al., 2020).

The YOLO network predicts N bounding boxes and a confidence score for each box. The classification score will be from ‘0.0‘ to ‘1.0‘, with‘0.0‘ being the lowest confidence level and ‘1.0‘ being the highest. Each of these bounding

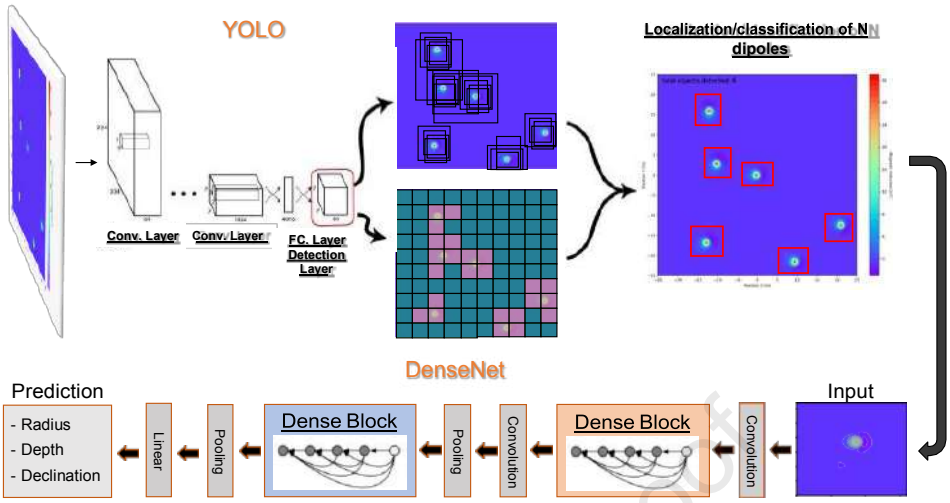


Figure 6: Diagram of our CNN architecture combining two distinct methods (YOLO and DenseNet).

boxes is made up of 5 numbers: the x-center, the y-center, the width, the height, and the confidence. The confidence represents the IOU between the predicted bounding box and the actual bounding box, referred to as the ground truth box. The IOU stands for Intersection Over Union and is the area of the intersection of the predicted and ground truth boxes divided by the area of the union of the same predicted and ground truth boxes. The greater the region of overlap, the greater the IOU. For example, in Figure 7, we have an orange box, and a green box. The orange box represents the ground truth, and the green box represents the prediction obtained from our model. The aim of this model would be to keep improving its prediction, until the orange box and the green box perfectly overlap, i.e., the IOU between the two boxes becomes equal to 1.

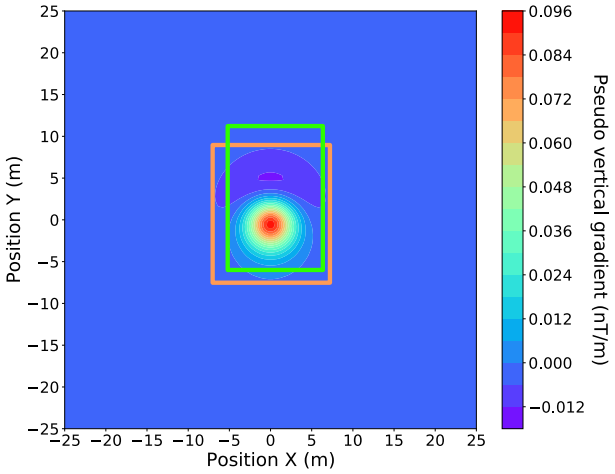


Figure 7: Example of an Intersection Over Union (IOU). The ground truth represented by an orange box and the predicted box by a green box.

Layers	Output Size	DenseNet-121
Convolution	32 x 32	7x7 conv, stride 2
Pooling	16 x 16	3x3 max pool, stride 2
Dense Block (1)	16 x16	x 6
Transition Layer (1)	16 x 16	1 x 1 conv
	8 x 8	2 x 2 average pool, stride 2
Dense Block (2)	8 x 8	x 12
Transition Layer (2)	8 x 8	1 x 1 conv
	4 x 4	2 x 2 average pool, stride 2
Regression Layer	1 x 1	global average pool
	3	256D fully-connected, MSE

Table 3
Detailed parameters of our DenseNet network. Information includes: output size, type of layers, number of filters and kernel size.

"DenseNet" is a type of CNN architecture where each layer is connected to all following layers. This idea alleviates the vanishing gradient problem and encourages feature reuse during learning. Our DenseNet network is composed of two DenseBlocks with a variable number of layers. We can see that the first part consists of a 7x7 stride 2 Convolutional Layer followed by a 3x3 stride 2 MaxPooling layer. Then, the two dense blocks, with [6 and 12] layers respectively, are followed by a Fully Connected Layer that accepts the feature maps of all layers of the network to perform the regression (Figure 8). Detailed information on the parameters is presented in Table 3.

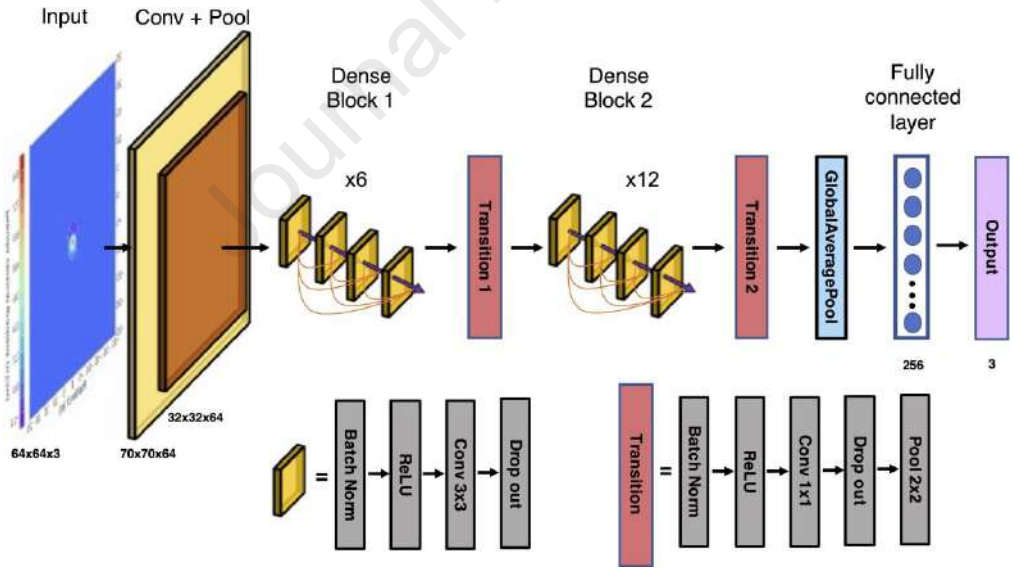


Figure 8: DenseNet network diagram. It is composed of two DenseBlocks with a variable number of layers, respectively [6 and 12] layers.

When we created the synthetic database, we considered the issue of variable length YOLO detection, since DenseNet only accepts a fixed input size. To overcome it, we created a database with multiple anomalies. Then, we cropped each example with different variable-length windows in order to simulate a YOLO bounding box detection. We included

several examples of overlapping anomalies which can be expected in real cases. Finally, we padded all these variable-length windows to a 64x64 shape (Figure 9), which is related to the maximum size an anomaly in our database can cover, and geophysically to the maximum depth.

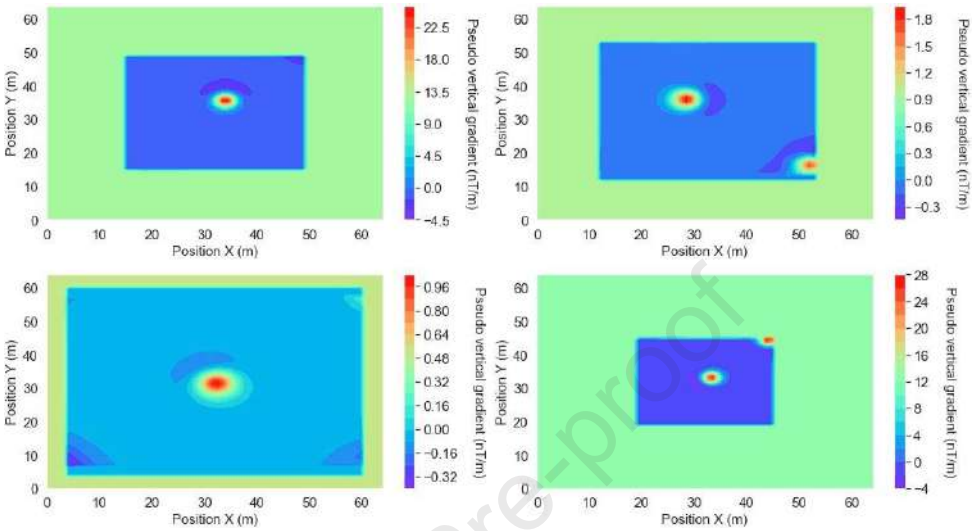


Figure 9: Examples of simulated data created to consider different variable-length windows detected by YOLO. The green filled areas represent the padding applied to have a constant input size of 64x64 for DenseNet.

To improve the performance of our regression model and thus determine a good set of hyperparameters, we used the "hyperband" algorithm (Li et al., 2018) (Figure 10). It consists of speeding up the random search for hyperparameters through adaptive resource allocation and early stopping, which provides a higher speedup than other explorationalgorithms. In this work, we searched for the best combination of the number of dense blocks, the number of fully connected layers, the number of neurons per layer and the learning rate.

Among the hyperparameters, we used "dropout" (Srivastava et al., 2014) for regularization. Multiple optimization functions were compared and "AdaGrad" (Duchi et al., 2011) was selected for its stability and fast convergence.The activation function "tanh" (LeCun et al., 2015) was chosen because of negative and positive values in the input quantities. It is important to specify that this function was applied to all layers except the output layer because, for a regression model, we need continuous values at the end. Finally, we used the coefficient of determination R^2 to calculate the accuracy and the mean square error (MSE) function as the cost function. Table 4 shows the name of the regression model, the values of the hyperparameters, and its performance.

For additional information, some of the terms used above are defined in the list below:

- **Dense block** : module used in convolutional neural networks that connects all layers (with matching feature map sizes) directly with each other. To preserve their feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature maps to all subsequent layers.

Model	Dropout	Learning rate
DenseNet	0.1	0.0001
Activation	**Accuracy**	**Loss**
tanh	0.9998	0.0005

Table 4
Hyperparameters used and performance metrics obtained after training phase.

- **Fully Connected layers** : those layers where all the inputs from one layer are connected to every activation unit of the next layer. In most popular machine learning models, the last few layers are full connected layers which compile the data extracted by previous layers to form the final output.
- **Dropout** : regularization method that approximates the training of a large number of neural networks with different architectures in parallel. During training, a certain number of layer outputs are randomly ignored or "dropped out". This has the effect of making the layer resemble(and be treated like) a layer with a different number of nodes and connectivity to the prior layer.
- **Adagrad** : algorithm for gradient-based optimization that adapts the learning rate to the parameters, performing smaller updates (i.e. low learning rates) for parameters associated with frequently occurring features, and larger updates (i.e. high learning rates) for parameters associated with infrequent features.

