ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MASTER DEGREE IN ARTIFICIAL INTELLIGENCE

MASTER THESIS IN DEEP LEARNING

DEEP LEARNING MODELS FOR DOWNSCALING OF METEREOLOGICAL VARIABLES

Supervisor:
Prof. Andrea ASPERTI

Co-supervisor:
Dr. Fabio MERIZZI

Candidate:
Stefano COLAMONACO

Session 1st
Academic year 2023/2024

Ai miei genitori e alle mie sorelline.

Abstract

Reanalysis datasets play an important role in meteorological and climate research, offering a consistent and long-term record of atmospheric conditions by assimilating past observations with modern forecast models. These datasets are of great utility in various applications, including weather forecasting, climate change research, renewable energy prediction, resource management, air quality risk assessment, and the forecasting of rare climatic events. Among the most prominent reanalysis datasets is the Copernicus Regional Reanalysis for Europe (CERRA), which stands out due to its high-resolution coverage of the European domain. CERRA has demonstrated significant utility across multiple climate-related tasks, providing detailed insights that are essential for precise and localized studies. Despite its advantages, the availability of CERRA lags two years behind the current date, primarily due to the intensive computational demands and the complexities involved in acquiring the necessary external data. To address this temporal gap, this thesis proposes a novel method employing several deep neural models to approximate CERRA downscaling in a data-driven manner without the need for additional external information other than ERA5. By leveraging the lower resolution ERA5 dataset, this research frames the problem as a super-resolution task. The study focuses on downscaling wind speed data over Italy, utilising a model trained on existing and freely available data. The results are encouraging, as the model produces outputs closely resembling the original CERRA data, with validation against in-situ observations confirming its accuracy in approximating ground measurements. This innovative approach not only demonstrates the potential of deep neural models in overcoming the computational and data acquisition constraints associated with high-resolution reanalysis datasets but also offers a viable solution to improve the timeliness and accessibility of such data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of meteorology is inherently complex, tasked with understanding and predicting the intricate dynamics of the Earth's atmosphere. A crucial aspect of this endeavour lies in the ability to accurately model weather variables at various spatial and temporal scales. High-resolution weather data offers the most detailed representation of atmospheric conditions, but its computational demands and storage requirements can be really prohibitive. This presents a significant challenge, particularly for applications requiring real-time weather forecasts or long-term climate simulations [52, 74].

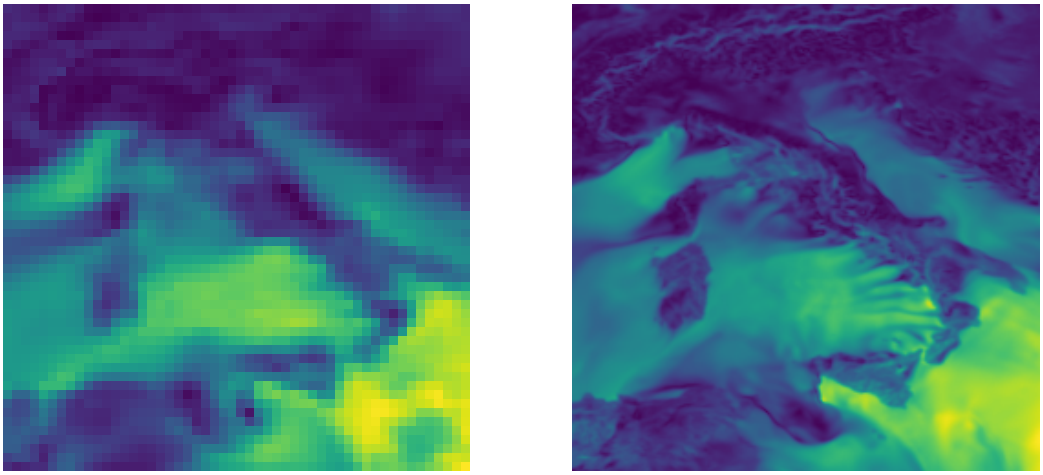
Downscaling techniques serve to bridge this gap by transforming coarse-resolution global climate models (GCMs) into high-resolution data that is suitable for regional and local applications. Traditional downscaling methods, such as statistical downscaling, have served as valuable tools for many years. However, with the recent advancements in Artificial Intelligence, deep learning models have emerged as a promising alternative for Downscaling weather variables.

