Hybrid Approaches for Spatial Data Interpolation

Marcos Domingues Barata

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins

Examination Committee

Chairperson: Prof. Doutor Pedro Tiago Gonçalves Monteiro
Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins
Members of the Committee: Prof. Doutor Nuno Miguel Soares Datia

January 2022

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao Professor Bruno Emanuel da Graça Martins e ao Professor Jacinto Paulo Simões Estima pela orientação durante este processo, para o qual contribuíram de forma muito significativa para o trabalho que desenvolvemos juntos, com seu conhecimento, motivação e compreensão.

Em segundo lugar, gostaria de agradecer à minha mãe e ao meu pai, por sempre me terem dado o máximo de amor, terem lutado por mim e pelos sacrifícios feitos para que nada me faltasse e eu pudesse ser o meu melhor. Falando em sacrifícios, agradeço muito aos meus avós, pois sempre me amaram e trabalharam muito, para dar aos meus pais condições para evoluirem e consequentemente para a minha própria evolução. A minha admiração também vai para meus padrinhos, Madrinha e Padrinho, pois sempre me trataram como filho. À Madrinha Linda e ao Padrinho Jorge também tenho uma palavra de apreço e respeito pelo amor que me deram. Aos meus tios, João e Paula, um obrigado pelo o amor e apoio. Não poderia escrever isto sem todos os meus primos pela boa relação que temos e o apoio que me dão. Se eu agradecesse a todos individualmente e detalhasse porque gosto deles e sua importância na minha vida, perderia muito tempo precioso durante o qual poderia estar com eles. Posto isto, queria agradecer a toda a minha família do fundo do meu coração e que eu não estaria aqui se não fosse por cada um de vós e cada momento que vivemos juntos.

Em terceiro lugar, queria agradecer à minha namorada, Pipa, por toda a ajuda e motivação fundamentais para a concretização deste trabalho, mas agredecer acima de tudo a cumplicidade e o amor que nos une.

Seguidamente, quero agradecer e homenagiar os meus amigos: Alex, Bruno, Gongas, Guerra, Jaime, Jomi, Pedro, Rodolfo e Xico pela amizade e pelo apoio que sempre me deram.

Por último agradecer a todos os professores e mestres que tive ao longo da vida, por me terem dado bases e métodos para o meu futuro. E aos a todos os colegas com quem me cruzei por me terem ensinado o comportamento humano e a viver em sociedade.

Marcos Domingues Barata

Especialmente para a minha russa.

Resumo

A interpolação de dados espaciais é um problema que atravessa uma panóplia de ciências naturais e sociais. Como em qualquer interpolação, o objetivo é aproximar a estimativa de novos valores aos valores reais correspondentes. As técnicas tradicionais de interpolação espacial são baseadas em abordagens matemáticas e, em algumas técnicas, combinadas com conceitos geográficos. Uma das consequências do crescimento do poder de computação foi a evolução de técnicas de aprendizagem, especialmente ao nível de redes neuronais. Um dos passos mais relevantes no que toca a essas técnicas foi o desenvolvimento de redes neuronais capazes de processar imagens e capturar caraterísticas que as compõem. Quando aplicadas ao problema de interpolação espacial, as redes neuronais, como o perceptrão multicamada, podem igualar o desempenho das técnicas tradicionais, mas as redes neuronais mais recentes, baseadas no processamento de imagens e utilizando camadas convolucionais podem superar as técnicas tradicionais. As técnicas tradicionais podem ser aplicadas a todos os contextos, mas as redes neuronais enfrentam o problema de generalização, onde a rede treinada dentro de um contexto particular pode ter um desempenho inferior quando colocada num contexto diferente. Este artigo apresenta uma abordagem híbrida. Com base numa rede adversarial generativa, combinamos a rede neuronal com uma abordagem de interpolação espacial tradicional, a fim de superar as abordagens tradicionais e alcançar um grau de generalização maior. Este estudo mostra que a nossa abordagem é uma opção viável para resolver o problema da interpolação espacial uma vez que os resultados da interpolação do modelo digital de elevação de Portugal Continental e das ilhas das regiões autónomas portuguesas evidenciam um sólido desempenho e capacidade de generalização.

Palavras-chave: Interpolação de dados espacias, Ponderação de distância inversa, Camadas convolucionais, Redes adversariais generativas, Modelos de elevação digital.

Abstract

Spatial data interpolation is a problem that crosses a panoply of natural and social sciences. The goal is to estimate a surface based on a sample of locations. Traditional spatial interpolation techniques are based on mathematical approaches and, in some cases, combined with geographic concepts. The growth of computational power consequently resulted in the evolution of machine learning approaches, especially neural networks. One of the most relevant steps in terms of these techniques was the development of neural networks capable of processing images and capturing features that characterize them. When applied to the spatial interpolation problem, neural networks, like the multi-layered perceptron, can equalize the performance of more traditional techniques, but the neural networks based on image processing and relying on convolutional layers can overcome the traditional techniques. Traditional techniques can be applied in every context, while neural networks face the problem of generalization where a network trained within a particular context can underperform when put in a different context. This paper presents a hybrid approach. Based on a generative adversarial network, where we combine a neural network with a traditional spatial interpolation approach to overcome the traditional approaches and to achieve a higher generalization degree. This study shows that our approach is a viable option to solve the spatial interpolation problem since the results of interpolating the digital elevation model of mainland Portugal and the islands of the Portuguese autonomous regions evidence a solid performance and generalization ability.

Keywords: Spatial data interpolation, Inverse distance weighting, Convolutional layers, Generative adversarial networks, Digital elevation models.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Spatial Data Interpolation is a problem that covers many subjects and fields going from the natural to the social phenomena. This problem consists in the estimation of new spatial information from the already known information, and can be formally defined by the following d-variate function that satisfies the condition for each N known points:

$$F\left(r_j\right) = z_j\ ,\ j = 1, \ldots, N \tag{1.1}$$

where z_j, is the observed value at the r_j discrete point.

As can be seen, an infinite number of functions can fulfil this constraint, so additional conditions have to be added, differentiating the interpolation techniques. Traditional techniques restrain the solution based on a local neighbourhood approach (e.g., Inverse Distance Weighting - IDW), geostatistical approach (e.g., Simple Kriging) or a varionational approach (e.g., Thin Plate Spline). Although these methods are easy to understand/implement and very adaptable to all contexts, they are limited when it comes to capture complex geospatial features and contexts, affecting negatively their accuracy. Oppose to traditional techniques, modern approaches based on machine learning condition the solution by going through a set of input-output pairs in order to capture the geospatial contexts of the available data. This solutions started by having the potential to match the traditional approaches performance, but nowadays have proved to overcome them by relying in advanced image processing techniques like convolutional neural networks. Although neural networks can overcome traditional techniques in terms of performance, they face the disadvantage of generalization since a network trained within a specific context can underperform when put in a different context. This limitation contrasts with the traditional ones since these techniques do not rely on a learning phase over input data to capture the spatial contexts. Summarizing, we have a modern technique that can overcome the traditional techniques, but since it is data dependent it faces a high probability of failure when applied to different contexts than the ones it was trained in.

1.2 Thesis Proposal

In order to exploit the strengths and weaknesses of classical and modern approaches we propose an hybrid approach that combines them, aiming for a method that overcomes all the negative aspects of the approaches we are combining. The traditional technique we are using in the hybrid approach is the IDW, since is one of the most mentioned traditional techniques due to its trivial equation and implementation. On the side of the modern approach we decided to use a generative adversarial network (GAN) designed by Zhu et al. (2019) that proved to overcome traditional techniques in interpolating digital elevation model (DEM) images from four Chinese regions. Since the modern approaches can overcome the older ones, they face a new problem of generalization. Having that, we propose to train the network with a two-channel image that combines the sampled image with an interpolated image, which is obtained by applying the IDW to the same sampled configuration of a real image. This train forces the network to capture the geospatial features that the IDW produces, conditioning even more the generation from a sampled image to a sampled image combined with a limited estimation of the desired output. We believe that this conditioning will guide the model to better results and generalization degree, since the input already gives an impression of what the output should look like.

1.3 Contributions

The main contributions of this thesis are as follows:

- The proposal of a novel spatial interpolation method that combines a traditional and a modern approach. This method bases on the GANs proposed by Zhu et al. (2019) where the model overcame the traditional models. Our approach aims to tackle the IDW and the model presented by Zhu et al. (2019).

- Study the generalization ability of our model and compare this ability with the ability of the simpler model inspired by Zhu et al. (2019) to generalize when transposed to a different context.

- The study and experiment of a traditional technique (IDW) and the model proposed by Zhu et al. (2019), comparing the techniques with the proposed hybrid method. This experiment is done by applying the techniques to a digital elevation model of Portugal and the islands of the Portuguese autonomous regions.

1.4 Structure of the Document

The rest of this document is organized as follows. Chapter 2 presents traditional spatial interpolation methods, an introduction to machine learning and how it can be adapted to solve the spatial interpolation problem, as well as important related work in the areas of spatial interpolation. Then, Chapter 3 describes the different steps involved in the considered hybrid approach, detailing also the ancillary datasets. Chapter 4 presents the results of evaluation experiments. Finally, Chapter 5 concludes this document by summarizing the main findings of this work, and highlighting possible directions for future research.

2 Concepts and Related Work

This chapter presents fundamental concepts and related work regarding the spatial interpolation problem, machine learning models and how machine learning can be adapted to solve the spatial interpolation problem.

2.1 Fundamental Concepts

As defined in Chapter 1, the spatial interpolation problem takes N known points and estimates all the unknown points of the given area/raster. This estimation can have an infinite number of solutions, so conditions based on additional information to the problem must be added in order approximate the equation 1.1 to the real values. In the next sections, we present some methods that are used to solve this problem, exploring how they condition the solution and their advantages and disadvantages.

2.1.1 Classical Spatial Interpolation Methods

This subsection overviews two classical methods that are used to solve the spatial interpolation problem, kriging and IDW. These methods are the most used and mentioned when it comes to spatial interpolation, but there are many other classical methods and variations of these methods that were not explored in this work.

2.1.1.1 Kriging

Kriging is a geostatistical procedure based on a concept of random functions (Gaussian process), where the area or volume is assumed to be the realisation of a random function with a certain spatial covariance. There are various kriging techniques, which are mainly determined by the heterogeneity of the property to be modelled and the availability of data, since they impact the assumption of a local stationary. Consider a random variable, Z(x), defined in a spatial domain:

$$\{Z(x) : x \subset \mathbb{R}^k\} \tag{2.1}$$

where x is the sample location and Z(x) is the property value at x within the defined domain D, which is a bounded subset of the k-dimensional real space, R^k. In the next paragraph we describe simple kriging, a basic form of kriging.

Simple Kriging relies on a linear equation for spatial prediction:

$$Z^*(x) = m + \sum_{i=1}^{n} \lambda_i [Z(x_i) - m] \tag{2.2}$$

Where m is the mean of the random function Z(x), n is the number of known samples and λ_i is the weight associated with that sample. The estimation error between the kriging estimator and the unknown point is $\varepsilon = Z(X) - Z(x)^*$. The Kriging system is obtained by minimizing the squared-errors, ε^2, and can be expressed by:

$$C_{zz}\Lambda_{sk} = c_z \tag{2.3}$$

C_{zz} is the matrix of the spatial covariance of the data used for prediction, Λ_{sk} is the vector of the weights and c_z is the vector of spatial covariance between Z(x) and $Z(x_j)$. This technique assumes that the mean(m) is a known constant, so to solve the equation 2.2 we need to resolve the weights. To do that we need to solve the equation 2.3 for the kriging weights after we calculate C_{zz} and c_z. After solving the kriging weights we can apply them to the equation 2.2 to interpolate the unknown point. Finally we calculate the estimator and estimation error variance (Myers, 1994; Mitas and Mitasova, 2005; Ma, 2019).