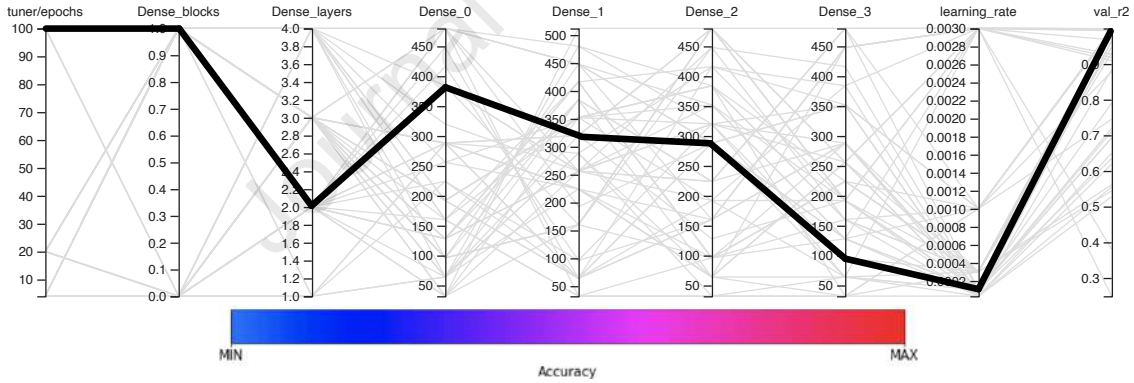


Figure 10: Selection process of hyperparameters. We highlighted the best combination of hyperparameters with a wider black line. Total elapsed time : 2 hours 54 minutes on google colab with gpu.

2.3. Data labeling

We performed a labeling process following the generation of the geological models. Since two deep learning models will be used, the formats adopted for the output data are: the "YOLO" labeling format for the detection and classification model, and the magnetic real values for the regression model.

The "YOLO" labeling format consists of the creation of a .txt file that contains the annotations for the corresponding image, i.e., the class (in our case: dipolar anomaly) and the values of the rectangle ("Bounding Box") around each dipolar anomaly.

CNNs are particularly well suited for image data analysis. In addition to using them for classification, they can also be used to predict continuous data. In our research, these data are the parameters of the magnetic dipoles (declination, depth and amplitude of the magnetic field).

3. Explainability of algorithms

Using machine learning raises different types of questions concerning its operationality. In the context of our application in geophysics, there is a strong demand to explain results produced by machine learning models. The explanation can be intended for example for developers or engineers in R&D who most often use machine learning in "black box" mode. The explanation can also be intended for academic researchers to improve scientific knowledge because of the limited nature of prediction without explanation.

In this contribution, the need for explanation was motivated for validation purposes to select a model from a range of other possibilities with a similar statistical performance. We used computer tools to obtain visual information:

- The discriminatory zone of our network was visualized using the *Grad-CAM* tool (Selvaraju et al., 2020);
- The *t-SNE* tool (Van der Maaten and Geoffrey, 2008) was used to reduce the dimensionality of the data.

Grad-CAM is a technique for producing "visual explanations" for decisions in a large class of CNN-based models. This tool identifies the gradient information flowing through the last convolutional layer of the CNN to understand what each neuron is interested in. On the other hand, *t-SNE* is another dimensionality reduction technique and is particularly well suited for visualizing a high-dimensional data set.

4. Results and Discussion

We conducted multiple experiments to test the robustness of our model and to evaluate its generalization capacity. In order to do this, we took into account the variation of the physical and magnetic characteristics of the dipoles (Table 1), the noise level and the number of dipoles present in the model.

4.1. Dipoles characterization

We observed that the YOLO method produces very good performances, reaching a mean average precision of 96.49% (considering an IoU threshold of 50%). Although our model was trained with a limited number of dipoles (between 1 and 8), its generalization capability allows it to identify up to 30 dipoles (Figure 11). Specific metrics obtained after the training phase are presented in Table 5.

Regarding the DenseNet network, we observe a good level of accuracy when we use the test data to compare predicted values with real values (Figure 12). We also detected high performance at the end of the training phase (R^2

	True Positive	False positive	False Negative	Average IoU	Precision	Recall
For confidence threshold = 0.25	11 109	452	510	83.19%	0.96	0.96

Table 5
Specific metrics obtained for our YOLO model at the end of the training phase. *Precision* is the degree of exactness of the model in identifying only relevant objects. It is the ration of True Positives over all detections made by the model. *Recall* measures the ability of the model to detect all ground truths.

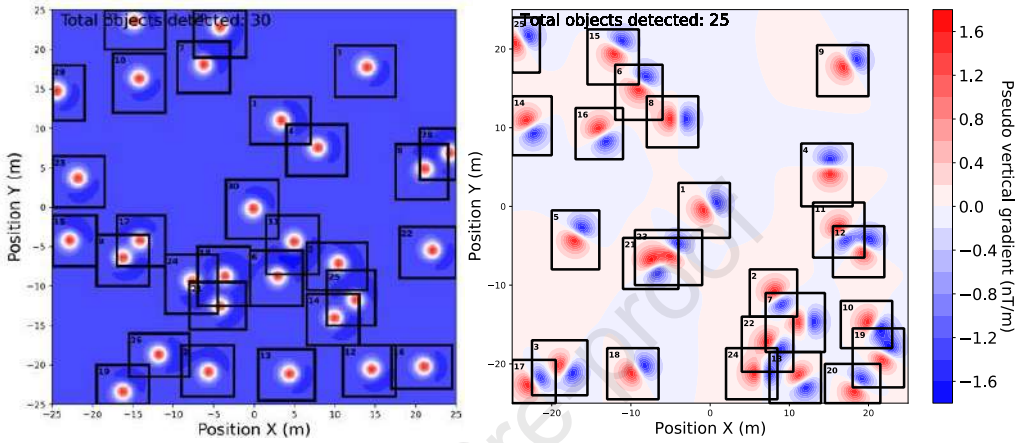


Figure 11: YOLO model predictions for two magnetic inclinations of 60° (a) and 0° (b). The algorithm locates each anomaly by identifying it in a rectangle. The total number of detected dipoles is 30 on the left and 25 on the right; these values are written in the upper left corner of each figure.

score higher than 95%) (Figure 13). These results can be explained by the fact that the network only analyzes the zones containing one anomaly of interest, previously located by the YOLO network. However, we observe that the performance of both networks decreases as from the presence of 10 dipoles (Figure 13). This decrease occurs because a larger number of dipoles increases the coalescence of anomalies, as well as the probability of having an anomaly near the edges of the modeled area.

Furthermore, it is important to mention that the results obtained correspond to a data set of a certain complexity and without noise. The detection of an anomaly in the presence of a magnetic background is much more complex than exceeding a constant threshold (Butler, 2001). Therefore, we added Gaussian noise to our data set in a first attempt to simulate natural magnetic noise. Although this noise is basic in comparison, its study is important to evaluate the ability of our models to fit new data correctly, drawn from the same distribution as the one used for their creation.

4.2. Noise impact

To evaluate the new noisy magnetic models, we used our previously trained model. We observe an alteration in the shape of the dipoles as the noise level increases and a considerable variation in the magnetic induction values. These changes could reduce the performance of our model because it initially depends on the model's ability to detect each object. However, with regard to detection, the model manages to predict the exact number and the corresponding

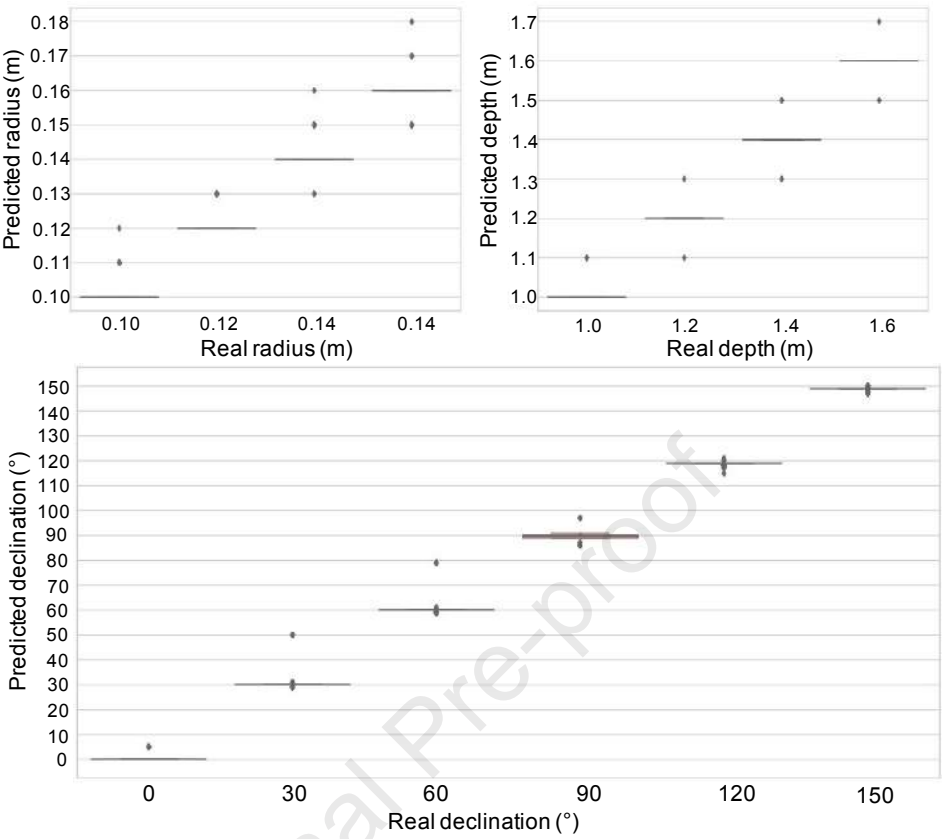


Figure 12: DenseNet predictions of parameters (radius, depth and declination) on the test data.

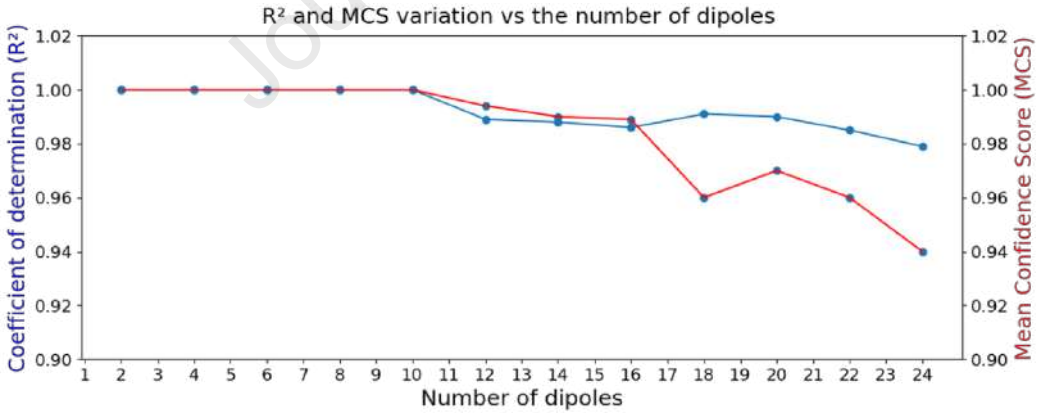


Figure 13: Variation of the coefficient of determination R^2 (DenseNet) on the left and the average confidence score (YOLO) on the right versus the number of dipoles present in the model.

rectangles of each dipole (Figure 14) with a confidence higher than 92%. On the other hand, the model predicts the parameters of each dipole with a coefficient of determination higher than 96%. These results allow us to deduce that our model is robust up to this noise level.

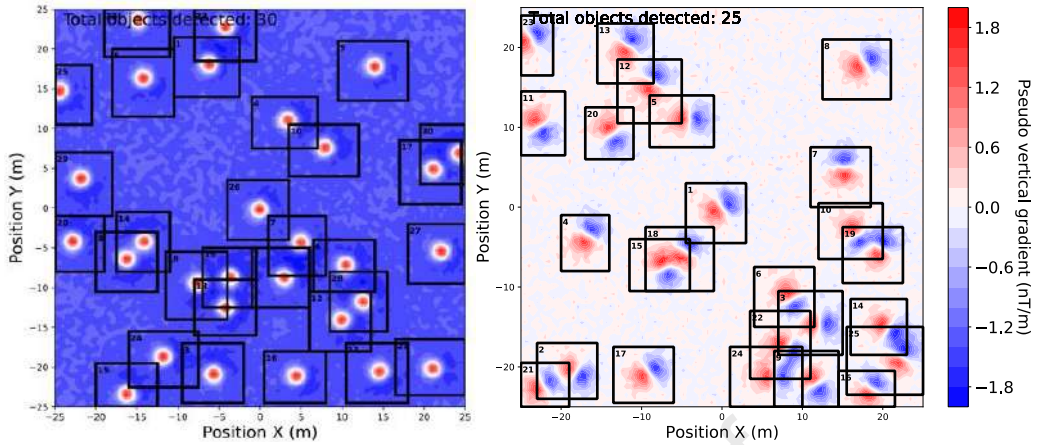


Figure 14: YOLO model predictions for 2 magnetic inclinations of 30°(a) and 0°(b) with Gaussian noise. The total detected dipoles are a) 30 and b) 25.