Deep Learning, a subfield of Machine Learning, employs the power of artificial neural networks with intricate architectures to learn complex relationships from large amounts of data. These models exhibit exceptional capabilities in pattern recognition and feature extraction, rendering them well-suited for tasks involving nonlinear and multivariate relationships, which are often prevalent in weather data [11].

The relationship between downscaling models typically used for reanalysis and Deep Learning models lies in the super-resolution task, with both approaches aiming to bridge the gap between coarse and high-resolution data [66, 67]. Conventional downscaling methodologies frequently depend on statistical relationships or physical principles to infer the high-resolution information from the coarse data. In contrast, Deep Learning models are capable of learning these relationships directly from large datasets of historical observations, thereby potentially capturing more complex and nuanced patterns. In essence, deep learning models

can be viewed as a form of data-driven super-resolution for weather data. Super-resolution, a technique commonly used in image processing, aims to reconstruct a high-resolution image from a low-resolution version. Deep learning models can be adapted to achieve a similar result, beginning with weather data in a gridded image-like format and resulting in a reconstruction at a finer resolution [30, 75].

This thesis is dedicated to the application of several deep learning models in the downscaling of meteorological variables (with application to wind speed), with a direct comparison made with one of the most advanced reanalysis datasets of recent years: the Copernicus European Regional Reanalysis (CERRA) [55]. To achieve this, a subset of the ERA5 [25] dataset was exploited, encompassing hourly data of the wind speed variable for Western Europe over a period of 10 years, from 2010 to 2019, with a temporal resolution of 3 hours. Two years, 2009 and 2020, were selected as the test set. This choice is particularly advantageous because CERRA is derived using ERA5 as a baseline, thereby establishing a robust connection between the two datasets and minimising their differences.



ERA5 CERRA

FIGURE 1.1: This image presents a comparison of the wind speed over Italy between the ERA5 and CERRA datasets.

The proposed approaches were not only able to demonstrate that contemporary deep learning models are capable of replicating the functionality of numerical models that are still employed in downscaling, but some of them even managed to emulate the accuracy of CERRA with a minimal amount of error on the observed variable. The study is further enhanced by a validation of the reference datasets and the results obtained through a comparison with the in-situ data from the IGRA-V2 dataset [19]. This comparison allows for an estimation of the extent

to which a reanalysis may deviate from the actual observations of meteorological data.

This thesis is structured into six chapters, each addressing a distinct aspect of the study and providing a comprehensive exploration of deep learning models for the downscaling of meteorological variables:

- Chapter 1: Introduction - This chapter presents the background, motivation, and objectives of the research, providing a foundation for understanding the importance of downscaling in meteorology and the potential of deep learning models.

- Chapter 2: Related Works - This chapter presents a review of existing literature and previous research related to downscaling methods through deep learning models and the validation of reanalysis data. It identifies the gaps and opportunities that this thesis aims to address.

- Chapter 3: Background - This chapter provides an overview of downscaling techniques, including statistical and dynamical downscaling, as well as an introduction to super-resolution and its connection to the task of downscaling meteorological data. It concludes with a presentation of the dataset used in this study.

- Chapter 4: Deep Learning Models for SR Downscaling - This chapter examines the various state-of-the-art deep learning models that can be employed for super-resolution downscaling. These include dense neural networks, convolutional neural networks, recurrent neural networks, generative adversarial networks, diffusion models, and transformers.

- Chapter 5: Experiments - This chapter outlines the experimental setup, including a description of the datasets, the preprocessing steps, and the evaluation framework. It also presents and analyses the results of the experiments conducted to assess the performance of the proposed models. Finally, it presents a validation for both reanalysis (ERA5 and CERRA) and the results of the models, exploiting data from in-situ observations.