2.1.1.2 Inverse Distance Weighting - IDW

One of the simplest methods to interpolate data is IDW. IDW is the mathematical representation of Tobler's first law of Geography, "Everything is related to everything else, but near things are more related than distant things." Tobler (1970), given by:

$$V_j = \frac{\sum_{i=1}^{n} v_i \frac{1}{(d_i)^p}}{\sum_{i=1}^{n} \frac{1}{(d_i)^p}} \tag{2.4}$$

Where the estimation of the value V_j in the position j, takes into account all the n known values (v_i) and the distance (d_i) between the known point i and unknown point j, giving more weight to closest values. The value p represents the power parameter and it's responsible to control the impact of known points on the interpolated values based on their distance. This value is a positive real number and the default value is 2.

Recent studies on IDW (Greenberg et al. (2011); Stachelek and Madden (2015)) proved that we can

add additional information to the IDW model, by exchanging the euclidean distance for a non-euclidean distance obtained by traversing a search tree for source i to sink j. This informed search allows us to add useful knowledge to the model by establishing barriers and highlighting known relations between points. The use of a non-euclidean distance has proved better than the classic euclidean distance, but only when the distance is defined properly and in problems where the presence of barriers is necessary.

2.1.2 Machine Leaning with Deep Neural Networks

Artificial neural networks are computational approaches that aim to mathematically represent information processing as we observe in biological systems. Inspired by the animal brain these systems are a collection of connected units that process and propagate the input signal through the network to produce an output. Like an animal these systems need to learn to do their task and they do it by updating their internal state when they evaluate the output. The simplest artificial neural networks follow a supervised learning paradigm, since they rely on pairs of inputs and their desired outputs to update their internal state (by minimizing an error function), but they can also follow unsupervised or reinforcement learning paradigms. The training is the most important phase in these systems because to approach these systems for real-world operation it is required time and data to train the model. In the next sections we travel from the simplest example of an artificial neural network to more complex models and how they can be adapted to solve the spatial interpolation problem.

2.1.2.1 Perceptron

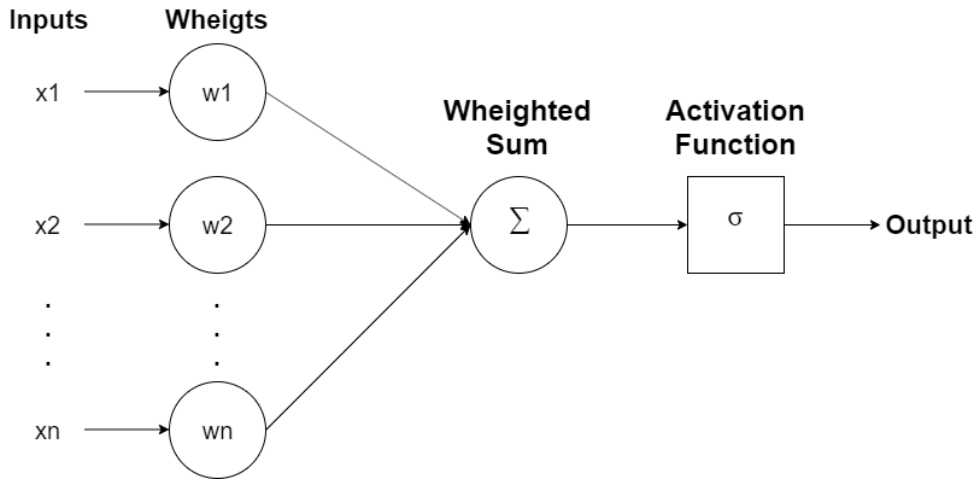

Figure 2.1: Representation of the Perceptron

Inspired in the neuron, the perceptron is a binary classifier of supervised learning. As can be observed in Figure 2.1, this model transforms the input vector using a fixed nonlinear transformation to

create a feature vector $\phi(x)$, which usually includes a bias component. Each feature is then multiplied by its given weight $w^T\phi(x)$ and the results are added (weighted sum) and then applied to the activation function f to map the input to the output binary values. The bias component have an important relation with the activation function, since it shifts the activation function, fitting the input to the desired output $y(x)$.

$$y(x) = f\left(w^T\phi(x)\right) \tag{2.5}$$

The algorithm learns by iterating through the input and: if it is correctly classified the algorithm does not update the weights; else the perceptron updates the weights by applying a stochastic gradient descent, as described by (2.6) and (2.7):

$$w_j^{new} = w_j^{old} + \Delta w_j \tag{2.6}$$

$$\Delta w_j = \eta(t_k - o_k)x_j \tag{2.7}$$

Where the new value of the weight is equal to the sum of the old value of the weight plus the delta of that weight. The delta of that weight is equal to the learning rate (η) times the result of the subtraction of the target/expected (t_k) value by the output value (o_k), times the input of that feature (x_j) (Bishop, 2007).

2.1.2.2 Multi-layer Perceptrons

Multi-layer perceptrons (MLPs) are artificial neural networks used to classify non linear data. Composed of three or more layers of fully connected perceptron(nodes), meaning that every node of the input or hidden layers is connected to all the nodes of the following layer. As can be seen in Figure 2.2, the layers are divided in three groups:

1. **Input layer** - receives the signal.
2. **Hidden layer(s)** - are the key to make the classification, because they transform the input into a useful result for the output layer.
3. **Output layer** - predicts/classifies the input.

Like the perceptron, the MLP is a supervised learning classifier, because it uses input-output pairs to train. The following equation (2.8) describes the output production of an MLP with one input layer and one hidden layer.

$$y(x, w) = f(\sum_{i=1}^{M} w_{kj} h(\sum_{i=1}^{D} w_{ji} x_i)) \tag{2.8}$$

As we can see, the equation can be divided into two parts: the f activation function and the h activation function. The h part is the activation function that each hidden layer node applies to the input weighted

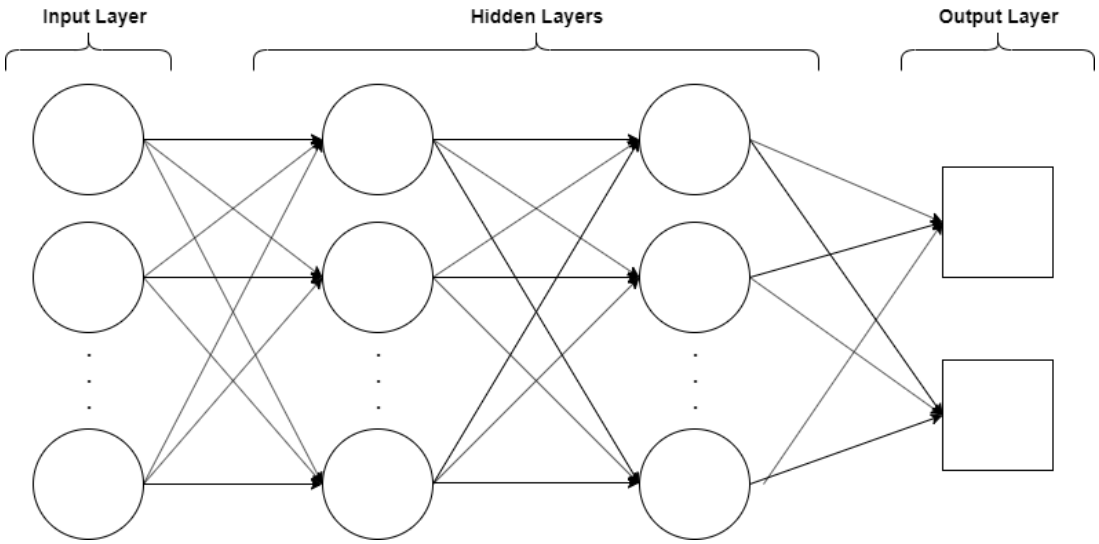

Figure 2.2: MLP model

sum. The f part is the activation function that the output layer applies to the weighted sum of the hidden layer nodes activation function result and the weights that connect the output layer. After producing the output a comparison with the real/expected output is made by calculating a loss function. After calculating the loss, the error is backpropagated in order to update the weights using the gradient descent algorithm.

2.1.2.3 Convolutional Networks

With the goal of creating models that are invariant to certain transformations of the inputs, a neural network was created with the invariance properties built into it. This is the basis of the Convolutional Neural Networks (CNNs). To understand CNNs we first need to understand what a convolution is. Convolution is an operation on two functions of a real valued argument which results in a third function that expresses how the shape of one is modified by the other. Convolution is the integral of the product of the two functions and it is represented by $*$ as we can see in the equations below:

$$s(t) = \int x(a)w(t-a)\,da \tag{2.9}$$

$$s(t) = (x * w)(t) \tag{2.10}$$

In network terminology, the first function (x) in the convolution is referred to as input and the second function (w) is referred as kernel, and the output of the convolutional operation is referred as the feature map. Since we typically use data at regular intervals, which means t can only take integer values we

discretize the convolution into:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.11}$$

Each element of the input and the kernel are stored separately so we assume that the functions are zero except for the finite set of points that represent the stored values allowing us to convert the infinite summation in equation (2.12) into a summation over a finite number of array elements. The next equation exemplifies a convolution with a two-dimensional input and a two-dimensional kernel:

$$S(i,j) = (K * I)(i,j) = \sum_{m} \sum_{n} I(i-m, j-n)K(m,n) \tag{2.12}$$

This equation, 2.12, is illustrated in Figure 2.3, where the discrete convolution is applied to a 4x3 tensor. The kernel in this convolution is a 2x2 tensor and since the stride value is 1 the result is a 3x2 feature map.

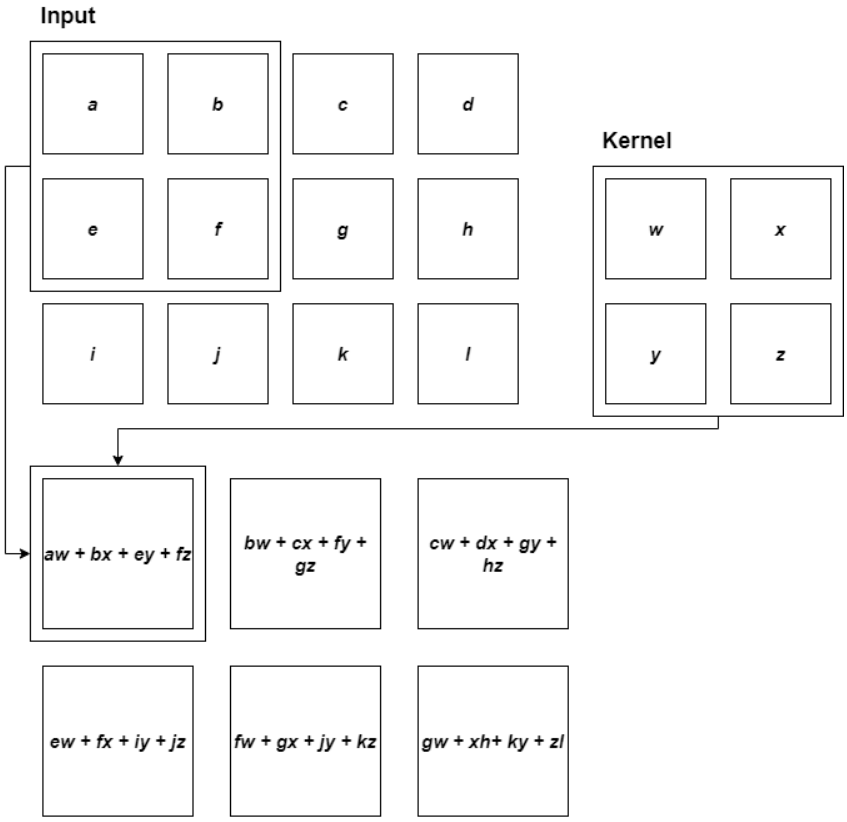

Figure 2.3: Convolution operation

Having defined the convolution operation in machine learning models, we now present the three ideas that are useful in making the model invariant to certain transformations of the inputs and capture deep contexts and features:

1. **Sparse Interactions**: unlike the MLP where every output unit interacts with every input unit, CNNs, have sparse interactions by making the kernel smaller than the input, as can be seen in Figure 2.4.