4.3. Visual explanation of the results

As mentioned in Section 4, in addition to using statistical measurements to evaluate the performance of our model, the application of visual tools is important in order to understand its operationality and explain its working principle, which is essential for its operational acceptability.

Initially, applying the Grad-CAM tool to the first version of our current model, we observed a blurry heat map in the last layer of neurons that did not display the targeted discriminatory area (Figure 15, upper part). This map suggested that we should remove several layers that probably had no influence on the predictions. Following this strategy, we found a layer with a consistent discriminatory area (Figure 15, lower part).

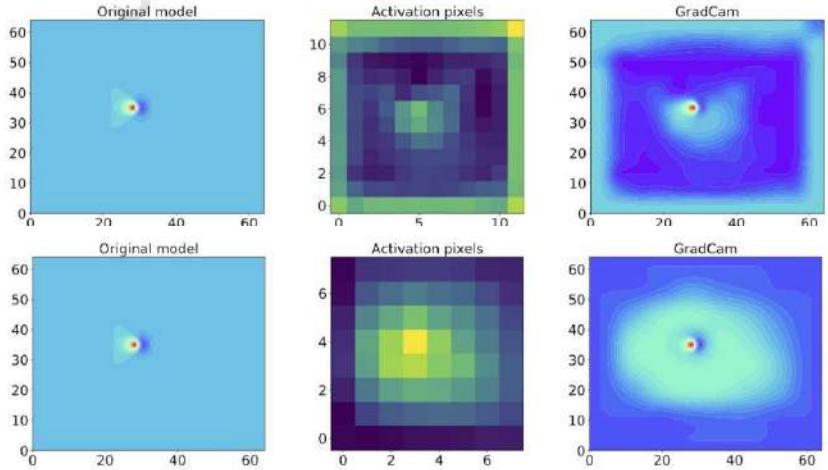


Figure 15: Identification of discriminatory zones for a) an older version and b) a recent version of the same model. From top to bottom, we show first the analyzed geophysical model, second the activation pixels identified by *Grad-CAM*, and third the discriminatory zone.

This change improved the stability of the learning curve, the performance of the predictions, and significantly decreased the number of parameters to be trained from the model, thus decreasing the computation time of the learning phase.

In addition to the ability to discriminate, another important feature to evaluate is the ability of the model to discriminate the parameters used to create the data set. For this evaluation, the data set consisted of 16 possible combinationsof depth and radius values (Table 1) and we applied the *t-SNE* tool to the training (Figure 16a), cross-validation (Figure16b) and test (Figure 16c) data set. Following this application, we observe that the predictions of these parameters were not random. The generated graphs confirm that the model manages to differentiate the 16 combinations. Although weobserve small clusters that the model fails to attach to a larger group (Figure 16a) and erroneous point classificationsin the test data (Figure 16c), the results confirm the clear ability of our model to differentiate among the different parameter combinations.

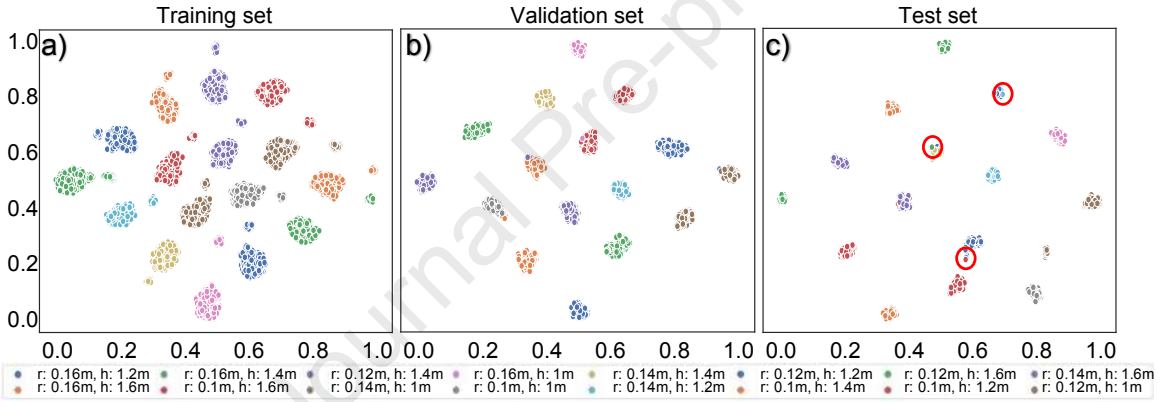


Figure 16: Example of graphs obtained by using the *t-SNE* tool on the (a) training, (b) cross-validation and (c) test data set. The axes have no specific unit and the colors of the clusters represent the 16 possible combinations among the parameters used to create the data set. We use red circles to highlight erroneous point classifications.

4.4. Real-world test case

Up to this point, we obtained an excellent prediction accuracy i.e., our models are able to properly characterize synthetic dipoles in both noiseless and noisy maps. However, the complexity of our synthetic data collection is lower compared to real data, due to the presence of environmental noise, other geological bodies and anthropogenic artifacts. Therefore, it is essential to use real data to evaluate our model's ability to properly adapt to new unseen data and to have an idea of its range of application. In addition, this step is important to consider a future application in data acquisition campaigns.

We chose Unexploded Ordnance (UXO) detection as a real-world application because it strongly relies on mag-

netometry (among other methods as EMI or TDEM) and whose UXO magnetic signature is close to a dipole. Soils contaminated with these objects are a major concern because they impede civilian land use, threaten public safety and are the source of a significant number of environmental worries. As a result, many studies have been carried out to discriminate and identify these objects (e.g., Billings et al., 2002; Billings et al., 2010), including the use of machine learning models (e.g., Bray and Link, 2015; Heagy et al., 2020)

In this paper, we present a real example obtained from a geophysical survey using a fluxgate magnetometer to produce a high resolution map of the magnetic field. The data observed in the map have been anonymized for confidentiality reasons. Figure 17 shows the map obtained after statistical processing to improve the visualization and dipole behavior of the magnetic anomalies.

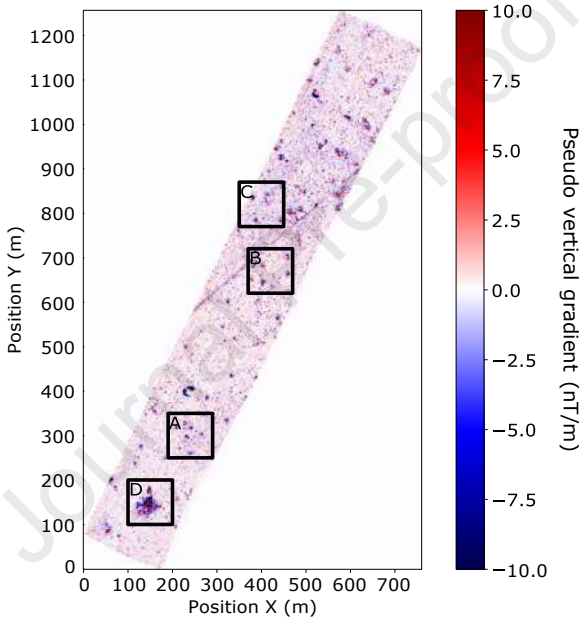


Figure 17: Example of a magnetic map obtained from a geophysical survey. The range of pseudo vertical gradient values was modified to better observe the positive and negative poles of metallic objects. Each black box corresponds to the areas selected to evaluate our model. Our model was able to identify 8 (a), 12 (b), 12 (c) and 22 (d) potential dipolar anomalies.

We selected 4 areas containing multiple dipolar anomalies for prediction. Therefore, we filled the unexplored areas with average values to avoid possible errors during prediction.

Results show that our model is able to identify 8 (Figure 18.a), 12 (Figure 18.b), 12 (Figure 18.c), and 22 (Figure 18.d) potential dipolar anomalies. Even in the presence of unreachable areas, shown in the upper left corner of the map (Figure 18.c), our model manages to locate multiple anomalies. These results demonstrate that our model detects dipolar anomalies in a real magnetic map even after learning from a synthetic database with a lower complexity, which indicates a significant generalization capability. In addition, we note that areas without data do not potentially affect

the predictions, a very important finding as this is a limitation frequently observed in acquisition campaigns. On the other hand, these results give us an idea of the application domain of our current model. In the three chosen areas, we observe that the model fails to detect small dipole anomalies and those with different shapes from the one used in our database. These results are totally expected because, when we created our synthetic database, we did not consider near-surface anomalies. In the case of pyrotechnic prospecting, what is on the surface is pollution in general and is not of interest. In addition, the anomaly detections and parameter predictions in zone D are presented in Figure S1 and Table 1 respectively in the "supplementary material" section. Comparative analysis with the real parameters will be the subject of a future application article.

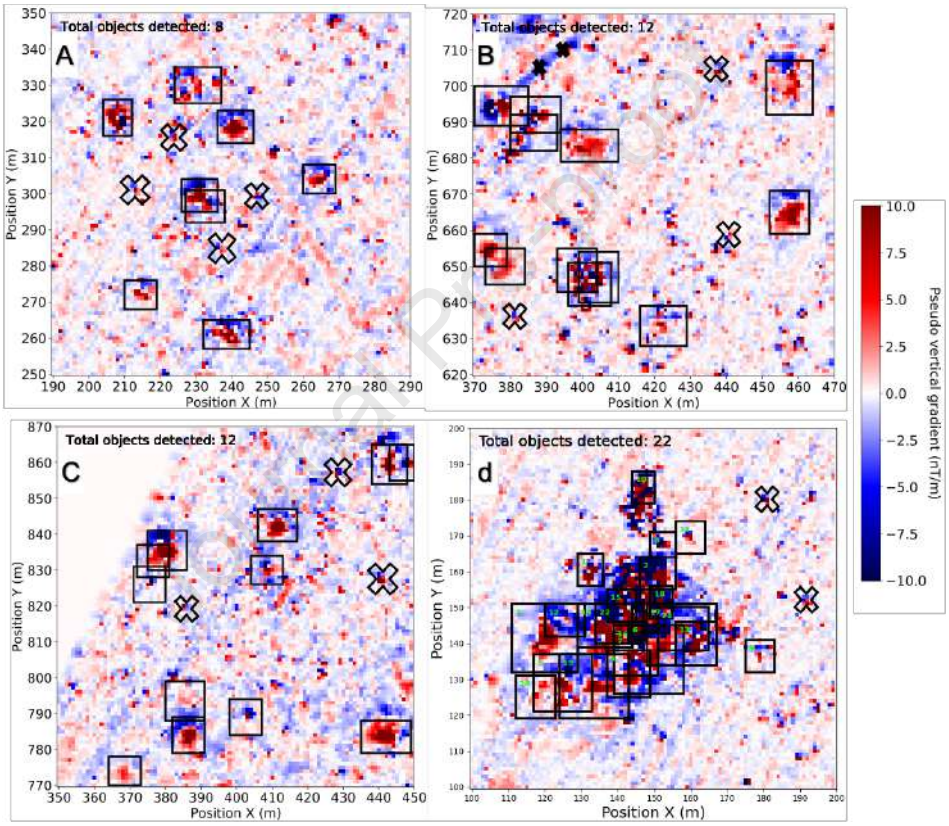


Figure 18: Detection results on the four selected areas. Our model was able to identify 8 (a), 12 (b), 12 (c) and 22 (d) potential dipolar anomalies. Each of them is highlighted with its respective bounding box (black box) generated by our YOLO model. In addition, we added black crosses to indicate some examples of unidentified anomalies.