- Chapter 6: Conclusions - This chapter presents a summary of the key findings of the research, discusses their implications, and suggests potential directions for future work in the field of meteorological downscaling using deep learning.

By following this structure, the thesis aims to provide a comprehensive understanding of the potential and challenges of using deep neural models for downscaling meteorological variables, contributing to the advancement of high-resolution weather forecasting and climate modeling.

The results of this study have led to a paper [43] being published in the international journal Neural Computing & Applications.

Chapter 2

Related Works

The necessity for high-resolution weather data at a regional and local scale has driven the development of robust downscaling techniques. Statistical downscaling represents a well-established methodology in this domain, effectively bridging the resolution gap between global climate models (GCMs) and fine-grained weather observations. Innovative efforts in statistical downscaling employed foundational statistical methods such as linear regression and ridge regression [33, 71]. These techniques enabled the identification of linear associations between large-scale predictors (e.g., pressure, temperature fields) and local weather variables of interest (e.g., precipitation, wind speed).

A number of machine learning techniques, such as k-nearest neighbours [22], random forests [50, 24] and support vector machines (SVM) [51], have been applied in the context of downscaling to generate various surface variables, showing that these techniques are particularly suited to the task.

More recently, deep learning approaches based on convolutional neural networks (CNNs) have emerged as a promising downscaling technique. CNNs have the ability to automatically infer spatial features that encode predictive information from the predictor fields. In particular, the most popular application of CNNs and other deep learning architectures for downscaling follows the super-resolution (SR) approach [67, 66, 5], inspired by the homonymous field of computer vision. This approach learns to generate high-resolution fields from their low-resolution counterparts. The advent of new deep learning models has led to an increased focus on their use for downscaling. Indeed, the present study has identified implementations that exploit Long Short Term Memory (LSTM) [44], U-Net [57] and Generative Adversarial Networks (GAN) [10].

In recent years, this area of research has become increasingly popular, with a growing corpus of innovative implementations documented in the literature. In [30] the authors investigated the application of three super-resolution deep learning frameworks for downscaling daily precipitation forecasts in Southeast China

from a starting resolution of approximately 50 kilometers to 25 and 12.5 kilometers. These frameworks included the Super-Resolution Convolutional Neural Network (SRCNN), Super-Resolution Generative Adversarial Networks (SRGAN), and Enhanced Deep Residual Networks for Super-Resolution (EDSR). For comparison, the study also performs Bias Correction Spatial Disaggregation (BCSD) as a traditional SD method under the same framework. This traditional method is used to refine the spatial resolution of daily precipitation forecasts, but it does not employ deep learning techniques. The results show that the SR models outperform the traditional BCSD method in capturing local details and improving the spatial resolution of precipitation forecasts.

In [46] the authors present ClimateLearn, a comprehensive machine learning (ML) library designed to facilitate the application of ML to challenging weather and climate modelling tasks. The library is comprised of four key components: tasks, datasets, models, and evaluations, which collectively enable the benchmarking of ML models for weather and climate forecasting, downscaling, and projection. The authors focus on those tasks utilising a range of datasets, including ERA5 and CMIP6. They evaluate the performance of various deep learning methods, including ResNet, U-Net, and ViT, and compare them to simple baselines such as climatology and persistence. With regard to downscaling, two different settings were chosen. The first involved downscaling the 5.625° ERA5 to 2.8125° ERA5 with hourly intervals and at the global scale. The target variables were Z500, T850 and T2m. In the second setting, the authors consider downscaling from 2.8125° ERA5 to 0.75° PRISM over the continental United States at hourly intervals for the variable daily maximum T2m.