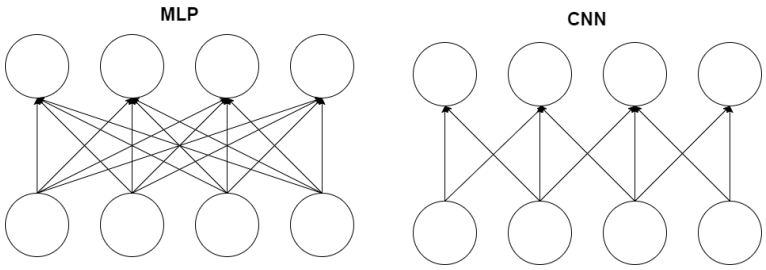

Figure 2.4: Units interactions between layers. On the left side we can observe the interactions of the feedforward MLP, where every unit connects to all the units of the following layer. On the right side we can observe the sparse interactions between the units of a CNN.

2. **Parameter sharing**: the same parameter is used for more than one function in a model. In CNN's each component of the kernel is used at every position of the input, learning only one set of parameters for all the locations.

3. **Equivariance to translations**: if we apply a translation to the input followed by a convolution it would be the same as if we apply a convolution and then the translation.

A Convolution Neural Network has several convolutional layers that apply other operations to the output of the convolution operation. This layer is illustrated in the following Figure 2.5 and its components are explained below.

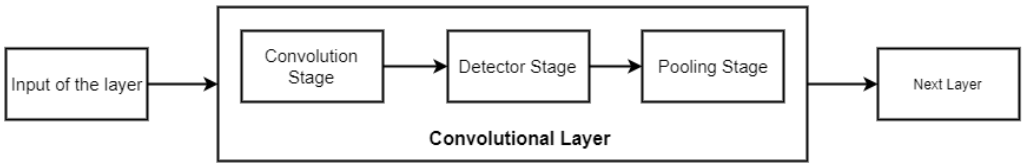

Figure 2.5: Convolutional Layer.

1. **Convolutional Stage** - several convolutions are performed in parallel to produce a set of linear activations.

2. **Detector Stage** - each of the produced linear activations is run through a non-linear activation function.

3. **Pooling Stage** - the pooling function is applied. The pooling function replaces the output of the layer at a certain location with a summary of the nearby outputs, generalizing and making the model more invariant to small changes in the input.

To train this network, error minimization is used through the use of backpropagation to evaluate the gradient of the error function. Having in mind that since this network relies on parameter sharing, the shared weights constraints imply a slight modification in the backpropagation algorithm, proposed by Goodfellow et al. (2016).

2.1.2.4 Generative Adversarial Neural Networks

Generative Adversarial Neural Networks were introduced by Goodfellow et al. (2014) revolutionizing the neural networks training paradigm with a zero-sum game between its two modules. These networks are mainly used in image generation and manipulation but its applications can go from simulations to classification. This networks are composed by two sub models: a generator, G, and a discriminator, D. The Generator goal is to capture the data distribution and generate outputs as close as possible to the real data distribution. The Discriminator function is to estimate the probability of its input being generated from fake or real data. To learn the Generator distribution p_g over data x, a prior on input noise variables $p_z(z)$ is defined and then a mapping to data space is represented as $G(z;\theta_g)$, where G is a differentiable function represented by a MLP with parameters θ_g. The second sub model is also defined by a MLP $D(x;\theta_d)$ that outputs the probability that x came from real data rather than p_g. The discriminator is trained to maximize the probability of assigning the correct label (from real data or from the Generator), while the generator is trained to minimize the difference between real data distribution and the generator data distribution ($log(1 - D(G(z)))$). This adversarial game that these models play is represented by the following equation defined by Goodfellow et al. (2014):

$$\min_{\theta_g}\max_{\theta_d} V(G,D) = \mathbf{E}_{x\ p_{data}(x)}[log(D(x))] + \mathbf{E}_{z\ p_z(z)}[1 - log(D(G(z)))] \tag{2.13}$$

The training of the network can be achieved by simultaneously updating the weights (θ_d,θ_g). The update of the Discriminator weights can be conducted by ascending its stochastic gradient of the loss function, i.e.

$$\nabla_{\theta d}\frac{1}{n}\sum_{i=1}^{n}[logD(x^{(i)}) + log(1 - D(G(z^{(i)})))] \tag{2.14}$$

and the update of the Generator weights can be conducted by descending its stochastic gradient of the loss function, i.e.

$$\nabla_{\theta g}\frac{1}{n}\sum_{i=1}^{n} log(1 - D(G(z^{(i)}))) \tag{2.15}$$

where n is the number of samples in each data batch during training.

Since the spatial interpolation problem relies on sampled data in order to interpolate the solution,

the noise vector z defined above needs to be replaced by the sampled data, which the generator takes as input and interpolates. Thus, we need to extend the GAN to a conditional version named Conditional GANs (CGANs), since the generator and the discriminator are both conditioned by the same auxiliary information, which can restrict both the generation and the discrimination processes.

For spatial interpolation problems, the traditional adversarial strategy described in equation (2.13) needs to be modified to ensure the stability of conditional generations. The random noise vector z should be removed such that the conditional generation could be considered to be determined by the sampled data as the only constraint. When training an adversarial spatial interpolation net, the generator requires the sampled image $f(x)$ as input in order to output a generated fake image $G(f(x))$ as similar as possible to the real image x. Adding to that, the discriminator needs to be trained to distinguish the fake image $G(f(x))$ from the real image x based on the sampled image $f(x)$. The min max game is then altered to:

$$\min_{\theta_g} \max_{\theta_d} V(G, D) = \mathbf{E}_{x\ p_{data}(x|f(x))}[log(D(x, f(x)))] + \mathbf{E}_{z\ p_{data}(x|f(x))}[1 - log(D(G(f(x)), f(x)))] \quad (2.16)$$

Where G is a differentiable equation representing the Generator's structure and it's parameters θ_g and D is a differentiable equation representing the Discriminator's structure and it's parameters θ_d. G attempts to approximate $p_g(G(f(x))|f(x))$ to $p_{data}(x|f(x))$ in the real dataset, minimizing the second term of the equation above. At the same time, D evaluates if the spatial image comes from $p_g(G(f(x))|f(x))$ or $p_{data}(x|f(x))$, maximizing both terms of the equation. These changes make the adversarial spatial interpolation learning to approximate the conditional generative probability distribution given spatial sampled images $p_g(x|f(x))$ rather than the probability distribution of data existence (p_{data}). (Goodfellow et al. (2014); Zhu et al. (2019)).

2.2 Related Work

This section overviews important related work in the area of spatial interpolation and is divided in three parts:

1. How additional information can be included in the classic IDW model and the study of the performance when compared with the traditional IDW model.

2. How machine learning applied to spatial interpolation evolved from a MLP with the power to match classic techniques to a conditional encoder-decoder GAN model that is able to capture deep spatial features in images.

3. How to augment a neural network ability to generalize in order to become more robust to different contexts.

2.2.1 Spatial Data Interpolation with Least-Cost Path Analysis

IDW is one of the most used and basic spatial interpolation techniques. As explained in the previous section IDW is a deterministic procedure and estimates unknown values by applying the formula present in equation (2.4). After analysing that formula, we can observe that the value to be estimated strongly depends on the distance (from the know data point, to the point to be estimated) value. Traditionally the distance is based on Euclidean estimates of distances, but the need to represent the connectivity between complex geographic data, which can fail when represented by Euclidean distances, gave raise to IDW using non-Euclidean distances. Non-Euclidean distances are optimal path solutions obtained by applying a path search algorithm to the cost surface raster, where each cell contains the cost to traverse to the eight surrounding cells.

In the study of Least Cost Path Analysis, Greenberg et al. (2011) presents us with the use of non-Euclidean distances in the IDW formula and compares both Euclidean and Non-Euclidean distances by interpolating water temperatures in the Sacramento-San Joaquin River Delta. To achieve that they randomly selected 18 gauges out of 35 to derive the least cost path distance where the cost distance was equal to the cell size for the water cells and assigning the land cells as barriers with an infinite cost. The other 17 gauges where used to tune the model parameters by minimizing the sum-of-squares error between the predicted values and the values measured in those 17 test sites. After comparing the best procedure of IDW using euclidean and non-euclidean distances, Greenberg et al., observed that since the euclidean interpolator does not contain land barriers, points that are close to each other in an euclidean Space but in reality aren't connect for a long distance, traversing only by water points, share similar values to one another, which may be very detrimental in certain interpolations. Another issue that this work approaches is that the improvements offered using non-Euclidean distances depends on the nature of variables to be interpolated and their degree of connectivity and by the assigning of costs to the surface.

More recently, Chris Madden and Joseph Stachelek Stachelek and Madden (2015) applied a least cost path analysis to interpolate water quality values in the coast of Florida Bay in USA, comparing inverse path distance weighting (non-Euclidean) with its euclidean counterpart IDW.

To assert the salinity values off the Florida Bay coast, 23 dataflow surveys from 2006-2012, where 4 surveys were conducted in the dry season (May - June) reaching hypersaline values (>40) and 4 surveys were conducted in the wet season with salinities close to fresh water (<1), with all surveys registering a minimum, maximum and range values. After interpolating using each method, the salinity value

using each dataflow survey per interpolation, Madden and Stachelek concluded that the non-Euclidean method provided increased accuracy and resolution of water quality features compared with the Euclidean method. This increased accuracy was visible in two ways.

First, for both methods, when the range of measured salinities was larger (>25) the interpolation errors (mean absolute error) increased for both methods, but evidencing that the non-Euclidean method provided better results, since as the ranges increased the spatial gradients became more intense and non-Euclidean method can recover location and shape of those locations, providing a more accurate estimation.

The second point in which the non-euclidean procedure was beneficial is related to the fact that it preserved the separation of non-shore embayments in which water bodies have distinct characteristics and are close from the ones they interpolated. Since inverse path distance weighting uses and is familiarized with the concept of barriers these areas do not affect the interpolation accuracy. Finally, another interesting conclusion is that in open areas inverse path distance weighting provided little benefit, since the presence of barriers is close to zero.

With these studies we can conclude that using the IDW, using a non-euclidean distance can be better than using euclidean distance because with non-euclidean distances we can adapt to the problem needs, by defining traverse costs and barriers. This proves to be very useful and more accurate, but having always in mind that the real problem is to define the costs and barriers because if not assigned properly, then euclidean distances prove to be better and more consistent.

2.2.2 Deep Learning for Spatial Data Interpolation and Generative Adversarial Networks

Traditionally, a wide variety of globally and locally computed techniques are used to solve interpolation problems. Since those problems usually consist in sparsely distributed data, incorporate every available information is extremely important. This results in mathematically more complex and unique interpolation methods. In addition to that it is beneficial to incorporate additional understanding of natural phenomena. Unlike traditional methods, neural networks can easily incorporate additional information to the estimation process.

In 2001, Rigol et al. (2001), used feed-forward backpropagation networks to interpolate air temperatures in the UK having using meteorological data from 1986. Out of a total of 195 meteorological stations, 146 stations provide daily observations, throughout the year, and the other 49 provided incomplete records and were only used in the testing phase. Also, they selected a total of 34 topological and

geographical information as additional input data. The data provided by the 146 stations were used to select the best architecture of the 3-layered (MLP) having split the data in 80% for training and 20% for testing, adding the data from the 49 stations with incomplete data. Three experiments were conducted, developing three different forms of MLPs, selected by sequentially using networks with increasing number of hidden layer units (each one with logistic function as activation function) and minimizing the least squared error function, until no significant improvement in the validation error was found.