Finally, the prediction time of our model is about 20 seconds per each area of 50x50m^2, which would make it compatible with a real time application during a survey.

5. Conclusions

In this paper, we present a new methodology based on machine learning, to detect magnetic anomalies present in a prospected area. We then predict the physical and magnetic characteristics of each detected magnetic response perceived as dipolar. A synthetic data set was generated by simulation to compensate for the limited number of labeled geophysical measurements. Our experiments have shown that the coupling of two models, YOLO and DenseNet, provides the best performance. The YOLO model enables the detection of each dipole while the DenseNet network estimates the parameters of each identified dipole with an accuracy higher than 90%.

Beyond the statistical performance, we have optimized the model so that the discriminatory zone, visualized by the *Grad-CAM* tool, is consistent with the expected results to avoid the creation of statistical artifacts. It should be noted that this makes it easier to explain the results obtained. In particular, we optimized the number of model parameters to increase its stability. This is therefore a first step towards the functional verification of the black box of our model.

The use of the *t-SNE* tool allowed us to evaluate the ability of the model to differentiate among all possible combinations of parameters present in the data set. These results show that these visualization tools are essential allies for studying the robustness and stability of a deep learning model by going beyond the measurement of statistical accuracy.

Finally, the application of our model to a real magnetic anomaly map showed that it is able to detect multiple dipole anomalies in four different areas, even in the presence of zones with no data. These results show the potential capability of using a model that has been trained with synthetic data in real applications. Furthermore, it is important to evaluate the performance of the model with real data in order to understand its prediction limits and its application domain.

The perspectives for this work involve creating a more complex database to approach the complexity traditionally observed in magnetic maps. In addition, the consideration of a wider range of parameters and multiple dipole shapes will allow us to contemplate a more realistic application. This application involves sites in the presence of a natural magnetic background, e.g., sites with little or intense noise or with significant geological anomalies. The prediction of the real parameters (declination, amplitude and depth) will be the subject of a future study in which we will compare the prediction made with a traditional inversion program and those obtained by our neural network model. It will be necessary to explain the predictions made by our models using post-hoc analysis techniques. These explanations will be qualitatively evaluated by a panel of geophysical experts. Transfer learning techniques will also be important, for instance, for the application of domain adaptation techniques and the inclusion of real examples in our future databases.

Authorship statement

Julio Cárdenas is responsible for conceptualizing and evolution of research ideas, development and design of methodology, programming, implementation of the computer code and supporting algorithms, data curation, visualization, writing and editing the manuscript. Christophe Denis is responsible for refining research ideas, verification and explainability of results, conducting a research and investigation process, review and editing the manuscript. HajarMousannif is responsible for formulation and evolution of ideas, implementation of the computer code, validation ofoutputs, conducting a research process. Christian Camerlynck is responsible for formulation and evolution of ideas, funding acquisition, validation, conducting a research process, visualization, writing and editing the manuscript. Nicolas Florsch is responsible for formulation and evolution of ideas, funding acquisition, validation, conducting a researchprocess, writing and editing the manuscript.

Acknowledgments

This work is part of the author's doctoral thesis, financially supported by the grant awarded under the international doctoral program created by the convention between IRD-UMMISCO, Sorbonne Université (Paris, France), and Université Cadi Ayyad (Marrakech, Morocco). In addition, we are grateful for the valuable collaboration with UXOMAPcompany (France), which allowed us to test our models on a collection of real-world examples and gave us ideas on potential field applications.

Code availability section

Name of the code : YOLO, Densenet, *Grad-CAM* and *t-SNE*

Developers : Julio Cárdenas

Program language: Python 3.6

Software/Hardware required: All scripts are available to be executed in Google Colab.

Program size: 3.3 mb, 866 kb, 197kb, and 198kb

All source codes are available for download at the following link: https://github.com/juliocardenas25/Magnetic_methods

References

Ali, M., Jiang, R., Ma, H., Pan, H., Abbas, K., Ashraf, U., Ullah, J., 2021. Machine learning - A novel approach of well logs similarity based on synchronization measures to predict shear sonic logs. Journal of Petroleum Science and Engineering 203, 108602. doi:10.1016/j.petrol.2021.108602.

An, Y., Guo, J., Ye, Q., Childs, C., Walsh, J., Dong, R., 2021. Deep convolutional neural network for automatic fault recognition from 3D seismic datasets. Computers & Geosciences 153, 104776. doi:10.1016/j.cageo.2021.104776.

Anemangely, M., Ramezanzadeh, A., Amiri, H., Hoseinpour, S.A., 2019. Machine learning technique for the prediction of shear wave velocity using petrophysical logs. Journal of Petroleum Science and Engineering 174, 306–327. doi:10.1016/j.petrol.2018.11.032.

Araya-Polo, M., Jennings, J., Adler, A., Dahlke, T., 2018. Deep-learning tomography. The Leading Edge 37, 58–66. doi:10.1190/tle37010058.1.

Billings, S.D., Pasion, L.R., Beran, L., Lhomme, N., Song, L.P., Oldenburg, D.W., Kingdon, K., Sinex, D., Jacobson, J., 2010. Unexploded ordnance discrimination using magnetic and electromagnetic sensors: Case study from a former military site. Geophysics 75, B103–B114. doi:10.1190/1.3377009.

Billings, S.D., Passion, L.R., Oldenburg, D.W., 2002. UXO Discrimination and Identification Using Magnetometry, in: Symposium on the Application of Geophysics to Engineering and Environmental Problems 2002, Environment and Engineering Geophysical Society. pp. 12UXO4–12UXO4. doi:10.4133/1.2927059.

Bray, M.P., Link, C.A., 2015. Learning Machine Identification of Ferromagnetic UXO Using Magnetometry. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8, 835–844. doi:10.1109/JSTARS.2014.2362920.

Butler, D.K., 2001. Potential fields methods for location of unexploded ordnance. The Leading Edge 20, 890–895. doi:10.1190/1.1487302.

Chollet, F., 2018. Deep Learning with Python. Manning Publications Co, Shelter Island, New York, 504pp.

Das, V., Pollack, A., Wollner, U., Mukerji, T., 2018. Convolutional neural network for seismic impedance inversion, in: SEG Technical Program Expanded Abstracts 2018, Society of Exploration Geophysicists, Anaheim, California. pp. 2071–2075. doi:10.1190/segam2018-2994378.1.

Deng, J., Dong, W., Socher, R., Li, L.J., Kai Li, Li Fei-Fei, 2009. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL. pp. 248–255. doi:10.1109/CVPR.2009.5206848.

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159.

Guo, J., Li, Y., Jessell, M.W., Giraud, J., Li, C., Wu, L., Li, F., Liu, S., 2021. 3D geological structure inversion from Noddy-generated magnetic data using deep learning methods. Computers & Geosciences 149, 104701. doi:10.1016/j.cageo.2021.104701.

Heagy, L.J., Oldenburg, D.W., Pérez, F., Beran, L., 2020. Machine learning for the classification of unexploded ordnance (UXO) from electromagnetic data, in: SEG Technical Program Expanded Abstracts 2020, Society of Exploration Geophysicists, Virtual. pp. 3482–3486. doi:10.1190/segam2020-3428369.1.

Huang, G., Liu, Z., Van der Maaten, L., Weinberger, K.Q., 2018. Densely Connected Convolutional Networks. arXiv:1608.06993 .

Ishitsuka, K., Iso, S., Onishi, K., Matsuoka, T., 2018. Object Detection in Ground-Penetrating Radar Images Using a Deep Convolutional Neural Network and Image Set Preparation by Migration. International Journal of Geophysics 2018, 1–8. doi:10.1155/2018/9365184.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Communications of the ACM 60, 84–90. doi:10.1145/3065386.

Laloy, E., Linde, N., Ruffino, C., Hérault, R., Gasso, G., Jacques, D., 2019. Gradient-based deterministic inversion of geophysical data with generative adversarial networks: Is it feasible? Computers & Geosciences 133, 104333. doi:10.1016/j.cageo.2019.104333.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. arXiv:1603.06560 .

Li, Y., Wang, H., Dang, L.M., Nguyen, T.N., Han, D., Lee, A., Jang, I., Moon, H., 2020. A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving. IEEE Access 8, 194228–194239. URL: https://ieeexplore.ieee.org/document/9238023/, doi:10.1109/ACCESS.2020.3033289.

Liu, B., Guo, Q., Li, S., Liu, B., Ren, Y., Pang, Y., Guo, X., Liu, L., Jiang, P., 2020. Deep Learning Inversion of Electrical Resistivity Data. IEEE Transactions on Geoscience and Remote Sensing 58, 5715–5728. URL: https://ieeexplore.ieee.org/document/8994191/, doi:10.1109/TGRS.2020.2969040.

Meier, U., Curtis, A., Trampert, J., 2007. Fully nonlinear inversion of fundamental mode surface waves for a global crustal model. Geophysical Research Letters 34. doi:10.1029/2007GL030989.

Mosser, L., Dubrule, O., Blunt, M.J., 2018. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. arXiv:1806.03720 .

Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767 .

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597 . Rumelhart,

D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. URL: http://www.nature.com/articles/323533a0, doi:10.1038/323533a0.

Scollar, I. (Ed.), 1990. Archaeological prospecting and remote sensing. Topics in remote sensing, Cambridge University Press, Cambridge [England] ; New York, 674pp.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. International Journal of Computer Vision 128, 336–359. doi:10.1007/s11263-019-01228-7.

Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 .

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958.

Travassos, X.L., Avila, S.L., Ida, N., 2021. Artificial Neural Networks and Machine Learning techniques applied to Ground Penetrating Radar: A review. Applied Computing and Informatics 17, 296–308. doi:10.1016/j.aci.2018.10.001.

Van der Baan, M., Jutten, C., 2000. Neural networks in geophysical applications. Geophysics 65, 1032–1047. doi:10.1190/1.1444797.

Van der Maaten, L., Geoffrey, H., 2008. Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research 34, 2579–2605.

Wu, X., Liang, L., Shi, Y., Fomel, S., 2019. FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation. Geophysics 84, IM35–IM45. doi:10.1190/geo2018-0646.1.

Zhao, T., 2019. 3D convolutional neural networks for efficient fault detection and orientation estimation, in: SEG Technical Program Expanded Abstracts 2019, Society of Exploration Geophysicists, San Antonio, Texas. pp. 2418–2422. doi:10.1190/segam2019-3216307.1.

Highlights

Magnetic Anomalies Characterization : Deep Learning and Explainability

J. Cárdenas, C. Denis, H. Mousannif, C. Camerlynck, N. Florsch

- YOLO and DenseNet combination allows locating synthetic dipoles, as well as predicting their parameters.
- Noisy data up to a certain level of complexity do not affect the characterization of synthetic dipoles.
- *Grad-CAM* tool identifies unused layers and neurons.
- *t-SNE* tool displays an idea of the logic behind the predictions.
- The complexity of the synthetic database allows our neural network to perform well on real data.

Supplementary material on the paper titled: "Magnetic Anomalies Characterization : Deep Learning and Explainability".

Authors: J. Cárdenas [a,d], C. Denis [b], H. Mousannif [c], C. Camerlynck [d] and N. Florsch[a]

[a]Sorbonne Université, IRD, Unité de Modélisation Mathématique et Informatique des Systèmes Complexes, UMMISCO, F-93143, Bondy, France.

[b]LIP6, Sorbonne Université, 4 place Jussieu, 75252 Paris Cedex, France.

[c]LISI Laboratory, FSSM, Cadi Ayyad University, Bd. Prince My Abdellah, B.P. 2390, 40000 Marrakech, Morocco.

[d]UMR 7619 METIS, Sorbonne Université, 4 place Jussieu, 75252 Paris Cedex, France.