In the study proposed by Zhang and Li [75], the authors focus on the downscaling of wind speed via deep learning models, specifically utilising a novel method based on Bidirectional Gated Recurrent Units (BiGRU). The primary objective of this research is to develop a more accurate and efficient approach for predicting local offshore wind energy resources in China using large-scale climate models and the ERA-Interim reanalysis dataset as a reference. The authors apply the BiGRU downscaling method to a dataset comprising daily 10-m wind speed, sea level pressure, and 850 hPa WS from the ERA-Interim reanalysis dataset at a 1.5° lat/lon resolution. The daily 100-m WS from the ERA5 reanalysis dataset at a 0.25° lat/lon resolution is used as the output layer vectors. The performance of the BiGRU method was then compared to that of traditional methods such as PCA-SMLR and BPNN and it outperformed both of them in terms of root mean squared error (RMSE) and Pearson correlation coefficient (PCC).

It is important to note that, in addition to downscaling, deep learning models, such as those employed in this study, can be utilised for a variety of purposes and tasks related to meteorological variables. For instance, in [3], the authors employ diffusion models, with a particular focus on Generative Ensemble Diffusion (GED), for the purpose of precipitation nowcasting. This is the forecasting of rain indicators on a short-term meteorological scale, typically between two and six hours. Furthermore, the authors demonstrate that their results outperform those of recent deep learning models in terms of overall performance.

The present thesis also addresses the issue of result validation, a topic that frequently arises in the literature concerning reanalysis and downscaled data.

In [49] the authors evaluate the performance of the Copernicus European Regional Reanalysis (CERRA) dataset as a proxy for ground-based agrometeorological data in assessing daily reference evapotranspiration (ETO) at the regional scale. To achieve this goal, the authors employ a comprehensive methodology that involves the use of 38 automatic weather stations (AWSs) in Sicily, Italy, which provide ground-based weather data from 2003 to 2022. The authors initially compute the daily reference evapotranspiration utilising the FAO Penman-Monteith equation, which is a widely accepted and accurate formulation for ETO estimation. This is then compared with the corresponding values obtained from the CERRA dataset. The results of the study demonstrate that CERRA data exhibit excellent performance in estimating the considered variables.

In the study conducted in [45], the authors perform a validation of various modelling systems of global and regional reanalyses, including ERA5, ERA5-Land, ERA5-Crocus, CERRA-Land, UERRA MESCAN-SURFEX and MTMSI. The in-situ validation data included observational references (gridded observational datasets and satellite observations) across the European Alps from 1950 to 2020 for a set of weather variables, including snow cover, temperature, and precipitation. The results of the study revealed that no single modelling strategy outperforms all others within the experimental sample but reanalyses (and datasets based upon observation sources) hold the potential to partially fulfill the gap of in situ observations.

In [9], the authors conduct a comprehensive intercomparison and validation of high-resolution surface air temperature reanalysis fields over Italy. The analysis focuses on five reanalysis datasets: ERA5, ERA5-Land, MERIDA, CERRA, and VHR-REA_IT. The authors employ a methodology that involves reconstructing climate normals and anomalies from a large network of homogenized station records, which are then used to evaluate the performance of the reanalysis

datasets. Once the monthly observation-derived gridded series have been reconstructed and the daily values retrieved, they are then used to calculate the bias and mean absolute error (MAE) of the reanalysis datasets in comparison to the observed data. The results demonstrate that the reanalysis datasets exhibit varying degrees of bias and MAE, with some datasets performing better than others in specific regions and seasons. For instance, ERA5 exhibits a bias of -0.5 to -1°C in the majority of regions, whereas ERA5-Land displays a more pronounced cold bias, particularly in the Alpine region. MERIDA and CERRA exhibit more intricate patterns, with cold biases concentrated in specific areas but very good results. The study also reveals that the spatial distribution of the MAE is closely linked to the degree of agreement between the observed and reanalysis anomalies.

It can be concluded that in the majority of validations presented in the literature, CERRA consistently demonstrated superior performance in the analysed variables, with only a few exceptions. Therefore, the ability to reproduce its characteristics would represent a valuable reanalysis and thus a precious resource.