In the first experiment, they estimated the temperature using terrain variables and weather type by using a selected 33-1-1 MLP. After training and testing the network was able to capture 60% of the data variability and achieved a reasonable accuracy with mean absolute error of 2.5^{o}C and root mean squared error of 3.2^{o}C. In the second experiment, the interpolation was achieved by using neighbouring observations. A 19-4-1 network was selected and produced better accuracy than the previous experiment having a mean absolute error and root mean squared error of 0.9^{o}C and 1.2^{o}C. The last experiment estimated the temperature using neighbourhood observations and terrain variables. The select network was formed by 45 input units, 4 hidden layer units and 1 output unit, and again this network produced better results than the previous experiments with 0.8^{o}C and 1.1^{o}C mean absolute error and root mean squared error values.

Overall, we can see an improvement in the results from the first to the last experience, since the last is a combination of the first two experiments, with the weather type having almost no weight in the first experiment results. But looking at the bias values of all experiments we can observe that all the trained networks underestimated minimum temperature values, since they are negative. Even though, the network of experiment 3 provided a better result in estimating minimum temperatures than the other two.

After conducting the experiments described above and since the authors did not directly compare the results with the classical methods, they concluded that the interpolation using neural networks is a viable technique, since they offer methods of interpolation without some of the drawbacks of classical methods, like linearity assumptions and the need to pre-specify the division between trend and covariance in the model. The studied networks provided models for non-linear relationships in the data with the model accuracy depending on the selection of the proper additional information, because that is the main advantage of this technique, allied with the fact that no assumption about the nature of the data is made.

Since this study, conducted in 2000, there has been an increasingly development in neural networks with more precise and complex models that can be applied to countless areas and spatial interpolation is no exception. A very good example of an extremely efficient neural network is the CNN, considering that it's local connectivity and shared weights enable the model to concentrate on features that are near to each

other as well as features that aren't close, representing the complex nature of the data, that traditional models can not represent.

A powerful deep learning approach for training generative models was introduced in 2014, by Goodfellow et al. (2014). The GANs has the potential to solve the interpolation problem and approximate the data relations, because it adopts an adversarial structure to train the loss and therefore avoiding the difficulty of approximating many intractable probabilistic computations.

In 2019, Zhu et al. (2019) explored and studied the potential of the GANs to interpolate terrain elevations using a dataset of digital elevation model (DEMs) in China. The GAN model proposed by Goodfellow et al. (2014) is a primitive and generic form of adversarial network that takes a MLP for the generator and the discriminator and utilizes a noise vector as input for the generator instead of sampled data to condition the solution. To adapt the primitive GAN to condition the generation using sampled data and to take advantage of the convolutional operation positive aspects when dealing with images, Zhu et al. proposed a Conditional Encoder-Decoder GAN (CEDGAN). The encoder-decoder architectures rely on the convolution operation to encode the input into deep feature maps and then on the deconvolution operation to upsample/decode the deep features into full-size image representations, capturing deep spatial representations.

After the proposed CEDGAN was trained they interpolated 9 different sampling configurations using the CEDGAN and IDW and for all of them the average interpolation error at the pixel level was lower for the CEDGAN as well as the computing speed that was approximately 1000-times lower in the CEDGAN. In addition to that, they observed that the images produced by CEDGAN (i.e., fake images produced by the generator) are very similar to the real ones, whereas IDW produced very blurry images. The reason why the Generator outperforms the benchmark spatial interpolation methods in accuracy and visual fidelity is that it is trained through an encoder-decoder structure that captures local geographical structure patterns. Essentially, the encoder looks for relationships among the sampled locations and the decoder assembles structural spatial patterns (valleys and ridges) and outputs a spatial distribution. Instead of only remembering training samples, the CEDGAN-based spatial interpolation model captures complex spatial features underlying the given spatial dataset, which then could be applied into other domains with different distribution patterns but similar spatial features. Since the model is designed to capture spatial dependencies as basic knowledge, it fails to predict values at locations where the spatial attributes vary too quickly, as the authors show by doing a slope analysis, where the CEDGAN does not preforms well at reproducing a very steep slope. This pre-trained model can be applied across domains if the spatial features in the new domain can be considered roughly similar to the ones of the previous domain, but if the features are too different from the previous domain additional data may be needed to improve its performance. Another limitation this model encounters is that this deep learning framework

requires a lot of training data, such that the training captures the complex spatial patterns and often that data is not available. Even with this limitations this model outperforms traditional techniques like IDW and ordinary kriging.

Deep learning techniques have seen an improvement through the years. The MLP studied in 2001 by Rigol et al. (2001) had the power to match or perform a little better than the classical geostatistical techniques when solving the spatial interpolation problem. Nowadays, with much more complex deep learning models, the power to outperform the classical methods matched by the MLP is only limited by the amount of available data to train the model and by the capacity of the model to capture the complex spatial features and relations present in the data.

2.2.3 Generalizing Deep Learning Models

Since the generalization capability of well-trained models is limited by the amount of data available and by the dataset, seeing that it learns in a specific location and if used in a different location may not work well. This is commonly known as the cross-location generalization problem.

In the end of 2019, Deng et al. (2019) decided to approach this problem in the context of Overhead Image Segmentation. They took the fact that objects are not laid randomly on the Earth surface and that it does not have a big variation from location to location, and they question how prior information of the fundamental rules of geography could be incorporated into the architecture of a CNNs, used for Image Segmentation. After consideration, they came up with Getis-Ord $G*_i$ pooling, which is a pooling method based on spatial Getis-Ord $G*_i$ analysis of CNN feature maps. Getis Ord $G*_i$ analysis is a technique for geo-spatial clustering that is used to encapsulate the fundamental rules of geography, especially the first law of geography, that states that everything is related to everything else, but near things are more related than distant things.

As we have seen in the previous section, pooling is used by CNNs to downsample the feature maps. Like other pooling methods, a stride sliding window is used to downsample the input. Given a feature map within the stride window, we denote the feature values within the sliding window $X = x_1, x_2, \hat{a}, x_n$ where n is the number of pixels in that window. Consider that the feature value is x_i and if the center does not coincide with a pixel location then x_i is the average of the adjacent values. Let $p^x(x_j)$ and $p^y(x_j)$ denote x and y position of the pixel x_j in the image plane. A weight matrix w that measures the Euclidean Distance on the image plane between x_i and the other locations in the sliding window is computed as

$$w_{i,j} = \sqrt{(p^x(x_i) - p^x(x_j))^2 + (p^y(x_i) - p^y(x_j))^2} \tag{2.17}$$

The Euclidean distance is then used to computed the Getis-Ord, $G*_i$, value as follows:

$$G*_i = \frac{\sum_{j=1}^{n} w_{i,j}x_j - \overline{X}\sum_{j=1}^{n} w_{i,j}}{S\sqrt{\frac{[n\sum_{j=1}^{n} w_{i,j}^2 - (\sum_{j=1}^{n} w_{i,j})^2]}{n-1}}} \tag{2.18}$$

Where $\overline{X}$ and S are:

$$\overline{X} = \frac{\sum_{j=1}^{n} x_j}{n} \tag{2.19}$$

$$S = \sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n} - (\overline{X})^2} \tag{2.20}$$

Spatial Clusters can be detected based on the G_i^* value, where the higher the value the more significant the cluster is. G_i^* only indicates if a spatial cluster exists and in order to achieve their pooling goal, they summarized the local region of the feature map to a representative value they used. A threshold was used to achieve the pooling goal, where the G_i^* value is evaluated and if it is greater or equal to the threshold, a spatial cluster is detected and the value x_i is used, else the maximum value in the window is used, like max pooling. This G pooling function was used in a pre-trained network commonly used for semantic segmentation tasks, the VGG network.

Deng et al. (2019) then applied this network to a dataset with overhead images of two German cities: Potsdam and Vaihingen. After training and testing the network with the proposed Pooling function they compared it with equally tested networks with other pooling functions (Max-pooling, stride conv and P-pooling), they observed that incorporating geospatial information into a pooling function of the standard CNN can improve segmentation accuracy. In Potsdam G-pooling had a better accuracy in segmenting buildings, low vegetation, trees, cars and pixel accuracy, only performing worse than P-pooling in segmenting Roads. In Vaihingen, P-pooling had better accuracy (82.44 compared with the 81.78 from G-Pooling) and segmented better buildings and trees with a little margin and in the other labels G-Pooling was better. This results are related with the use of geospatial statistics that G-pooling use in comparison with non-spatial statistics like max or average value in a sliding window.

The key aspect that the authors wanted to demonstrated and succeeded, was the generalization capability of their pooling function. They first trained the network using only Potsdam images and then tested it in Vaihingen and then performed the other way around. In the Potsdam->Vaihingen adaptation G-pooling performed better in all labels and accuracy except for the segmentation of trees in which Max-pooling performed better. In the Vaihingen->Potsdam adaptation G-pooling performed in the same way as the Potsdam-Vaihingen adaptation and the trees label was ”won” by the P-pooling function. This performance is again due to the incorporation of knowledge.

Having that, the authors could conclude that since deep-learning is a data-driven approach, it relies on data to train, where a model that trains with a large amount of diverse data will have a good generalization ability. G-pooling or additional information (related with the model problem) are a good solution to increase the generalization of the model, especially where the training data is not abundant, making the model perform better when it is shifted into a different domain.

Marcos et al. (2018) studied the task of semantic labelling. Semantic labelling, consists in corresponding to the automatic assignment of each pixel to a set of predefined land-cover or land-use classes, being the classes selected specifically for the task and define the learning problem for the model. When the image resolution increases the number of pixels that define the image also increases. Due to this trade-of, single pixels tend not to contain enough information to be confidently assigned to its correct semantic class or, in some cases, classes can be semantically ambiguous, when relying on spectral characteristics only. To get around these problems there is the need to consider the spatial context.

CNNs are the state-of-the-art when focusing on image data, thanks to their ability to learn complex problem-specific features, having the problem of being a data-driven model that requires large amounts of ground truth information to be trained. Facing that drawback, the authors decided to exploit the arbitrary orientation of the objects, having in mind that overhead images differ from the natural images in the fact that the absolute orientation of objects and features in the image tends to be irrelevant for most tasks, including semantic labelling. CNNs are equivariant (e.g. a bird is always a bird no matter the orientation of the image) to the rotation of inputs if they are trained with randomly rotated input images (i.e. data augmentation). As long as the CNN is trained with samples in a sufficient number of orientations it will learn to be invariant to rotations. But this kind of data augmentation does not offer any advantage in terms of model compactness, since similar filters, but with different orientations need to be learned separately. Another recent method explores ways of encoding model-based rotation invariance into CNNs, where a rotation of the input image is performed in order to reduce the sample complexity of the problem and extend this to affine transformations. Although providing invariance to a global rotation, this method does not provide invariance to relative rotations being unsuitable for segmentation tasks. Two segmentation tasks suitable methods rely on: encoding equivariance to shifts and to rotations by multiples of 90^{o} by tying filter weights; use linearly interpolated filters. Inspired by these two methods the authors decided to combine both methods. The combined method applies an arbitrary number of rotated instances to each filter at every location, in a way that each filter activation is composed by a vector of activations. Then the authors proposed to apply the max-pooling function to these activations, compressing the information in a simple vector that represents the magnitude and orientation of the maximally activating filter, allowing to encode very sensitive rotation invariance, while allowing more expressive filters, as well as decreasing the sample complexity.

To implement rotation equivariance in the CNN they implemented Rotation Equivariant Vector Field Networks (RotEqNet) Modules into the CNN architecture. RotEqNet involves rotating CNN filters and pooling across orientation space to the retrieve maximal activations and their angle observed at each location and per filter. To implement this behaviour into the CNN: firstly, they implemented rotating convolution; secondly, they implemented orientation pooling. Rotating convolution is achieved by calculating rotated versions of each main filter at R orientations and outputting a set of R activations per filter. In order to avoid the explosion of dimensionality a pooling function is applied to every output filter set propagating to the next layer the information relative to the direction of maximal activation.