1. Detections and predictions of the parameters at region D (X: 100-200; Y: 100-200) of the real map :

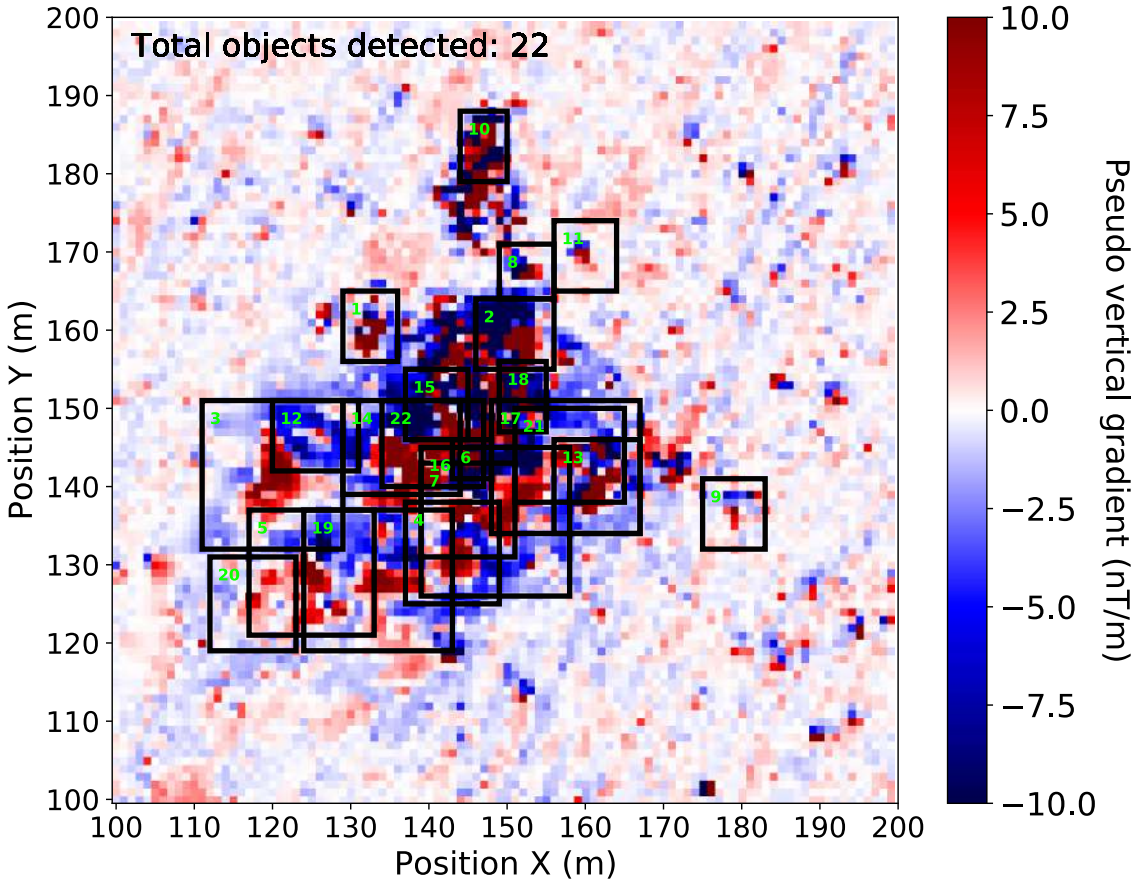


Figure S1 : YOLO model detections at region D (X: 100-200; Y: 100-200) of the real map. Each dipole anomaly is labeled with a green number for comparison with its corresponding parameters predicted by DenseNet.

	Declination (°)	Radius (m)	Depth (m)
1	19	0.13	1
2	60	0.18	1
3	59	0.12	1.5
4	135	0.12	1
5	87	0.12	1.4
6	75	0.51	2.5
7	21	0.18	1
8	120	0.13	1
9	91	0.09	1
10	114	0.27	1.5
11	80	0.09	1
12	28	0.11	1
13	124	0.13	1
14	76	0.17	1
15	15	0.18	1.1
16	104	0.55	2.6
17	93	0.22	1.2
18	42	0.24	1.3
19	30	0.13	1.4
20	114	0.09	1.2
21	29	0.2	1.1
22	45	0.54	2.6

Table 1 : Parameters predicted by our DenseNet model at region D (X: 100-200; Y: 100-200) of the real map.

3.2 Main results of this article

— YOLO and DenseNet combination allows locating synthetic dipoles, as well as predicting their parameters. The model is able to detect far more dipolar anomalies than the amount considered in the training base.

— Noisy data up to a certain level of complexity do not affect the characterization of synthetic dipoles.

— Grad-CAM tool identifies unused layers and neurons. It is a good debugging method to improve the performance of a model.

— t-SNE tool displays an idea of the logic behind the predictions.

— The complexity of the synthetic database allows our neural network to perform well on real data.

3.3 Comparison with geophysical programs

An essential initiative to improve the development of our deep neural model is its potential use during magnetic surveys in order to have a better understanding of the dipolar anomalies displayed on a magnetic map. Therefore, a comparison between the current version of our model and the geophysical programs used nowadays is a key step to progress towards this future application.

Multiple analyses of real magnetic maps must be performed by both geophysical programs and our deep neural network model in order to obtain a fair comparison. As access to the real data began at the conclusion of my research, I propose that these deep analyses will be part of the perspective of this thesis. Nevertheless, some features are commented below :

— Real time application : Our model can detect dipole anomalies and make predictions on large magnetic maps in a short time. Moreover, our script is written in Python, which facilitates an eventual integration with an instrument software. On the other hand, analyses by geophysical programs can be carried out after a completed survey. The advantage of this function is the possibility to visualize the underground geological bodies in order to take decisions regarding the course of a survey.

— Possible anomaly classification : A remarkable advantage of the YOLO model in image recognition is the classification of more than a thousand objects. In the geophysical literature, it is not very clear whether a metallic object can be classified by its magnetic signature. In consequence, further work on our model will consist on the classifica-

tion of UXOs. The information on the type of object is retrieved after the cleanup of a prospected area, which makes it possible to integrate it into our model.

— User-friendly : Geophysical programs are heavy-duty, and all require a good amount of study before they can be operated correctly. Our model appears as a possible solution to observe the geophysical interpretation in real time, showing the geophysical information already learned by the model.

— Deep understanding of the results : Geophysical programs are based on traditional magnetic characterization methods, and it is not always easy to understand why the program proposes a particular solution. Advances on explainable strategies on deep neural networks allow to understand the behavior of a model and to be transparent about the results and accuracy of predictions. This feature could be an important step to better understand magnetic interpretation.

3.4 Conclusion of the article

In this article, we present a new methodology based on machine learning, to detect magnetic anomalies present in a prospected area. We then predict the physical and magnetic characteristics of each detected magnetic response perceived as dipolar. A synthetic data set was generated by simulation to compensate for the limited number of labeled geophysical measurements. Our experiments have shown that the coupling of two models, YOLO and DenseNet, provides the best performance. The YOLO model enables the detection of each dipole while the DenseNet network estimates the parameters of each identified dipole with an accuracy higher than 90%.

Beyond the statistical performance, we have optimized the model so that the discriminatory zone, visualized by the *Grad-CAM* tool, is consistent with the expected results to avoid the creation of statistical artifacts. It should be noted that this makes it easier to explain the results obtained. In particular, we optimized the number of model parameters to increase its stability. This is therefore a first step towards the functional verification of the black box of our model.

The use of the *t-SNE* tool allowed us to evaluate the ability of the model to differentiate among all possible combinations of parameters present in the data set. These results show that these visualization tools are essential allies for studying the robustness and stability of a deep learning model by going beyond the measurement of statistical accuracy.

Finally, the application of our model to a real magnetic anomaly map showed that it is able to detect multiple dipole anomalies in four different areas, even in the presence of zones with no data. These

results show the potential capability of using a model that has been trained with synthetic data in real applications. Furthermore, it is important to evaluate the performance of the model with real data in order to understand its prediction limits and its application domain.

CHAPTER

4 Perspectives

4.1 Introduction of the chapter

In this chapter, I present current work on improving the characterization of magnetic anomalies for a future application in real time survey. We plan to write a practical article for submission to a geophysical journal, but I do not think it is developed enough to be included in the main body of this thesis. Moreover, I propose some applications that I think will benefit the geophysics and AI community in both the academic and private sectors.

Altogether, I believe that the work presented is a significant milestone in the integration of deep neural networks for the inversion, characterization, and even identification of magnetic dipolar anomalies. Much work remains to be done to apply these models to other geophysical methods, but I am confident that the methodology developed here will be an important pillar to be considered in future research.

4.2 Work in progress

At some point in this research, we realized that we have a significant amount of real-world magnetic data available thanks to the collaboration with the company UXOMAP. In consequence, we decided to remain focused on the magnetic case study and to enhance the work done on more complex data, possibly real data, by emphasizing the application aspects, particularly in industry (UXO).

4.2.1 Methodology

I designed a new methodology to progressively evaluate the improvement of our current model by comparing our actual results with the impact of including real data in our database.

First, I cropped several areas of the real magnetic maps to verify the performance of our current model in detecting, counting and predicting the parameters of dipolar anomalies with a high level of complexity (real world case). This would provide an idea of the limitations of its capacities. Second, I propose to create two different databases for the training phase, one with a combination of synthetic and real data, and the second with only real data. This step would clarify whether mixing data types would be beneficial for prediction accuracy or whether, for a real-world application, we should only consider real data. Third, to take advantage of our actual model, it would be valuable to compare the results using transfer learning and learning from scratch techniques. We could potentially save time when testing new real data in the training base or even try to apply this model to other geophysical applications with different input data. Finally, I propose to compare the version of the model that learned only from synthetic data and the model that will result from applying all the strategies presented earlier. The predictions of the parameters of this last model will be compared to the information generated after the clearance of the areas considered in the database. This information often includes depth, radius, weight, and type of ammunition or explosive discovered.

4.2.2 Data labeling

In order to have more data for training, it is not only important to have multiple magnetic maps, but also to find a way to automate the task of labeling magnetic anomalies. In (Figure 4.1), we observe a magnetic map generated from a magnetic survey on an surface of 2315x1859 m^2 where the cleaning up work found 10479 metallic objects. Then, I did a pre-processing of the data and the final number was reduced to 3263. Drawing a bounding box for each dipole anomaly would be time-consuming, since we only have the information of the center location of the anomaly, so I propose two different strategies :

— Try to find a representative value for the height and width of each bounding box that would cover all possible anomaly sizes. The disadvantage is that sometimes the proposed box would not be compatible because the shapes of the magnetic anomalies are often affected by nearby metal objects.

— Since we know that the size of a dipole anomaly is directly proportional to its depth, and thus I can generate synthetic examples considering several sizes, I propose to create an adaptable bounding box. It is possible to create a ratio between the actual depth of the metal object and the theoretical size of the bounding box. This strategy would take into account the valuable depth information obtained after cleaning-up process of the area. Figure 4.2 is an example of the use of an adaptable bounding box.

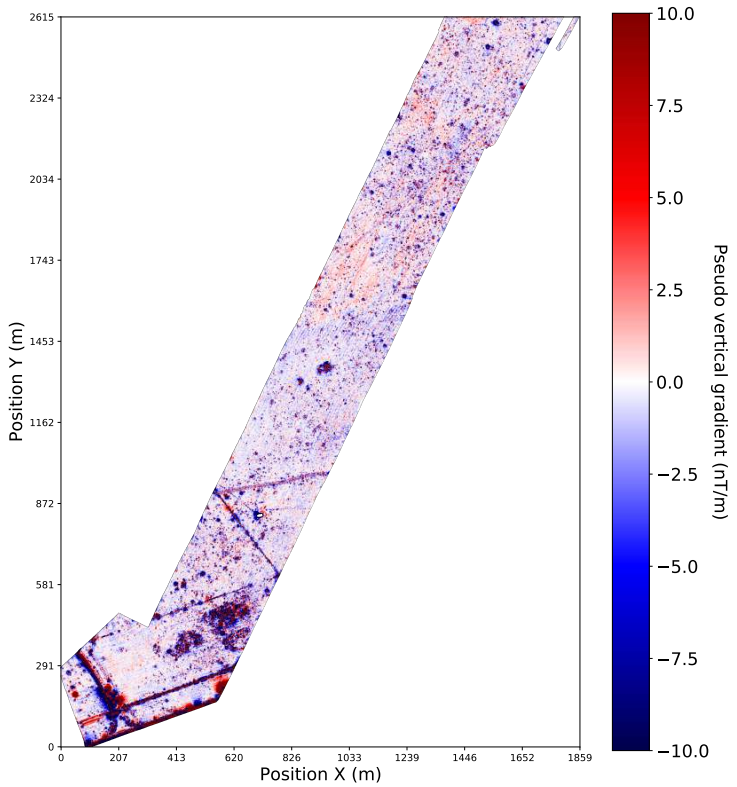


Figure 4.1 – Example of a map obtained after a magnetic survey.