Chapter 3

Background

3.1 Downscaling

The field of climate change science is characterised by a high degree of complexity, necessitating a comprehensive grasp of the knowledge and understanding required to analyse and interpret the results produced by the models and methods employed to simulate future climates. Despite these complexities, information generated by research in this field is frequently used to assess potential impacts on bio-physical systems, built environments, and people. Information on climate change is available in a variety of formats, including maps, time series, and graphical summaries, and is derived from a range of sources, including output from different climate models simulating change to the global climate [20]. Being able to make the most of these sources of data is crucial, as well as to be able to infer additional information beyond that which is available through statistical methods that exploit the patterns contained in past observations. To illustrate this with a hypothetical example, consider a wind farm project. This harnesses the power of wind to generate clean electricity. However, before construction can commence, there is one crucial factor to consider: the wind itself. It is of the utmost importance to ascertain the precise quantity of wind that passes through the area, including the wind speed and direction. This information is vital for a number of reasons, including optimising power production, strategic placement and ensuring the financial viability of a project.

A significant proportion of applications have a local or regional focus, necessitating the use of fine spatial resolution in their climate information, in addition to information about the larger-scale climate signal. Therefore, if there is any regional-scale predictability in the climate signal, this is highly relevant and should be included in those applications. However, global climate models (GCMs) typically operate on spatial resolutions of around 100 to 250 km. A number of techniques exist to bridge the scale gap between the output of GCMs and

the scale of interest of most applications. These techniques vary considerably in terms of complexity and sophistication, but are all collectively referred to as *downscaling* methods. This term encapsulates the process of translating a coarse scale GCM output to finer-scaled information on weather variables [20]. There are two main approaches to downscaling climate model outputs: Statistical and Dynamical downscaling.

3.1.1 Statistical Downscaling

Statistical downscaling establishes a statistical relationship between large-scale atmospheric variables simulated by GCMs (predictors) and local-scale climate variables of interest (predictands) using historical observations. The aforementioned relationship is then applied to project future local-scale climate changes based on the projected changes in large-scale variables from the GCMs.

The core concept of statistical downscaling can be represented mathematically as a function relating the large-scale predictors (X), observed local-scale predictands (Y), and potentially some random error term (ε):

$$\mathrm{Y} = f(\mathrm{X}, \varepsilon) \tag{3.1}$$

The goal of statistical downscaling is to estimate the function f using historical observations of both X and Y. Once estimated, this function can be used to project future local-scale climate changes (Y_future) based on the projected changes in the large-scale variables (X_future) from the GCMs:

$$\mathrm{Y\_future} = f(\mathrm{X\_future}, \varepsilon\_future) \tag{3.2}$$

It is essential to recognise that this is a simplified representation, and that more complex formulations may be employed depending on the specific downscaling method and the number of variables involved.

3.1.2 Dynamical Downscaling

In the context of dynamical downscaling, a higher-resolution climate model is employed. These models are frequently referred to as regional climate models (RCMs). RCMs utilise lower-resolution climate models (in the majority of cases, GCMs) as boundary conditions and physical principles to reproduce local climate. While offering a more physically-based approach, dynamical downscaling is associated with a significant computational cost compared to statistical methods.

RCMs require substantially more processing power to simulate the complex dynamics of the atmosphere at high resolution. This translates to limitations in the number of simulations that can be performed and the length of the simulated time periods.

3.2 Super-Resolution

Although high-resolution (HR) digital cameras are available, many computer vision applications still require imagery with resolutions that surpass the capabilities of these cameras. Fields such as satellite imaging, target detection, medical imaging, and others demand higher resolution imagery to achieve precise and accurate results. To meet this strong demand for superior imagery, these applications often employ advanced image-processing techniques. One of the most promising techniques in digital imaging is Super-Resolution (SR) image reconstruction. The objective of this method is to generate high-resolution (HR) images by combining the partial information from several under-sampled low-resolution (LR) images of the same scene. During the reconstruction process, super-resolution techniques up-sample these low-resolution images and mitigate distortions such as noise and blur, resulting in a higher quality final image [4].