To test their proposal, Marcos et. al. used 2 datasets regarding two cities: one in Vaihingen, Germany; and the other one in Zeebruges, Belgium. After training and tested the proposed network and the normal CNN their compared the results.

For the city of Vaihingen both models reached an overall accuracy of 87%, when using the whole dataset for training and when using only 4% of the dataset for training the overall accuracy drops to 84.7%, since the data is highly redundant. When looking at the global average, RotEqNet performed better than the CNN, because it detected cars with a 72.6% precision, while the CNN only predicted cars with a 54.5% precision. This means that RotEqNet is better at detecting objects with clear and consistent boundaries. RotEqNet does not need to learn filters that are rotated versions of each other because all these versions are explored by applying each filter at many different orientations. While standard CNNs require data augmentation to perform well in a rotation equivariant setting, RotEqNet extracts features at different orientations and keeps the largest activations, effectively analysing the input at different orientations without rotating it explicitly, and allowing the model to focus more on solving the semantic labelling task. For the city of Zeebruges the results performed in the same way as in the city of Vaihingen, where the accuracy of labelling cars and houses also stood out.

After looking at all the results we noted two main tendencies and important conclusions:

- Explicitly encoding rotation equivariance in deep learning dense semantic labeling models allows for much smaller models, between one and two orders of magnitude compared to traditional CNNs.
- A CNN encoding equivariance in its structure, rather than through data augmentation, also provides robustness against varying amounts of training data.

3 A Hybrid Approach for Spatial Data Interpolation

This chapter presents and explains the proposed spatial interpolation approach in Section 3.1. In addition to detailing the hybrid approach, we describe the data and how it is processed in Section 3.2. Finally, 3.3 overviews this chapter contents.

3.1 The Proposed Hybrid Algorithm

Previously, in the Related Work section, we exposed the IDW technique and how variations in the calculation of the distance parameter in the equation (2.4) can be used in certain cases to overcome or attenuate some of the disadvantages of this technique when using euclidean distance. In a more modern approach, we detailed the potential that neural networks could have in solving the spatial interpolation problem. The potential provided by the MLP in solving the spatial interpolation problem, was confirmed when new neural networks were able to capture complex features and patterns in images, allowing models to apprehend and generate/interpolate new images conditioned by sampled information. Machine learning went from being as good as a traditional technique, to overcome those traditional techniques. The last subsection of the Related Work focuses on the challenge of generalization of neural network used to interpolate spatial data and to maintain its performance in contexts different from the ones the network was trained in.

Our approach main focus is the architecture proposed by Zhu et al. (2019). Like Zhu et al. (2019), we propose a conditional encoder-decoder GAN combined with the IDW technique. This hybrid approach has the goal of overcoming the IDW technique alone and to achieve a higher generalization degree than the simpler network. As every GAN we can divide the network into two sub-models, the generator and the discriminator, which we will detail in the subsections below.

3.1.1 The Proposed Generator

The generator can be divided into an encoder and a decoder. The encoder takes the sampled patch as input and forwards it to three two-dimensional convolutional layers (conv 1, 2 and 3). The decoder part has three two-dimensional transposed convolutional layers (deconv 1, 2 and 3), that upsample the

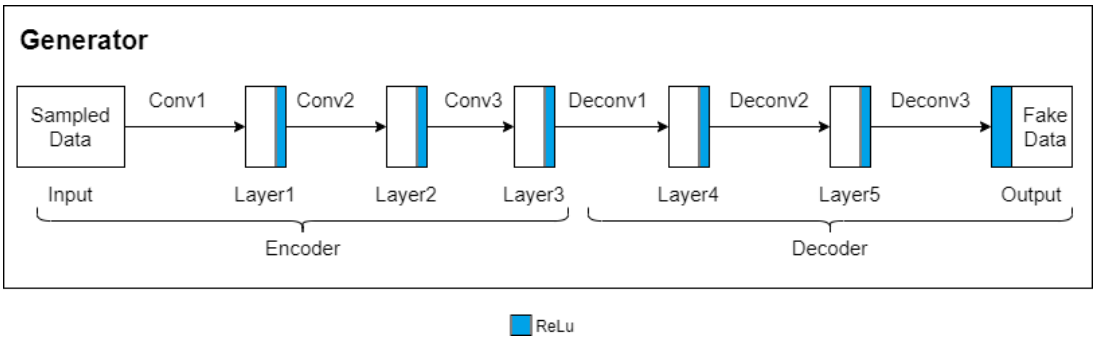

Figure 3.1: Architecture of the generator sub-model.

previously encoded feature maps. After each of the convolutional or transposed convolutional layers a ReLU activation function is applied, as we can observe in Figure 3.1.

The settings used for each encoder layer are the same ones used by each decoder layer, which means they share the same convolving 5x5 kernel and stride, valued as 1, and perform a zero-padding with the given kernel and stride.

3.1.2 The Proposed Discriminator

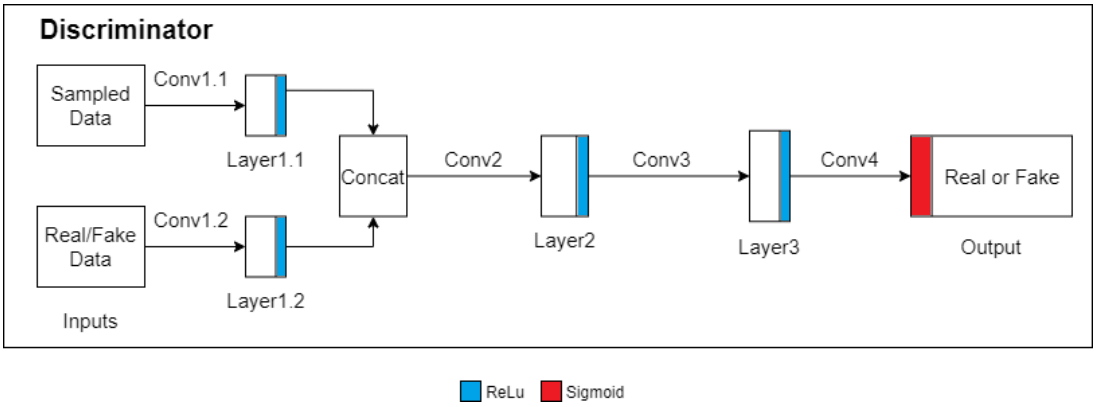

Figure 3.2: Architecture of the discriminator sub-model.

The discriminator takes two inputs, the sampled patch and the corresponding real patch or the patch generated from the corresponding sampled patch. Each input is then forwarded into an independent two-dimensional convolutional layer (conv 1.1 and 1.2). The feature maps that outcomes the convolutional layers (conv 1.1 and 1.2) are merged using a concat operation. The outcome of the concat operation is then passed through three two-dimensional convolutional layers (conv 2, 3 and 4). After each of the convolutional layers the ReLU activation function is applied, except in the last convolutional layer, where a Sigmoid activation function is applied, in order to obtain a scalar from 0 to 1 indicating that values closer to 1 represent a strong belief on the discriminator that the patch is real and values closer to 0 express the discriminator belief that the patch is fake/interpolated. The discriminator sub-model, can

be represented as Figure 3.2 demonstrates. The layers 1.1, 1.2, 2 and 3 share the same kernel and stride sizes and perform a zero-padding convolution with the convolving 4x4 kernels and a stride of 2. Layer 4 does not pad the feature maps derived from the layer 3 and has a stride, different from the previous layers, valued as 1.

3.1.3 The proposed Hybrid Algorithm

As mentioned before the hybrid algorithm combines the network we just described with IDW. This combination is achieved by replacing the one-channel input of the generator and the corresponding input of the discriminator by a two-channel input composed by the sampled patch and by the corresponding patch interpolated using IDW. Despite the interpolated patch on the second channel can be labeled as inaccurate information we believe it will improve the network ability to capture the spatial contexts, since some of the spatial features can be found in the interpolation by IDW, and the network can then learn new ones and improve the capture of the spatial features exposed in the patch interpolated by IDW. With this combination we provide even more information, hoping to overcome the IDW technique and the network that requires only a one-channel input. Besides overcoming other techniques we also focus the training behavior of the network and its generalization ability.

3.2 Sources of Ancillary Data

From all the innumerable variables that can be represented in a spatial context, we selected the elevation of the terrain as a variable to be displayed in space. This choice was due to the amount of available data, since the land in all of the globe is available with high resolution (25 meters)[1] and due to its appearance in the literature related with the spatial data interpolation challenge.

Since we live in Portugal, we decided to focus on the digital elevation model (DEM) of mainland Portugal and its autonomous regions, Azores and Madeira. Being mainland Portugal located in Europe and its autonomous regions placed in the Atlantic North we relied on digital elevation models of the European Union's Earth observation programme[2]. After obtaining the rasters corresponding to the mainland Portugal and islands, we merged them using gdal-merge[3]. To cut the merged raster into the mainland Portugal and each individual island that composes the archipelagos of Azores and Madeira, without considering the islands of Desertas , Selvagens and small islanders, we used gdalwarp[4] with the

[1] https://earthexplorer.usgs.gov/
[2] https://land.copernicus.eu/imagery-in-situ/eu-dem/eu-dem-v1.1
[3] https://gdal.org/programs/gdal_merge.html
[4] https://gdal.org/programs/gdalwarp.html

shapefiles (i.e., a popular format for geographic information information system) of the boundaries of these regions provided by Direção-Geral do Território[5].

Having extracted and processed the data, we are left with a dataset of twelve digital elevation models corresponding to each island of the Portuguese autonomous regions (nine Azores islands and two Madeira Islands) and mainland Portugal.

3.3 Overview

This chapter presented our proposed hybrid approach for spatial data interpolation that was considered in this dissertation. Concretely, it first presented the general network model inspired in the work proposed by Zhu et al. (2019). Then the chapter presents our modification to combine the modern network with the traditional IDW approach. Finally we present the ancillary data sources used to support the spatial interpolation problem and the study of our technique.

[5] https://www.dgterritorio.gov.pt/dados-abertos

4 Experimental Evaluation

This chapter presents the experimental evaluation of the hybrid approach to solve the spatial interpolation problem, which involved training and validation with spatial data of mainland Portugal and its autonomous regions. In the first place, we present the evaluation methodology together with a description of the metrics used to assert the error of the spatial interpolation procedures (Section 4.1). In the second place, we explain the sampling process we used to select the known values that were used in the interpolation process (Section 4.2). In the third place, we present the interpolation of the sampled data using the IDW and the used parameters (Section 4.3). Finally, we expose the training behavior divided by sampling configuration of each network and the results obtained by interpolating the sampled data with the simpler GAN and with the hybrid approach (Section 4.4). The sampling process and the IDW technique were implemented using python whereas the neural networks were implemented using PyTorch, a deep learning framework in python that relies on GPU acceleration.

4.1 Methodology and Evaluation Metrics

Spatial Interpolation is an estimation of the unknown values based on a sample of known values. Like in every estimation task the goal is to get as close as possible to the real values. To evaluate how close the interpolated values are to the real ones, we selected two metrics that are widely used in the literature regarding this problem. The root mean squared error (RMSE) and the mean absolute error (MAE) are the selected metrics and their mathematical representation is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \tag{4.1}$$

$$MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \tag{4.2}$$

The equations described above take the n number of predictions and calculate the average of the absolute difference or squared average of the squared difference between the $\hat{y}_i$ representing a predicted value and y_i corresponding to a true value. As we can observe, equations (4.1) and (4.2) emphasize

different aspects of the model performance. While the MAE assigns the same weight to all errors, the RMSE penalizes variance by giving errors with bigger absolute values more weight when compared to smaller absolute values. In order to obtain a better analysis, like in various studies regarding this problem, we selected these two metrics since they can provide a varied picture of the error distribution.