4.2.3 Early results

Regarding our script, by running multiple tests with real data, I was able to improve it so the model could only make detections and predictions on areas with at least one data point (Figure A1-3). In addition, I could significantly reduce the prediction time. As mentioned earlier, during magnetic surveys, we often observe access difficulties that result in non-squared magnetic maps. As a consequence, it is necessary to apply a padding to the data to create a grid that could be used by our model YOLO.

Subsequently, after realizing multiple detection with different real magnetic map, I observed that our model was not capable of detect dipolar anomalies smaller than the ones I considered in my database (Figure 4.3). So, I modified the range of the parameters that I considered when creating the synthetic database. I took into account a more realistic range that could be observed in a magnetic survey, for example : depth values between 0.1 m and 5 m, and radius values between 1 cm and 24 cm. (Figure 4.3) shows that our new model can detect more dipolar anomalies (including smaller ones), and it can better identify larger anomalies.

The last of our early results is related to including the real magnetic maps in our training database. We tested the new version of our model on areas that were not present during the learning phase.

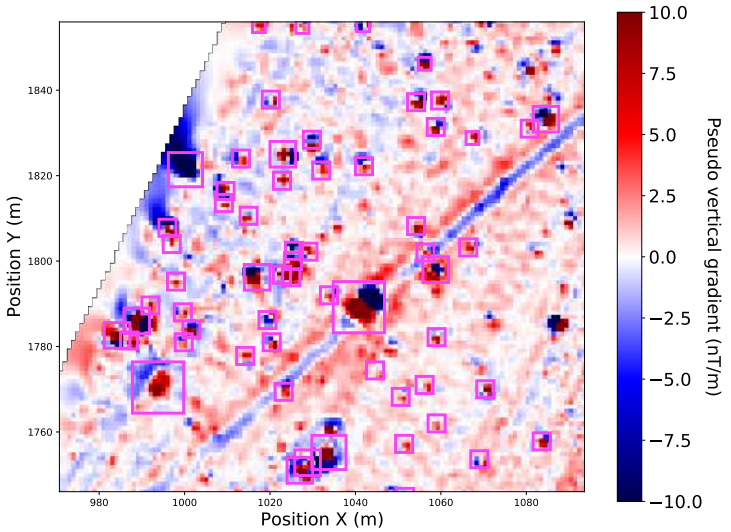


Figure 4.2 – Example of the use the adaptable bounding box strategy.

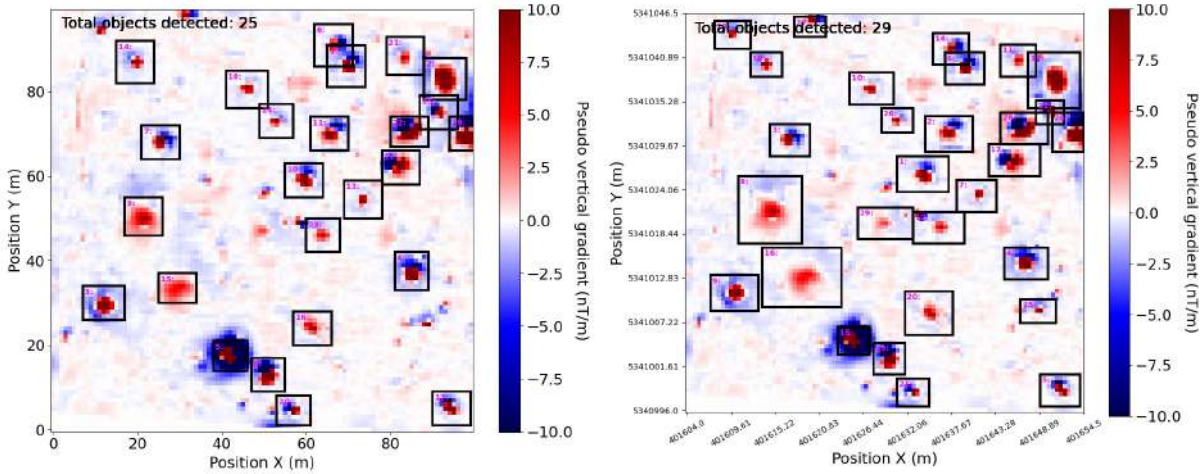


Figure 4.3 – Dipolar anomaly detections made by our model before (left-side) and after (right-side) changing the range of values considered in our database.

Some of the results are shown in (Figure 4.4). They show that our model, at this stage, manage to identify a great number of dipolar anomalies. Nevertheless, additional statistical and explanatory work needs to be done in order to confirm the performance observed. The next step in our current involves comparing the results using transfer learning and learning from scratch techniques.

Finally, after applying the improvements presented above, the total prediction time for the area presented in these results is approximately 400 seconds.

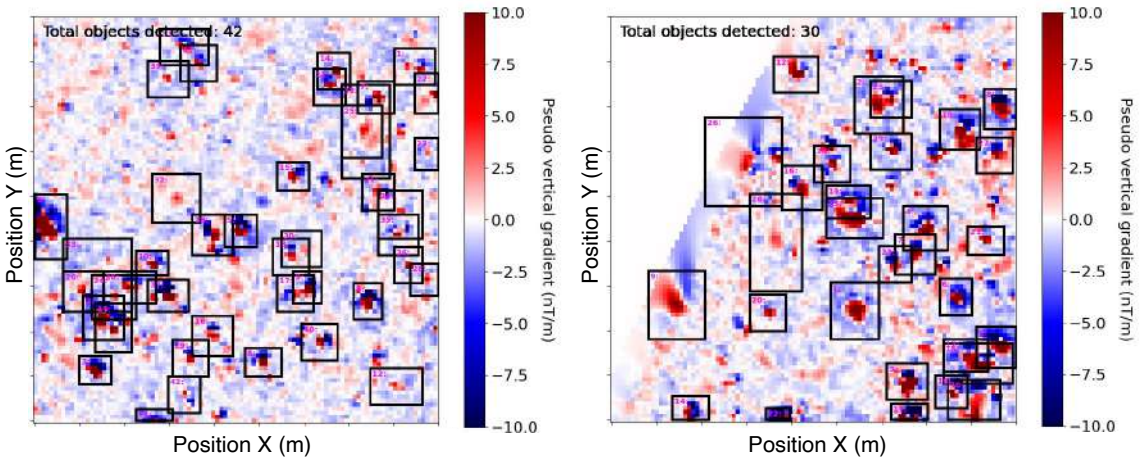


Figure 4.4 – Detections performed by the latest version of our model.

4.3 Future applications

The next step of our application would be to apply deep learning models to other geophysical methods. Specifically, for the electrical methods, the Python library pyGIMLi (Rücker et al., 2017) offers an environment to create apparent resistivity maps from geological bodies of different shape. Moreover, I believe that the U-Net architecture, that we used in this research, would be the most adequate network to perform a geophysical inversion on this type of data.

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are other deep neural networks that are becoming very popular nowadays. I think that one possible application is to create a model that could remove noise from the data. This would greatly improve detection performance. The second application involves a model capable of creating examples of real magnetic maps. Implementing this idea could help solve the problem of the limited amount of real data available in geophysics.

An invaluable experience as a doctoral student has been the opportunity to teach at the university to students of different academic levels. I think that the applications of neural networks are evolving in all disciplines. I therefore propose to start by teaching its basic principles in geophysics. Second, I consider that explainable neural networks should be taught to avoid the general idea of black box algorithms. Lastly, data labeling is a visual concept that could be an interesting pedagogic tool for a student to understand how we see a geophysical anomaly. In consequence, I suggest that a feed-back between the academic and the research could be beneficial for future applications of neural networks.

Conclusion

Many scientific disciplines are directing their research activities towards deep machine learning. This is also the case in geophysics, where it is used to facilitate the automated processing of geophysical data and the treatment of inversion problems. The application of deep learning algorithms have shown great promise, especially in the field of seismic interpretation (e.g. automatic fault recognition from seismic data sets). The purpose of this thesis is to further explore the potential of neural networks applications in geophysics to see if the remarkable results observed in the interpretation of seismic data would also be relevant in other geophysical methods, notably magnetic methods.

This thesis focused on the development of a new methodology based on deep machine learning, to characterize the dipolar magnetic anomalies present in a magnetic map. Although very little information on this application can be found in the literature, some new research has recently been carried out, which indicates a growing interest from the geophysical community. A recent work showed some advances in the characterization of magnetic anomalies. However, their application is still in the early stages of development, considering a future real-time implementation. We have observed that the predictions of their model are limited by one magnetic anomaly at a time. Moreover, the amount of synthetic data used for the learning phase was huge, which is a disadvantage considering the amount of time and computer memory to be used. Therefore, to further advance the magnetic application, the main objectives of this thesis are the identification of geological structures and the automatic recognition of anomalies.

We presented a new methodology based on machine learning, to detect magnetic anomalies present in a prospected area. Our model predicts the physical and magnetic characteristics of each detected magnetic dipolar-like responses. A synthetic dataset was generated by simulation to compensate for the limited number of labeled geophysical measurements. Our experiments have shown that the coupling of two models, YOLO and DenseNet, provides the best performance regarding our objectives. The YOLO model allows to detect and count each dipole while the DenseNet network estimates the parameters of each identified dipole with a high accuracy. Beyond the statistical performance, we

have optimized the model so that the discriminatory zone, visualized by the Grad-CAM tool, is consistent with the expected results to avoid the creation of statistical artifacts. It should be noted that this makes it easier to explain the results obtained. In particular, we optimized the number of model parameters to increase its stability. This is therefore a first step of functional verification of the black box of our model. The use of the t-SNE tool allowed us to evaluate the ability of the model to differentiate among all possible combinations of parameters present in the dataset. Our results show that these visualization tools are essential allies to study the robustness and stability of a deep learning model by going beyond the measurement of statistical accuracy. Finally, the application of our model to a real magnetic anomaly map showed that our model is able to detect multiple dipolar anomalies in four different surveyed areas, even in the presence of zones with no data. These results show the potential capability of using a model, that has been trained with synthetic data, in real applications.

At the end of this thesis, we show the current work that is being conducted in order to improve the performance of our models. We realized that we have a significant amount of real-world magnetic data available that should help us get closer to the idea of a real-time implementation. In consequence, we first suggest a methodology to progressively evaluate the improvement of our current model by comparing our actual results with the impact of including real data in our database. Then, I present two strategies to automate the task of labeling magnetic anomalies in real magnetic maps. This step is really important because, on the one hand, the performance of the YOLO model depends on a correctly labeled database, and on the other hand, more than 10,000 anomalies can be observed on a magnetic map. Lastly, I present some improvements I have made in the script to speed up the detection and prediction time of our model, and the actual results when adding more real data to the database.

To conclude, based on the results of this thesis, it appears that we are taking another step forward in the application of deep neural networks in geophysics. The next step should involve the application of deep learning models to other geophysical methods, specifically the electrical methods. Moreover, this work should be extended to hydrological applications, which would provide a valuable information for the interpretation of geological structures. Finally, parallel to these applications, explanatory work must be carried out to explain the behavior of a model, to be transparent about the results and accuracy of the models, and to justify their predictions to the geophysical/AI community.

Appendix 1

In this chapter, I will show some figures that I have generated throughout my research since the first application. I consider them useful to support the results I explain in this thesis.