In contrast to other image enhancement techniques, which may simply sharpen or adjust the brightness and contrast of an image, super-resolution image reconstruction fundamentally improves the spatial resolution of under-sampled images. It not only enhances the overall quality by increasing resolution but also systematically filters out various distortions. This makes super-resolution an invaluable tool for applications that rely on detailed and accurate imagery.

The super-resolution process can be defined as an ill-posed inverse problem, where the objective is to invert an unknown degradation function applied to a high-resolution image [43]. Given a low-resolution image denoted by y and the corresponding high-resolution image denoted by x, the degradation process can be described as follows:

$$y = \Theta(x, \rho_n) \tag{3.3}$$

The notation Θ represents the degradation function, while ρ_n denotes the degradation parameters, which may include factors such as the scaling factor and noise. In a real-world scenario, only y is available, while no information about the degradation process or the degradation parameters ρ exists. Super-resolution seeks to nullify the degradation effect and recovers an approximation $\hat{x}$ of the

ground-truth image x as:

$$\hat{x} = \Theta^{-1}(y, \rho_r) \tag{3.4}$$

where ρ_r are the parameters for the function Θ^{-1}. The degradation process is unknown and can be quite complex. It can be affected by several factors, such as noise, compression, blur and other artefacts. Therefore, most research works [2, 39, 6] have opted for the following degradation model over that of Equation 3.3:

$$y = (x \otimes k) \downarrow_s + n \tag{3.5}$$

where k is the blurring kernel and $(x \otimes k)$ is the convolution operation between the HR image and the blur kernel, $\downarrow_s$ is a downsampling operation with a scaling factor s.

3.2.1 Super-Resolution for Downscaling

While super-resolution and statistical downscaling are distinct in their methodologies and objectives, they share a common goal of refining data resolution, particularly in the context of weather datasets. As previously outlined in the preceding sections, this approach relies on establishing statistical relationships between large-scale atmospheric variables (such as temperature, pressure, and wind) and local-scale weather phenomena. This process is typically implemented via a sophisticated physical model, such as the HARMONIE-ALADIN model [7, 64]. These models require as input the low-resolution ground truth, in addition to the additional information necessary to compute the downscaling result. Consequently, if we represent the high-resolution image as x, the low-resolution images as y, the statistical downscaling model as HA and the additional information as ρ_{HA}, we can define the downscaling process as follows:

$$x = HA(y, \rho_{HA}) \tag{3.6}$$

It can be observed that this equation is closely related to Equation 3.4, which establishes a relationship between high-resolution and low-resolution images. This similarity effectively equates the task of super-resolution in image processing to the problem of statistical downscaling in meteorological studies. A comparison of the previous formulation with the one presented in section 3.1.1 also reveals this.

The primary distinction between the two problems lies in the inverse nature of their low-resolution/high-resolution relationship. In contrast to the super-resolution process described in Equation 3.5, where the low-resolution image is

derived from an unknown high-resolution counterpart, in statistical downscaling, the situation is reversed. Here, the high-resolution image is an enhanced version of the low-resolution image.

We can thus define the new super-resolution task as the problem of approximating the behaviour of the *HA* model. Since our aim is to follow a data-driven approach and not to use additional information for the downscaling process, we define our approximate model without the use of ρ_{HA}.

$$\hat{x} = \Phi(y)\,,\, \hat{x} \approx x \tag{3.7}$$

Where Φ is our super-resolution model and $\hat{x}$ is the approximation of the ground truth x generated by the statistical downscaling model.

3.2.2 Super-Resolution and Deep Learning

In recent years, deep learning has emerged as a revolutionary approach in the field of super-resolution. By leveraging the power of neural networks, deep learning-based methods have demonstrated remarkable improvements in generating high-resolution images from low-resolution inputs. These methods utilize large datasets to train deep neural networks to learn complex mappings between low and high-resolution image pairs, capturing intricate details and textures that traditional techniques may overlook.