4.2 Sampling Process and Configurations

In order to study the behaviour of the IDW technique, the GAN and our proposed hybrid approach, we sampled the DEM referring to mainland Portugal and the Portuguese autonomous regions using three different sampling configurations. These configurations consider only the points with values above the sea level (i.e., height ≥ 0 meters) and distribute the known points uniformly, where the percentage of known points relative to the total of points in the first configuration is 5%, in the second configuration is 10% and 15% is used for the third configuration. Like in the work of Zhu et al. (2019), the network take as input patches with a size of 32x32 pixels. For practical reasons we created one mask per configuration with the same size, where the known points are valued as 1 and the unknown points are valued as 0. Thus, we have a total of 1024 (32*32) points per mask and the amount of known points for each configuration is:

- Configuration 1 - with 49 known points, where $49/1024 = 0.0478515625 \approx 5\%$
- Configuration 2 - with 100 known points, where $100/1024 = 0.09765625 \approx 10\%$
- Configuration 3 - with 144 known points, where $144/1024 = 0.140625 \approx 15\%$

To uniformly distribute the known points m in the mask of size W x H, we assigned the value 1 to each coordinate (c_i, r_j), where:

$$\begin{aligned} c_i &= c_1 + (i-1)\delta_w, \\ r_j &= r_1 + (j-1)\delta_h \\ \forall i, j &= 1, ..., \sqrt{m} \end{aligned} \tag{4.3}$$

where the initial observed point (c_i, r_j) is (0,0), $\delta_w = (W-1)(\sqrt{m}-1)$ and $\delta_h = (H-1)(\sqrt{m}-1)$.

Having formally defined the masks corresponding to each sampling configuration we illustrate in Figure 4.1 the known points in white and the unobserved points in blue.

The sampling process is only complete when the masks are combined with the rasters that constitute the dataset. This combination is achieved by sliding the 32x32 masks through the rasters, keeping only

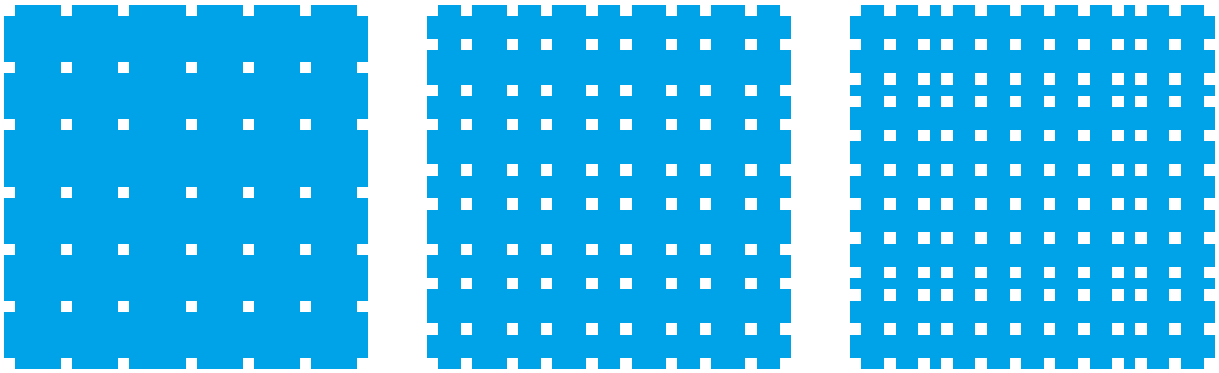

Figure 4.1: Illustration of the three sampling masks described above, where, from left to right, we have the 49 known points configuration, the 100 known points configuration and the 144 known points configuration.

the real observed values and valuing the unobserved locations as 0. This operation can be described as a convolution where the 32x32 kernel/mask slides through the raster using a (32,32) stride. After performing this operation for each raster and using all the three masks, the sampling process is complete and per each real raster we have three sampled rasters, one for each sampling configuration.

4.3 Interpolation using Inverse Distance Weighting

In this study we implemented the IDW approach as defined in equation (2.4), where we used the euclidean distance for the parameter d and the power parameter p with the value of 2, which is the most common value used in the literature and simplify the calculations with the euclidean distance.

In addition to the parameters described above, we set the number of known points used to interpolate an unknown point to the 100 closest points. This measure was taken since in bigger rasters, as the case of mainland Portugal, the amount of known points is in the scale of millions and we could not interpolate all of them, for each unobserved point, with the available amount of time.

After interpolating the rasters for each sampling configuration, the new interpolated rasters were stored to so they can be used combined in the two channel input and enabling the study of our hybrid approach.

4.4 Interpolation using a Generative Adversarial Network and the Proposed Hybrid Approach

This section is divided into two subsections related to the two stages of a neural network: training and validation. The first subsection exposes and relates the training behavior, for each sampling configu-

ration, of the 1 channel input GAN and our hybrid approach. The second subsection relates the obtained results for each previously trained network. For this study we chose to use only land patches, restraining and forcing the networks context to focus on land patches instead of land and coastal patches. In addition we also wanted to reduce the training duration, by having a smaller dataset.

4.4.1 Adversarial training procedure

The networks were trained using mini-batch stochastic gradient descent (SGD) with a batch size of 64. The selected training dataset was restricted to the 32x32 patches composed uniquely by values above the sea level (height ≥ 0) of the mainland Portugal raster and we saved the other regions including the mainland Portugal raster to validate the trained generator. Having that, we split the training dataset of 135577 patches into a training set of 115240 patches (85 % of the dataset) and into a test set of 20337 patches (15 % of the dataset). We used the Adam optimizer, where β_1=0.9, β_2=0.999, and the learning rate, α, for backpropagation was set to 0.00001. The update of the Discriminator and Generator weights was achieved using the binary cross entropy loss following the equations (2.14) and (2.15). We trained the network for 200 epochs and after an epoch was complete we tested the generator using the test set. To measure the performance of the generator after each input we calculated the average RMSE of the test set patches and if the performance of the generator did not improve for 10 consecutive epochs we halted the training process and considered the training to be finished. For each sampling configuration we train two networks the network where the input is composed by a sampled patch and the network where the input is composed by the sampled patch in one channel and the correspondent patch interpolated using the IDW technique in the other channel. The following sub subsections deepen in the networks training behavior for each sampling configuration, where we compare the one channel GAN on the left with our hybrid two channel GAN approach on the right.

4.4.1.1 Configuration 1 - 49 observed points

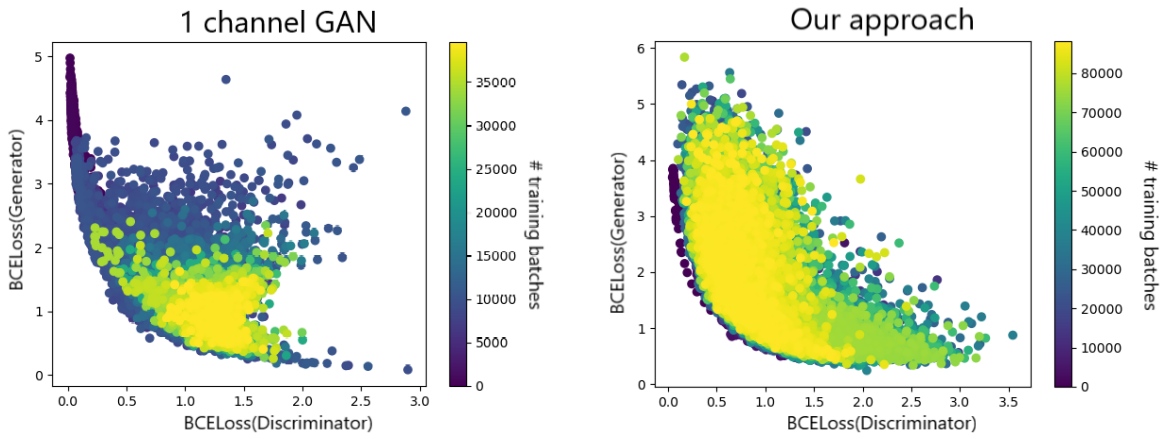

Figure 4.2: Scatter plots representing the adversarial game between the Discriminator and the Generator during the whole training process, where the focus is in the adversarial game evolution and its final state.

Figure 4.2 shows at the left plot we can observe that the majority of the discriminator losses is in the]0, 2] interval and most of the generator losses is in the]0, 4] interval. In the right plot the core of the discriminator losses vary between]0, 3] and the mass of generator losses fluctuate between]0, 5]. As the number of batches fed to the network increases we can observe a small cluster on the left plot and a bigger cluster in the right plot. The cluster on the left plot is restricted to a discriminator loss between]0.8, 1.5] and a generator loss of]0.5, 1.5]. The cluster on the right plot is restricted to a discriminator loss between]0.3, 1.5] and a generator loss of]0.5, 4]. Comparing the two clusters we can observe that the left cluster is much smaller, meaning that the adversarial game between the two sub-models achieved an equilibrium, while the adversarial game played between the two sub-models of our approach have not achieved a cluster so restricted like the left plot, we can state that a wider equilibrium in the right plot was achieved by focusing on the evolution of the adversarial game from older batches to the more recent ones.

Comparing the two plots on Figure 4.3 we can observe that:

- The left plot had a maximum accuracy of above 40 meters (6th epoch) while the right plot correspondent to our hybrid approach peaked an error below 35 meters (4th epoch);
- In terms of minimum accuracy the left plot performed an error close to 5 meters per patch (12th epoch) while our approach represented on the right plot performed a minimum accuracy close to 3 meters (39th epoch);
- The training process lasted for 22 epochs for the one channel GAN, meaning that the network did not improved since the 12th epoch. Looking at our approach, the training process lasted for 49

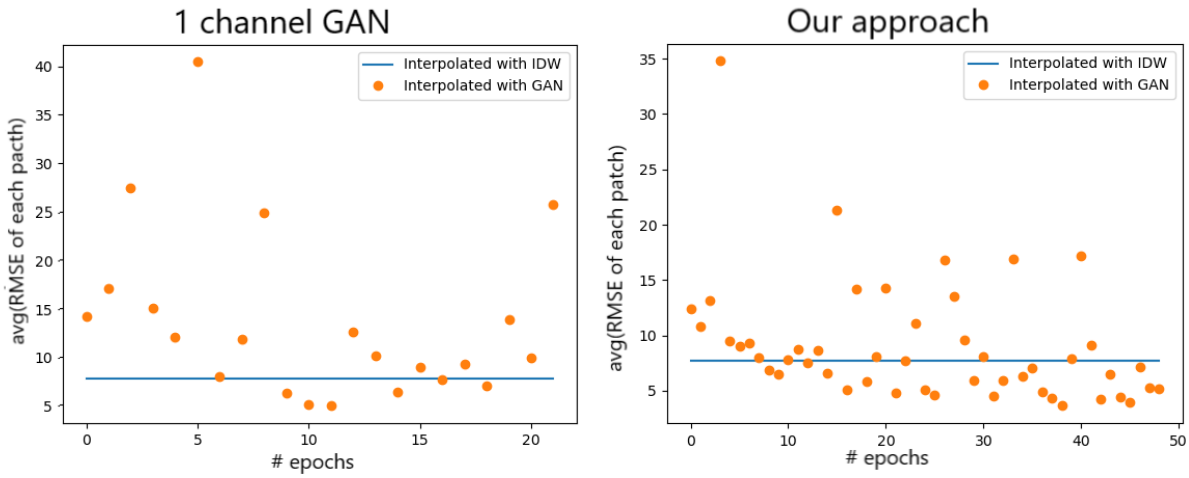

Figure 4.3: Variation of the models accuracy in terms of RMSE after each training epoch is complete. This accuracy is the average of the RMSE for each patch that constitutes the test set defined above. The orange dots represents the interpolation using the GAN and the blue line corresponds to the average RMSE of the patches interpolated using the IDW.

epochs, meaning that our approach improved its best performance within the 10 epochs interval until the 39th epoch. We highlight the stability of our approach by noticing the amount of epoch that performed better than the IDW technique and in some epochs to errors 3 meters below the IDW.