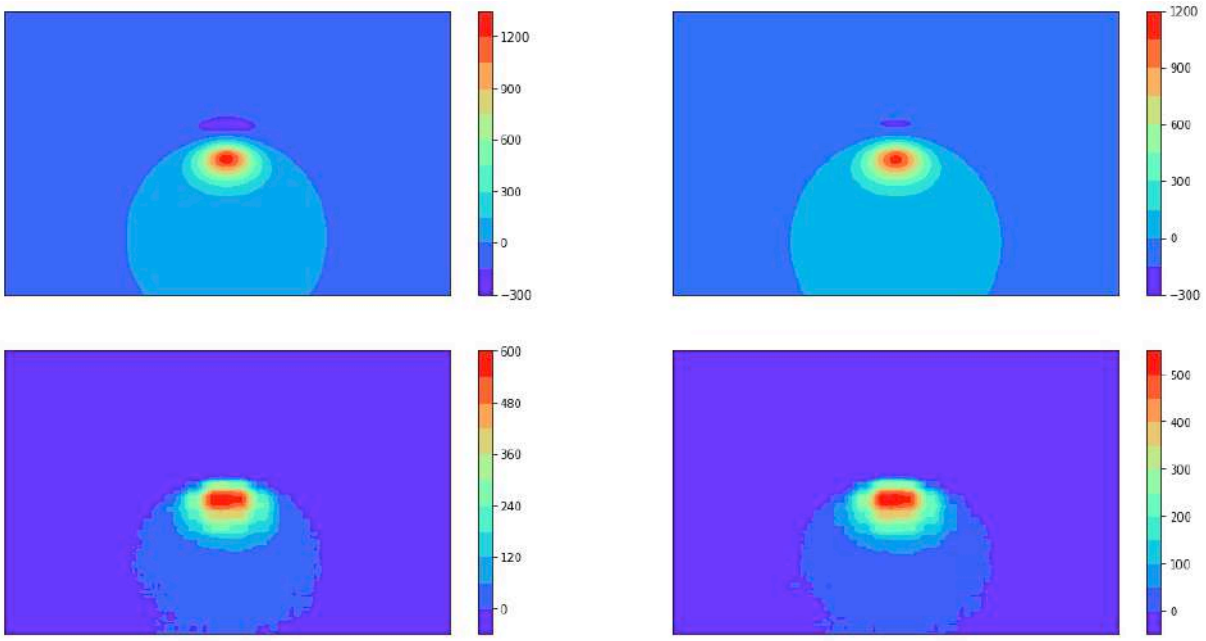


Figure A1-1 – Image reconstruction (bottom row) of an autoencoder model when a simple dipolar anomaly is used as input data (top row).

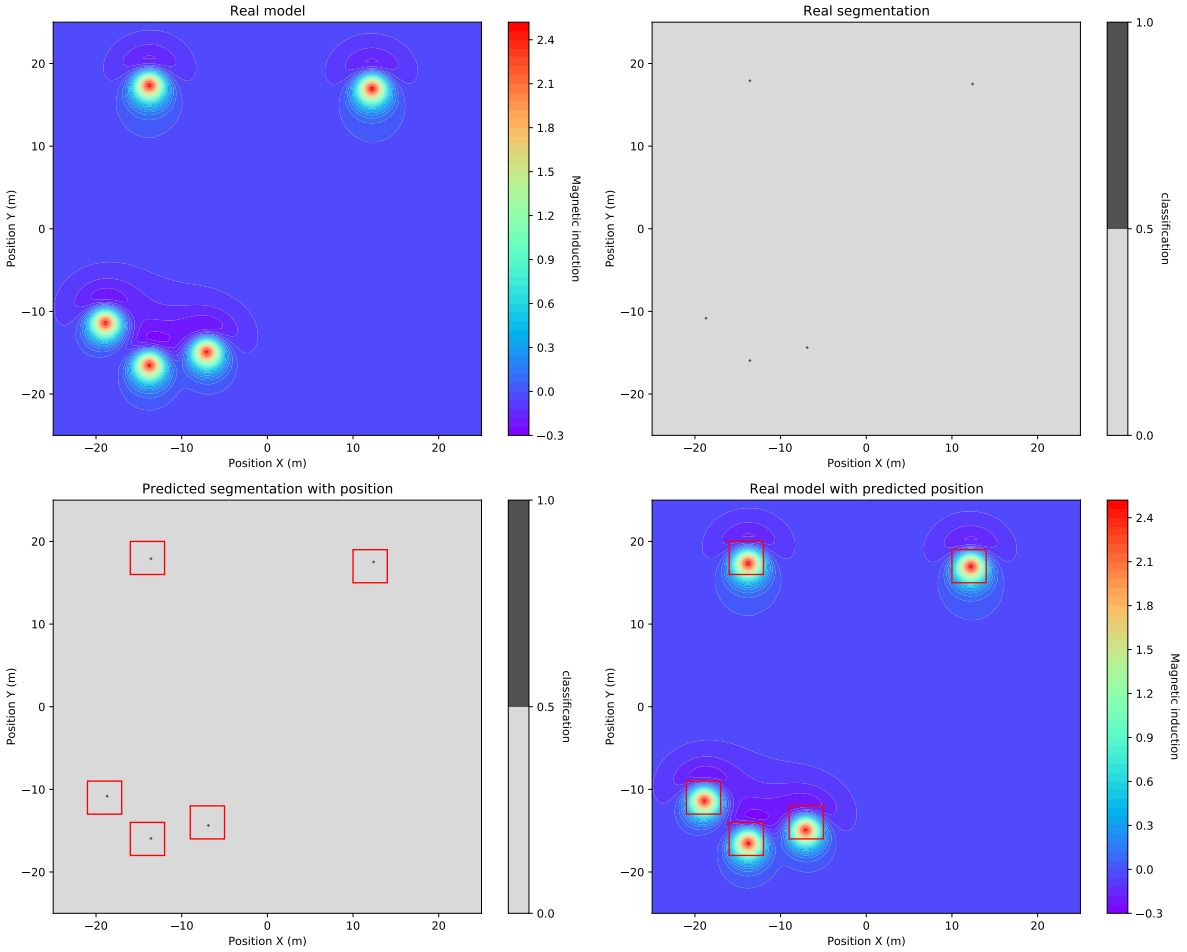


Figure A1-2 – Segmentation predictions of the U-Net model. The images displayed are the followings : a magnetic map that is use as input data (upper-left), the real segmentation map generated from the magnetic map (upper-right), the segmentation map predicted by the model (lower-left), and the overlay of the results with the values of height and width fixed previously to highlight the dipolar anomalies (lower-right).

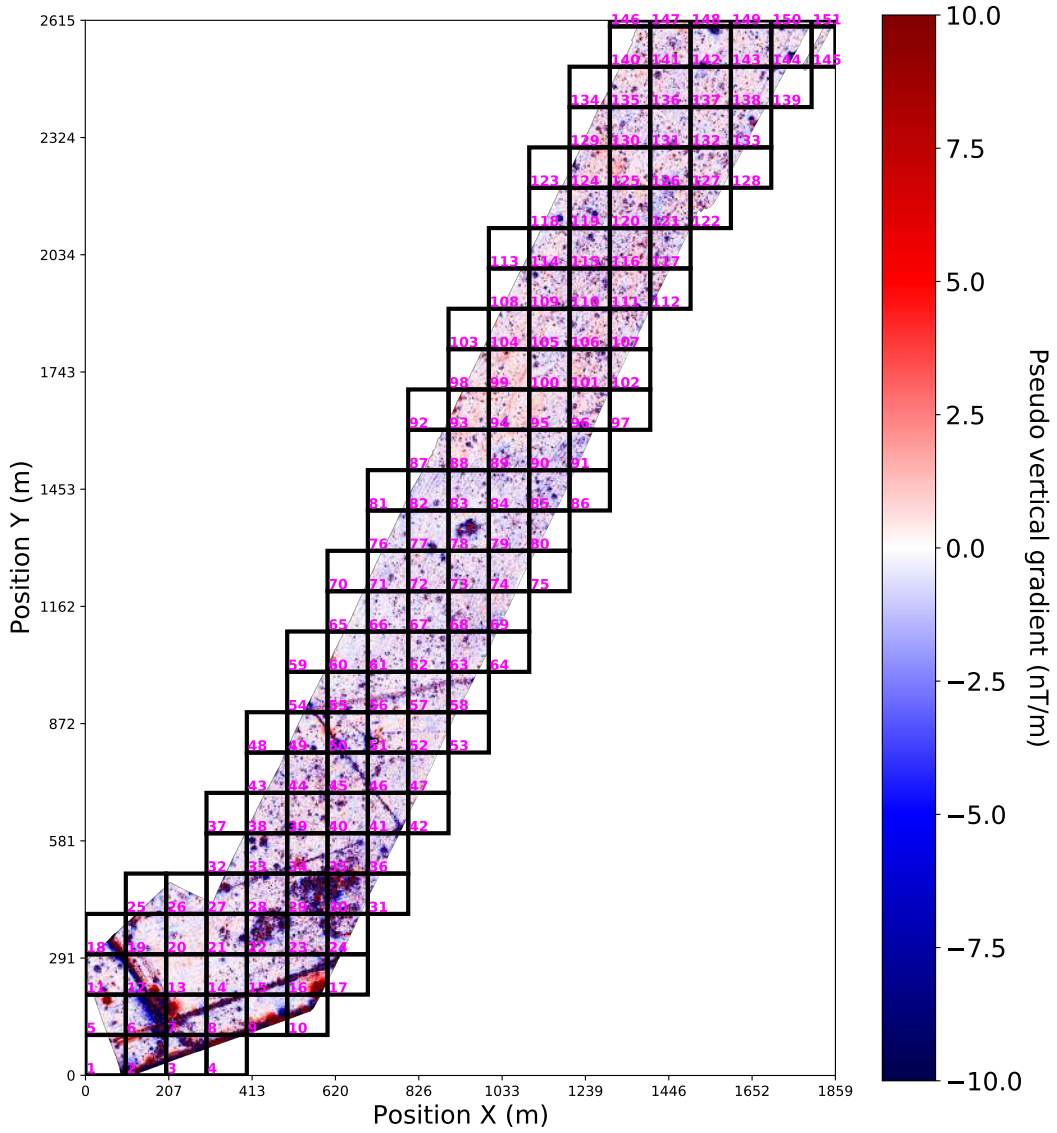


Figure A1-3 – Example of labeled areas used during prediction to avoid areas without data.

List of Acronyms

General

GIS Geographic Information System

R&D Research and Development

Deep Learning

CNN Convolutional Neural Network

GANs Feedforward Neural Network

GANs Generative adversarial networks

IoU Interception over Union

ML Machine Learning

MSE Mean Square Error

NN Neural Network

SVM Support Vector Machine

YOLO You Only Look Once

Geophysics

EMI Electromagnetic induction

SQUID Superconducting Quantum Interference Device

TDEM Time Domain Electromagnetic

UXO Unexploded Ordnance

Bibliography

Ali, Muhammad, Ren Jiang, Huolin Ma, Heping Pan, Khizar Abbas, Umar Ashraf, and Jar Ullah (Aug. 2021). « Machine learning - A novel approach of well logs similarity based on synchronization measures to predict shear sonic logs ». In: *Journal of Petroleum Science and Engineering* 203, p. 108602. ISSN: 09204105. DOI: 10.1016/j.petrol.2021.108602.

Almond, Richard (2006). *Potent*. Geophysical Software Solutions (GSS). 12 pp.

An, Yu, Jiulin Guo, Qing Ye, Conrad Childs, John Walsh, and Ruihai Dong (Aug. 2021). « Deep convolutional neural network for automatic fault recognition from 3D seismic datasets ». In: *Computers & Geosciences* 153, p. 104776. ISSN: 00983004. DOI: 10.1016/j.cageo.2021.104776.