Deep learning approaches to super-resolution encompass a variety of architectures, including convolutional neural networks (CNNs) [67, 66, 5], generative adversarial networks (GANs) [10], and recurrent neural networks (RNNs) [44]. Each of these architectures offers unique advantages in image reconstruction tasks. For instance, CNNs are adept at identifying spatial hierarchies and patterns within images, while GANs introduce a competitive learning process that enhances image realism. In contrast, recurrent neural networks (RNNs) are well-suited to the task of video super-resolution, as they are able to effectively handle sequential dependencies.

Metrics

In order to evaluate the quality of super-resolution methodologies, numerous metrics are employed in the literature. Some of these were utilised during the study conducted in this thesis, including the Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).

The Mean Squared Error (MSE) is a metric that is widely used in a variety of applications, including super-resolution. It is used to quantify the difference between the predicted values and the actual values by averaging the squares of the errors. In this context, the error is defined as the difference between each predicted value and its corresponding true value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.8}$$

Peak Signal-to-Noise Ratio (PSNR) [28] is a prominent metric employed in the field of image processing to assess the quality of reconstructed, compressed, or denoised images in comparison to the original image. It is defined as the ratio between the maximum possible power of a signal (represented by pixel values in an image) and the power of corrupting noise that affects its fidelity.

$$PSNR = 20 \cdot \log_{10}\left(\frac{1}{\sqrt{MSE}}\right) \tag{3.9}$$

The Structural Similarity Index (SSIM) [28] is an advanced metric used for measuring the similarity between two images. Developed to provide a more accurate and perceptually relevant assessment of image quality, SSIM differs from traditional metrics like MSE and PSNR by considering changes in structural information, luminance, and contrast. In contrast to MSE and PSNR, which focus on pixel-to-pixel differences, SSIM evaluates changes in structural patterns, thereby aligning itself more closely with human visual perception. SSIM can be expressed as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.10}$$

where the metric is computed between two corresponding patches represented by x and y, with μ_x and μ_y being the respective pixel average, σ_x^2 and σ_y^2 the variance and σ_{xy} the covariance. C_1 and C_2 are two custom parameters to o stabilize the division with weak denominator. The aforementioned metrics have been employed in a multitude of recent studies with objectives analogous to those of the present investigation [12, 34, 65].

The output data from a predictive model are typically normalised, necessitating a denormalisation step prior to comparison with ground truth. Once this has been completed, it is possible to employ other types of metrics to obtain a clear representation of the difference between different data, while retaining any meanings these data represent and obtaining a result that can be traced back to

something tangible. The study presented in this thesis includes a data validation step, which compares the results obtained from the models with real observations. This allows for the calculation of a physics-coherent measure of error. To achieve this, the MAE is employed. The MAE, or Mean Absolute Error, is another metric commonly used to assess prediction accuracy. It calculates the average magnitude of the errors between predicted and actual values, without considering their direction. In contrast to MSE, which squares the errors, MAE employs the absolute values of the errors. This renders MAE less susceptible to outliers in comparison to MSE, and it facilitates a more precise comprehension of the magnitude of the error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.11}$$

3.3 Datasets

In order to ascertain the viability of developing models capable of utilising large datasets containing meteorological data to facilitate the downscaling process, it is first essential to gain an understanding of the fundamental characteristics of the data.

In climatology, the term *reanalysis* refers to the process of re-estimating historical weather and climate conditions using a combination of observations and numerical weather prediction (NWP) models. This involves re-processing historical data using modern models and assimilation techniques to create a consistent and coherent dataset that can be used for research, forecasting, and other applications. The resulting datasets produced through this process typically include various atmospheric and surface variables such as temperature, humidity, wind speed, and precipitation. They are often created by combining historical observations with model simulations, using techniques such as data assimilation, in order to produce a consistent and coherent representation of the past climate.

Reanalysis data are of significant value for a number of reasons:

- they offer enhanced accuracy, providing more accurate and consistent representations of historical climate conditions compared to individual observations or model simulations alone;
- they offer long-term consistency, allowing the creation of long-term, consistent datasets that span multiple decades or even centuries, thus enabling more comprehensive studies of climate trends and variability;

- they can provide higher spatial and temporal resolution than individual observations, allowing for more detailed studies of local and regional climate phenomena.