4.4.1.2 Configuration 2 - 100 observed points

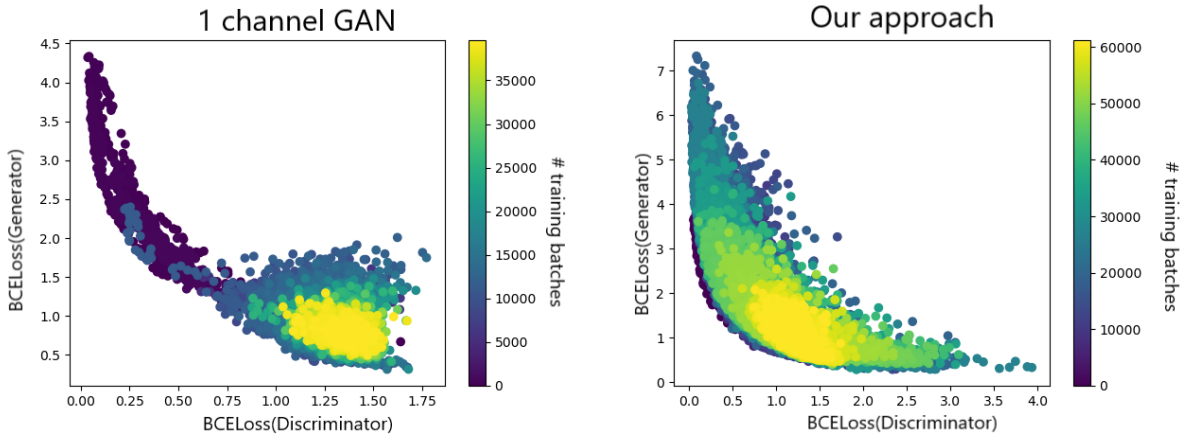

Figure 4.4: Scatter plots representing the adversarial game between the Discriminator and the Generator during the whole training process, where the focus is in the adversarial game evolution and its final state.

Figure 4.4 shows at the left plot we can observe that the majority of the discriminator losses is in the]0, 1.5] interval and most of the generator losses is in the]0, 4.5] interval. In the right plot the core of discriminator losses vary between]0, 3] and the mass of generator losses fluctuate between]0, 7]. As the number of batches fed to the network increases we can observe two small cluster on both plots. The

cluster on the left plot is restricted to a discriminator loss between]1, 1.5] and a generator loss of]0.5, 1.5]. The cluster on the right plot is restricted to a discriminator loss between]0.7, 1.7] and a generator loss of]0.5, 2]. Comparing the two clusters we can observe that they are very similar in terms of size meaning that the equilibrium in the adversarial game for both approaches was achieved.

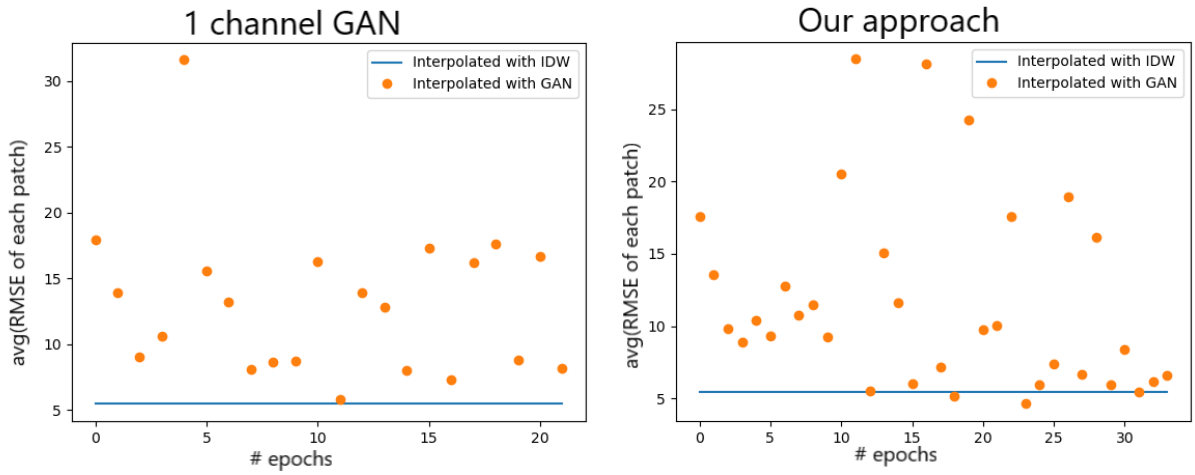

Figure 4.5: Variation of the models accuracy in terms of RMSE after each training epoch is complete. This accuracy is the average of the RMSE for each patch that constitutes the test set defined above. The orange dots represents the interpolation using the GAN and the blue line corresponds to the average RMSE of the patches interpolated using the IDW.

Figure 4.5 presents:

- The left plot had a maximum accuracy of above 30 meters (5th epoch) while the right plot correspondent to our hybrid approach peaked an error below 30 meters (12th epoch);
- In terms of minimum accuracy the left plot performed at a value above but close to the IDW (12th epoch) while our approach, represented on the right, performed a minimum accuracy below 5 meters (24th epoch);
- The training process lasted for 22 epochs for the one channel GAN, meaning that the network did not improved since the 12th epoch. Looking at our approach, the training process lasted for 34 epochs, meaning that our approach improved its bets performance with the 10 epochs interval until the 24th epoch where the accuracy dropped to values below 5 meters per patch.

4.4.1.3 Configuration 3 - 144 observed points

Figure 4.6 shows two scatter plots, the left plot we can observe that the majority of the discriminator losses is in the]0, 1.75] interval and most of the generator losses is in the]0, 4] interval. In the right plot the core of discriminator losses vary between]0, 3] and the mass of generator losses fluctuate between

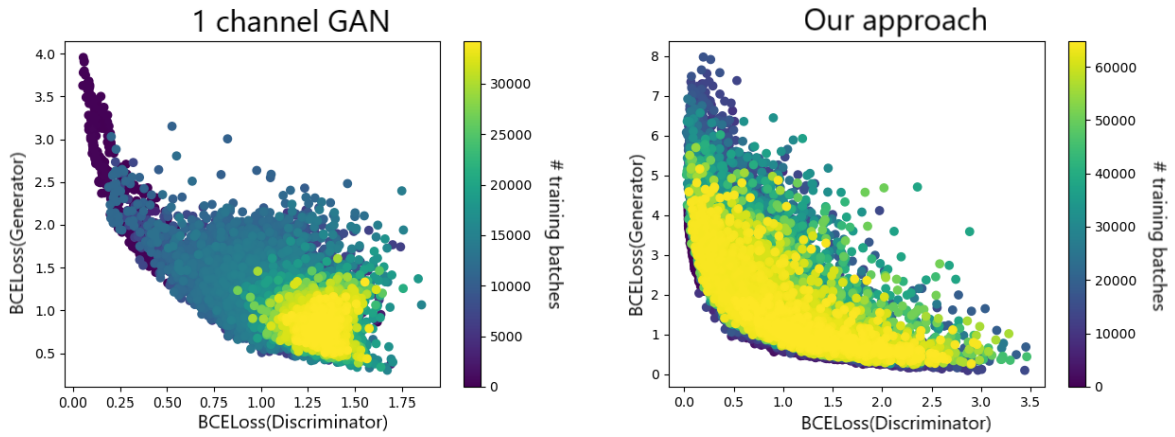

Figure 4.6: Scatter plots representing the adversarial game between the Discriminator and the Generator during the whole training process..

]0, 8]. As the number of batches fed to the network increases we can observe a small cluster on the left plot and a bigger cluster in the right plot. The cluster on the left plot is restricted to a discriminator loss between]1, 1.5] and a generator loss of]0.5, 1.5]. The cluster on the right plot is restricted to a discriminator loss between]0, 2.5] and a generator loss of]0.5, 4.5]. Comparing the two clusters we can observe that the left cluster is much smaller, meaning that the adversarial game between the two sub-models achieved an equilibrium, while the adversarial game played between the two sub-models of our approach have not achieved a cluster so restricted like the left plot, we can state that a wider equilibrium in the right plot was achieved by focusing on the evolution of the adversarial game from older batches to the more recent ones.

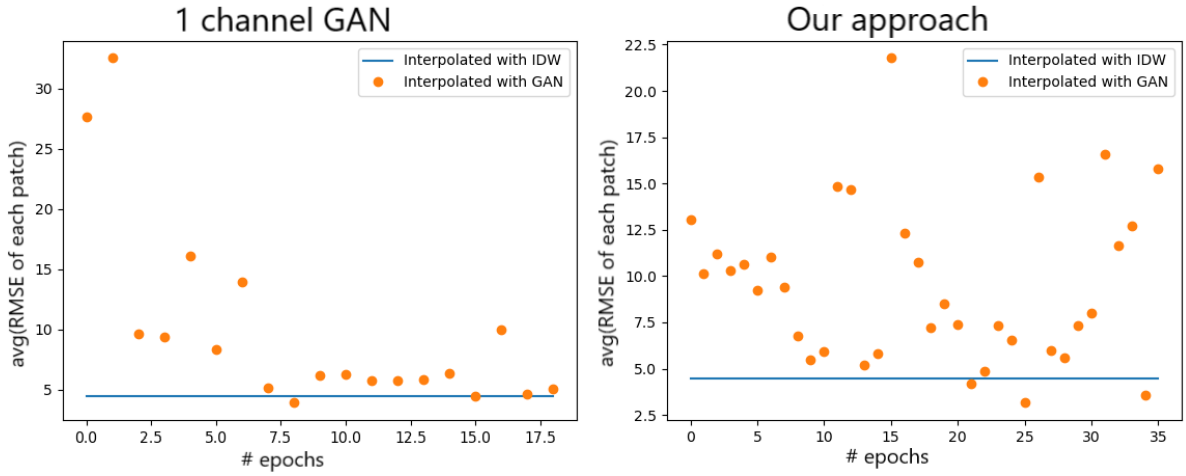

Figure 4.7: Variation of the models accuracy in terms of RMSE after each training epoch is complete. This accuracy is the average of the RMSE for each patch that constitutes the test set defined above. The orange dots represents the interpolation using the GAN and the blue line corresponds to the average RMSE of the patches interpolated using the IDW.

Comparing the two plots on Figure 4.7 we can observe that:

- The left plot had a maximum accuracy of above 30 meters (2nd epoch) while the right plot correspondent to our hybrid approach peaked an error below 22.5 meters (16th epoch);

- In terms of minimum accuracy the left plot performed at a value close to the IDW, with an error of $\approx$ 4.5 meters (9th epoch) while our approach represented on the right performed a minimum accuracy below 3 meters (26th epoch);

- The training process lasted for 19 epochs for the one channel GAN, meaning that the network did not improved since the 9th epoch despite coming close in the 16th, 18th and 19th epochs. Looking at our approach, the training process lasted for 36 epochs, meaning that our approach improved its bets performance with the 10 epochs interval until the 26th epoch where the accuracy dropped to values below 3 meters per patch that it no longer had the ability to drop below that value.