Anemangely, Mohammad, Ahmad Ramezanzadeh, Hamed Amiri, and Seyed-Ahmad Hoseinpour (Mar. 2019). « Machine learning technique for the prediction of shear wave velocity using petrophysical logs ». In: *Journal of Petroleum Science and Engineering* 174, pp. 306–327. ISSN: 09204105. DOI: 10.1016/j.petrol.2018.11.032.

Araya-Polo, Mauricio, Joseph Jennings, Amir Adler, and Taylor Dahlke (Jan. 2018). « Deep-learning tomography ». In: *The Leading Edge* 37.1, pp. 58–66. ISSN: 1070-485X, 1938-3789. DOI: 10.1190/tle37010058.1.

Bevan, Bruce W (2017). *Analysis of dipolar magnetic anomalies*. Tech. rep. DOI:10.13140/RG.2.2.35995.05929/1.

Bishop, Christopher M. (Jan. 1996). *Neural Networks for Pattern Recognition*. English. 1st edition. Oxford : New York: Oxford University Press, USA.

Blackett, P. (1952). « A negative experiment relating to Magnetism and the Earth's Rotation ». In: *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*. DOI: 10.1098/rsta.1952.0024.

Blackett, P. M. S. (May 1947). « The Magnetic Field of Massive Rotating Bodies* ». In: *Nature* 159.4046. Number: 4046 Publisher: Nature Publishing Group, pp. 658–666. ISSN: 1476-4687. DOI: 10.1038/159658a0.

Breiner, Sheldon (1973). *Applications Manual for Portable Magnetometers*. en. Google-Books-ID: xWZUNAEACAAJ. GeoMetrics.

Burkov, Andriy (2019). *The hundred-page machine learning book*. English. OCLC: 1083639942.

Cockett, Rowan, Seogi Kang, Lindsey J. Heagy, Adam Pidlisecky, and Douglas W. Oldenburg (Dec. 2015). « SimPEG: An open source framework for simulation and gradient based parameter estimation in geophysical applications ». In: *Computers & Geosciences* 85, pp. 142–154. ISSN: 00983004. DOI: 10.1016/j.cageo.2015.09.015.

Das, Vishal, Ahinoam Pollack, Uri Wollner, and Tapan Mukerji (Aug. 2018). « Convolutional neural network for seismic impedance inversion ». In: *SEG Technical Program Expanded Abstracts 2018*. Anaheim, California: Society of Exploration Geophysicists, pp. 2071–2075. DOI: 10.1190/segam2018-2994378.1.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (June 2009). « ImageNet: A large-scale hierarchical image database ». In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN: 1063-6919, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Florsch, Nicolas, Frédéric Muhlach, and Michel Kammenthaler (Jan. 2019). *Géophysique appliquée pour tous 2*. Français. London: ISTE Editions.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). « Generative Adversarial Nets ». In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html (visited on 06/15/2022).

Guo, Jiateng, Yunqiang Li, Mark Walter Jessell, Jeremie Giraud, Chaoling Li, Lixin Wu, Fengdan Li, and Shanjun Liu (Apr. 2021). « 3D geological structure inversion from Noddy-generated magnetic data using deep learning methods ». In: *Computers & Geosciences* 149, p. 104701. ISSN: 00983004. DOI: 10.1016/j.cageo.2021.104701.

Heagy, Lindsey J., Douglas W. Oldenburg, Fernando Pérez, and Laurens Beran (Sept. 2020). « Machine learning for the classification of unexploded ordnance (UXO) from electromagnetic data ». In: *SEG Technical Program Expanded Abstracts 2020*. Virtual: Society of Exploration Geophysicists, pp. 3482–3486. DOI: 10.1190/segam2020-3428369.1.

Hinze, William J., R. Von Frese, and Afif H. Saad (2013). *Gravity and magnetic exploration: principles, practices, and applications*. New York: Cambridge University Press.

Huang, Gao, Zhuang Liu, Laurens Van der Maaten, and Kilian Q. Weinberger (2018). « Densely Connected Convolutional Networks ». In: *arXiv:1608.06993*.

Ishitsuka, Kazuya, Shinichiro Iso, Kyosuke Onishi, and Toshifumi Matsuoka (Nov. 2018). « Object Detection in Ground-Penetrating Radar Images Using a Deep Convolutional Neural Network

and Image Set Preparation by Migration ». In: *International Journal of Geophysics* 2018, pp. 1–8. ISSN: 1687-885X, 1687-8868. DOI: 10.1155/2018/9365184.

Ives, Libby (2016). « Magnetic mineralogy and fabrics of small-scale glacial flutes, Múlajökull and Breiðamerkurjökull, Iceland ». In: Publisher: Unpublished. DOI: 10.13140/RG.2.1.3782.4247.

Kose, Utku, Omer Deperlioglu, and D. Jude Hemanth (July 2021). *Deep Learning for Biomedical Applications*. en. Google-Books-ID: XDQxEAAAQBAJ. CRC Press.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (May 2017). « ImageNet classification with deep convolutional neural networks ». en. In: *Communications of the ACM* 60.6, pp. 84–90. DOI: 10.1145/3065386. URL: https://dl.acm.org/doi/10.1145/3065386 (visited on 10/22/2021).

Laloy, Eric, Niklas Linde, Cyprien Ruffino, Romain Hérault, Gilles Gasso, and Diederik Jacques (Dec. 2019). « Gradient-based deterministic inversion of geophysical data with generative adversarial networks: Is it feasible? » In: *Computers & Geosciences* 133, p. 104333. ISSN: 00983004. DOI: 10.1016/j.cageo.2019.104333.

Liu, Bin, Qian Guo, Shucai Li, Benchao Liu, Yuxiao Ren, Yonghao Pang, Xu Guo, Lanbo Liu, and Peng Jiang (Aug. 2020). « Deep Learning Inversion of Electrical Resistivity Data ». In: *IEEE Transactions on Geoscience and Remote Sensing* 58.8, pp. 5715–5728. ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2020.2969040.

Lowrie, William and Andreas Fichtner (2020). *Fundamentals of geophysics*. eng. Third edition. Cambridge New York, NY Port Melbourne, VIC New Delhi Singapore: Cambridge University Press. DOI: 10.1017/9781108685917.

Martín Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Mattis, Daniel Charles (1965). *The Theory of Magnetism: An Introduction to the Study of Cooperative Phenomena*. Google-Books-ID: h9pEAAAAIAAJ. Harper & Row.

Meier, U., A. Curtis, and J. Trampert (Aug. 2007). « Fully nonlinear inversion of fundamental mode surface waves for a global crustal model ». In: *Geophysical Research Letters* 34.16. ISSN: 00948276. DOI: 10.1029/2007GL030989.

Merrill, Ronald T., M. W. McElhinny, and Phillip L. McFadden (1996). *The magnetic field of the earth: paleomagnetism, the core, and the deep mantle*. International geophysics series v. 63. San Diego, Calif: Academic Press.

Milsom, John and Asger Eriksen (2011). *Field geophysics*. 4th ed. The geological field guide series. OCLC: ocn657270827. Hoboken, NJ: Wiley.

Mosser, Lukas, Olivier Dubrule, and Martin J. Blunt (2018). « Stochastic seismic waveform inversion using generative adversarial networks as a geological prior ». In: *arXiv:1806.03720*.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (June 2016). « You Only Look Once: Unified, Real-Time Object Detection ». In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

Reynolds, John M. (2011). *An introduction to applied and environmental geophysics*. 2nd ed. Chichester, West Sussex ; Malden, Mass: Wiley-Blackwell.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (May 2015). « U-Net: Convolutional Networks for Biomedical Image Segmentation ». In: *arXiv:1505.04597 [cs]*. arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597 (visited on 11/02/2021).

Rücker, C., T. Günther, and F. M. Wagner (2017). « pyGIMLi: An open-source library for modelling and inversion in geophysics ». In: *Computers and Geosciences* 109, pp. 106–123. DOI: 10.1016/j.cageo.2017.07.011. URL: https://www.sciencedirect.com/science/article/pii/S0098300417300584.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (Oct. 1986). « Learning representations by back-propagating errors ». In: *Nature* 323.6088, pp. 533–536. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/323533a0.

Sarkar, Dipanjan, Raghav Bali, and Tamoghna Ghosh (Aug. 2018). *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. en. Google-Books-ID: aPFsDwAAQBAJ. Packt Publishing Ltd.

Schmidt, Armin, Michel Dabas, and Apostolos Sarris (Sept. 2020). « Dreaming of Perfect Data: Characterizing Noise in Archaeo-Geophysical Measurements ». In: *Geosciences* 10.382. Publisher: MDPI. DOI: 10.3390/geosciences10100382. URL: https://hal.archives-ouvertes.fr/hal-02949603 (visited on 06/13/2022).

Scollar, I., ed. (1990). *Archaeological prospecting and remote sensing*. Topics in remote sensing. Cambridge [England] ; New York, 674pp.: Cambridge University Press.

Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra (Feb. 2020). « Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization ». In: *International Journal of Computer Vision* 128.2, pp. 336–359. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-019-01228-7.

Simonyan, Karen and Andrew Zisserman (2015). « Very Deep Convolutional Networks for Large-Scale Image Recognition ». In: *arXiv:1409.1556*.

Tchernychev, Mikhail (2009). *MAGPICK - magnetic map & profile processing. User Guide*. Geometrics Inc. San Jose, USA. 155 pp.

Travassos, Xisto L., Sérgio L. Avila, and Nathan Ida (2021). « Artificial Neural Networks and Machine Learning techniques applied to Ground Penetrating Radar: A review ». In: *Applied Computing and Informatics* 17, pp. 296–308. ISSN: 2634-1964, 2210-8327. DOI: 10.1016/j.aci.2018.10.001.

Van der Baan, Mirko and Christian Jutten (July 2000). « Neural networks in geophysical applications ». In: *Geophysics* 65.4, pp. 1032–1047. ISSN: 0016-8033, 1942-2156. DOI: 10.1190/1.1444797.

Van der Maaten, Laurens and Hinton Geoffrey (Nov. 2008). « Visualizing High-Dimensional Data Using t-SNE ». In: *Journal of Machine Learning Research* 34, pp. 2579–2605.

Wu, Xinming, Luming Liang, Yunzhi Shi, and Sergey Fomel (May 2019). « FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation ». In: *Geophysics* 84.3, pp. IM35–IM45. ISSN: 0016-8033, 1942-2156. DOI: 10.1190/geo2018-0646.1.

Zhao, Tao (Aug. 2019). « 3D convolutional neural networks for efficient fault detection and orientation estimation ». In: *SEG Technical Program Expanded Abstracts 2019*. San Antonio, Texas: Society of Exploration Geophysicists, pp. 2418–2422. DOI: 10.1190/segam2019-3216307.1.

Abstract

This thesis presents the characterization of magnetic anomalies using convolutional neural networks, and the application of visualization tools to understand and validate their predictions. The developed approach allows the localization of magnetic dipoles, including counting the number of dipoles, their geographical position, and the prediction of their parameters (magnetic moment, depth, and declination). Our results suggest that the combination of two deep learning models, "YOLO" and "DenseNet", performs best in achieving our classification and regression goals. Additionally, we applied visualization tools to understand our model's predictions and its working principle. We found that the Grad-CAM tool improved prediction performance by identifying several layers that had no influence on the prediction and the t-SNE tool confirmed the good ability of our model to differentiate among different parameter combinations. Then, we tested our model with real data to establish its limitations and application domain. Results demonstrate that our model detects dipolar anomalies in a real magnetic map even after learning from a synthetic database with a lower complexity, which indicates a significant generalization capability. We also noticed that it is not able to identify dipole anomalies of shapes and sizes different from those considered for the creation of the synthetic database. Our current work consists in creating new databases by combining synthetic and real data to compare their potential influence in improving predictions. Finally, the perspectives of this work consist in validating the operational relevance and adaptability of our model under realistic conditions and in testing other applications with alternative geophysical methods.

Key words

Geophysics, Deep learning, Convolutional neural networks, Magnetic methods, Explainability, Unexploded ordnance.