This section provides us the training behavior of each network, the 1 channel and the 2 channel input GAN, for a specific sampling configuration context. The main results of this section is the behavior of our approach for the 49 observed points and for the 144 observed points sampling configuration, where the scatter plots show a wider equilibrium in the adversarial game, meaning a more generalized state that the 1 channel GAN in every sampling configuration and our hybrid approach in the 100 observed points configuration, since they achieved more restricted clusters. This two networks not only demonstrated a more generalized equilibrium but a lower error compared to the error of the interpolation using the IDW and to the 1 channel GAN. Regarding the error results, the network relative to the 49 observed points configuration presented us with very promising results, since the lowest error produce for this network is close to 3 meters, in a configuration where the IDW performance is close to 7.5 meters. For 100 observed points configuration our network performed better than the IDW and the 1 channel GAN but with a small gap when comparing the errors. For the remnant configuration we observe that our approach produced a lower error than the IDW and the other GAN with a considerable difference of 2 meters. In conclusion, we can state that our hybrid approach performed better than the IDW and the simpler GAN, despite the training process lasted more when compared to the 1 channel GAN. Despite being better there are some configurations where our approach achieved promising results (49 and 144 observed points configuration) and where its behaviour and performance did not stand out from the simpler model (100 observed points configuration). In fact when our hybrid approach network relies on the configuration with fewer observed points, it produces results close to the results produced by the same approach for the configuration with triple the observed points and outperforms by almost 2 meters the same approach for the configuration with double the observed points, making this network the most promising and efficient since it has the ability of doing more or the same with much less resources.

4.4.2 Validation of the trained generator

To validate the generator ability to interpolate images and its generalization power we apply the trained generator to the rasters of each Portuguese island and to the raster of mainland Portugal. In the following table 4.1 we can see the average of the errors, MAE and RMSE, associated to each 32x32 patch composed uniquely by values with height above the sea level of the different regions we studied. These results were produced by the techniques we proposed to study: IDW, 1 channel GAN and our hybrid approach, 2 channel GAN.

After analysing the obtained results we can draw several conclusions:

- Both networks overcame the traditional IDW technique, especially in the islands where the networks surpassed the IDW i.e. Madeira island;
- When comparing the two GANs, the 1 channel GAN only fully overcame (in terms of MAE and RMSE) our hybrid approach in the Porto Santo island for the 49 and 100 observed points sampling configurations. Having this, we can claim that our hybrid approach surpassed the more simple GAN showing a higher ability to adapt to different contexts;
- Focusing on the results of our hybrid approach we highlight the performance of the network for the 149 sampling configuration where it was never surpassed by the simpler GAN. Looking at the 49 and 100 sampling configurations we can observe that the 49 configuration overcame the 100 configuration for every region and for the MAE and RMSE. Despite being an odd result, the ability of the network produce more with less is very positive and can save a lot of resources when transposed to the real world.

These observations portray a positive result since our approach surpassed the IDW performance and the 1 channel GAN approach. Besides showing the capability to overcome traditional and modern techniques our approach also demonstrated a higher degree of generalization when compared to the simpler GAN, as we can observe in the performance of our model on the different contexts from the one it was trained, the Portuguese islands.

Regions	Sampling Configuration	IDW		1 channel input GAN		2 channel input GAN	
		avg(MAE)	avg(RMSE)	avg(MAE)	avg(RMSE)	avg(MAE)	avg(RMSE)
Aoŗes - Corvo (15 patches)	49	15.111	18.308	**5.591**	9.060	5.770	**7.790**
	100	10.020	12.313	**5.646**	**7.479**	5.862	7.863
	144	7.760	9.598	4.849	7.716	**4.203**	**6.074**
Aoŗes - Faial (226 patches)	49	5.594	7.022	3.316	4.844	**3.222**	**4.193**
	100	3.837	4.901	4.357	5.389	**3.696**	**4.798**
	144	3.077	3.953	2.972	4.367	**2.397**	**3.289**
Aoŗes - Flores (183 patches)	49	11.176	13.886	5.241	7.710	**4.775**	**6.340**
	100	7.765	9.742	5.701	7.299	**5.240**	**6.903**
	144	6.275	7.879	4.334	6.491	**3.617**	**5.049**
Aoŗes - Graciosa (72 patches)	49	6.150	7.981	2.964	4.374	**2.946**	**3.972**
	100	4.253	5.597	**2.699**	3.661	2.765	**3.647**
	144	3.426	4.520	2.385	3.625	**2.032**	**2.821**
Aoŗes - Pico (607 patches)	49	4.399	5.624	3.958	5.681	**3.787**	**4.832**
	100	3.059	3.970	5.718	6.910	**4.674**	**6.055**
	144	2.443	3.186	3.572	5.037	**2.839**	**3.850**
Aoŗes - São Jorge (285 patches)	49	8.946	11.418	**4.815**	7.223	4.910	**6.467**
	100	6.122	7.910	5.880	7.419	**5.260**	**6.943**
	144	4.869	6.292	4.315	6.684	**3.518**	**4.940**
Aoŗes - Santa Maria (114 patches)	49	6.241	7.857	3.150	4.700	**2.747**	**3.675**
	100	4.593	5.814	3.152	4.122	**2.869**	**3.814**
	144	3.832	4.832	2.564	3.836	**2.143**	**3.002**
Aoŗes - São Miguel (1037 patches)	49	7.412	9.393	3.954	5.760	**3.532**	**4.649**
	100	5.339	6.804	4.510	5.712	**3.929**	**5.138**
	144	4.418	5.620	3.295	4.815	**2.704**	**3.727**
Aoŗes - Terceira (556 patches)	49	4.503	5.764	2.973	4.236	**2.628**	**3.413**
	100	3.199	4.140	3.911	4.762	**3.102**	**4.024**
	144	2.631	3.416	2.490	3.522	**2.042**	**2.773**
Madeira - Madeira (1048 patches)	49	18.724	22.931	8.208	12.023	**7.468**	**9.875**
	100	12.776	15.745	9.163	11.813	**8.352**	**10.936**
	144	10.228	12.625	7.138	10.488	**5.715**	**7.952**
Madeira - Porto Santo (39 patches)	49	7.745	9.860	**2.906**	**4.181**	3.528	4.681
	100	5.181	6.606	**2.715**	**3.719**	2.885	3.739
	144	4.129	5.286	2.557	3.873	**2.160**	**2.921**
Mainland Portugal (135577 patches)	49	4.940	6.092	2.981	4.206	**2.428**	**3.122**
	100	3.551	4.388	4.012	4.897	**3.009**	**3.918**
	144	2.944	3.630	2.438	3.353	**2.002**	**2.723**

Table 4.1: Obtained results for the IDW, 1 channel GAN and our hybrid approach considering only above the sea level 32x32 patches for each region.

5 Conclusions and Future Work

Spatial interpolation problem is very present in our society and can be applied to countless areas from earth observation to social sciences and influencing many real world variables i. e., the real estate values or the price of a commodity. Since the spatial interpolation problem has the potential to influence our world, the most accurate approach to solve the problem has to be applied. We divided the available techniques into traditional and modern. The traditional techniques served us for many years since there was no alternative. With the development of modern techniques based on image processing neural networks we can now overcome the traditional approaches but face a problem the traditional techniques never faced: the generalization problem. This new problem is limited to neural networks since they are data dependent and when trained into a specific context it can underperform when applied into different contexts from the ones it was trained. Having this challenges and problems in mind we proposed and studied an hybrid approach set to overcome the traditional and modern techniques in accuracy and to surpass the modern ones in terms of generalization ability. To capture the best of traditional approaches and the best of modern approaches we combined the sampled patch with the correspondent patch interpolated with the inverse weighting into a two channel input that is fed to a GAN. In this dissertation, we reported on experiments with the proposed hybrid approach technique and compare the results with a GAN inspired in the work of Zhu et al. (2019) and with the IDW technique. These experiments were applied to DEMs of mainland Portugal and the main islands of the Portuguese autonomous regions. The obtained results confirmed that the proposed procedure outperforms the traditional and modern approaches in terms of accuracy and ability to generalize when compared to the modern approach.

5.1 Overview on the Contributions

The most important contributions of my M.Sc. thesis are as follows:

- **An new alternative to solve the spatial interpolation problem** Inspired in the work of Zhu et al. (2019). We combined a GAN with the IDW method in a two channel input fed to the GAN. This two channel input consists in a sampled patch and the correspondent interpolate patch, in our case interpolated using the IDW technique. This alternative proved to be a valid choice to solve the

spatial interpolation problem, since it overperformed the IDW and the 1 channel GAN approaches in interpolating the height of the land points of the Portuguese islands and mainland.

- **Generalization** Compared with the GAN like the one Zhu et al. (2019) proposed and studied, our hybrid approach proved to better tackling the generalization problem. The generalization problem applies to neural networks since they are dependent of data to learn a specific data distribution and capture spatial features. This problem is the next obstacle spatial interpolation techniques need to overcome and we think our approach is a valid and simple solutions to the generalization problem, since our network produced a wider and more general equilibrium, which we can observe in the training scatter plots for the 49 and 144 observed points configuration and in the performance our approach achieve when transposed to the Portuguese islands where it overcame the 1 channel GAN proving a higher degree of resisting to different contexts from the ones it was trained.

- **Experiments in digital elevation models** The DEMs relative to the land are very complete and interpolation problems do not apply to the elevation of our planet above sea level. Regarding the oceans and lies below the surface is a problem worth our attention since the oceans topography and oceanography can help us comprehend better ocean circulation and climate change. Despite we applied our technique to a well studied field we believe its impact in the same field but below the oceans surface will have significant and meaningful results.

5.2 Future Work

For future work it would be interesting to continue improving the spatial interpolation methodology. We believe that our approach can still be improved and produce better results. One improvement worth studying is to change the technique used to interpolate the patch in the 2 channel input. In or approach we used the IDW technique to interpolate th patch that we combined with the sampled patch to form the input. Since nowadays approaches like the one proposed by Zhu et al. (2019) outperforms the IDW technique we think that replacing the IDW interpolated patch by the GAN interpolated patch is an approach worth studying, despite it involves previously training a whole network to train the network of our approach.

Another possible direction we could take and study is the inclusion of batch normalization layers to our model. This layers regularizes the model and have a positive impact in the model generalization ability Ioffe and Szegedy (2015). With all this useful advantages to our model we believe they will have a very positive impact in our network in terms of performance and it is worth the studying.

Finally, there are also plans to apply our hybrid approach to other contexts and problems. Generally

spatial interpolation problems are more related to earth sciences, but we believe such solutions can have an important role in socio-economic spatial interpolation problems.

References

Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition.

Deng, X., Zhu, Y., Tian, Y., and Newsam, S. (2019). Generalizing deep models for overhead image segmentation through getis-ord gi* pooling.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Greenberg, J., Rueda, C., Hestir, E., Santos, M., and Ustin, S. (2011). Least cost distance analysis for spatial interpolation. *Computers Geosciences*, 37:272–276.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Ma, Z. (2019). *Geostatistical Estimation Methods: Kriging*, pages 373–401.

Marcos, D., Volpi, M., Kellenberger, B., and Tuia, D. (2018). Land cover mapping at very high resolution with rotation equivariant cnns: Towards small yet accurate models. *ISPRS Journal of Photogrammetry and Remote Sensing*.

Mitas, L. and Mitasova, H. (2005). *Geographic Information Systems: Principles, Techniques, Management and Applications, 2nd Edition, Part 2, Chapter 34*, volume 1. Spatial Interpolation. In: Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W., Eds.

Myers, D. (1994). Spatial interpolation: An overview. *Geoderma*, 62:17–28.

Rigol, J., Jarvis, C., and Stuart, N. (2001). Artificial neural networks as a spatial interpolation. *International Journal of Geographical Information Science*, 15:323–343.

Stachelek, J. and Madden, C. (2015). Application of inverse path distance weighting for high-density spatial mapping of coastal water quality patterns. *International Journal of Geographical Information Science*.

Tobler, W. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, pages 234–240.

Zhu, D., Cheng, X., Zhang, F., Yao, X., Gao, Y., and Liu, Y. (2019). Spatial interpolation using conditional generative adversarial neural networks. *International Journal of Geographical Information Science*, pages 1–24.