In addition, the reanalysis data is employed in a number of ways, including:

- Climate modelling and prediction: The reanalysis data is used to initialise and validate climate models, which are essential for predicting future climate trends and variability;
- Climate change research: The reanalysis data is used to study past climate trends and variability, helping to understand the causes and impacts of climate change;
- Weather forecasting: The reanalysis data is used to improve weather forecasts, providing more accurate and consistent representations of historical weather patterns;
- Climate impact assessments: Reanalysis data are employed to evaluate the impact of climate change on various sectors, including agriculture, water resources, and ecosystems.

Another valuable source of information is the data collected by the monitoring stations in the region, which will be incorporated into the in-situ observation datasets. These datasets are fundamental to the study of climatology, providing direct measurements of various atmospheric, oceanic, and terrestrial parameters. In contrast to remote sensing techniques, which collect data from a distance (e.g., via satellites), in-situ observations are gathered directly at the location of interest using instruments such as weather stations, buoys, and radiosondes.

3.3.1 ERA5

ERA5 [25], produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) under the Copernicus Climate Change Service (C3S), is a state-of-the-art atmospheric reanalysis dataset. It represents a significant advancement in the field of reanalysis, offering a comprehensive and high-resolution depiction of the global climate system. As a result, it is an indispensable resource for climate research, weather forecasting, and numerous scientific and practical applications. ERA5 is characterized by its high temporal and spatial resolution. The dataset provides hourly data, enabling a comprehensive examination of diurnal cycles and short-term weather phenomena. This high temporal resolution is vitally

important for the study of events such as storms, heatwaves, and other transient atmospheric occurrences. Furthermore, the ERA5 dataset offers a horizontal resolution of approximately 31 km (0.28 degrees) and 137 vertical levels extending from the surface up to 0.01 hPa.

Another crucial aspect of the ERA5 dataset is its temporal scope, which extends from approximately 1940 to the present. This allows the dataset to be used in applications that require large amounts of data spread over time. Furthermore, since the collection of requisite data and their computation results in a production latency of approximately five days, the dataset remains consistently up-to-date.

ERA5 is a state-of-the-art global atmospheric reanalysis tool that provides a robust framework for understanding and analysing the Earth's climate system. It offers a comprehensive representation of atmospheric, land, and oceanic conditions spanning several decades. Over the years, ERA5 has emerged as a crucial asset for scientists from a variety of fields, with the dataset appearing in a vast number of studies. Its applications span several key areas, including:

- Analysis of Precipitation trends: ERA5 data is extensively used for tracking precipitation changes, as evidenced in multiple recent works [73, 35, 13, 58]. These works demonstrate the value of this dataset in monitoring and analysing alterations in precipitation patterns in the context of climate change.

- Investigation of Temperature Trends: The dataset is of significant value in the analysis of long-term temperature trends, with studies encompassing global temperatures [72, 41] and urban heat [38].

- Wind Analysis: Recent works demonstrates its application in studying wind patterns for renewable energy monitoring [48] and addressing climate changes [8].

- Extreme Events Study: ERA5's role is pivotal in advancing our understanding of extreme climate events, exploring storm surge [18] and heatwaves [1] and extreme precipitations [16, 70]:

- Additionally, ERA5 data finds important applications in other fields such as agriculture [52], water resources management [74, 63], and urban planning and infrastructure design [31].

In conclusion, ERA5 represents a significant contribution to the field of weather and climate studies, serving as a crucial reference point for ongoing scientific research.

3.3.2 CERRA

The Copernicus European Regional Reanalysis (CERRA) [55] dataset, produced under the Copernicus Climate Change Service (C3S), represents a significant advancement in the field of regional climate reanalysis. CERRA is designed to provide high-resolution, detailed, and accurate climate data for Europe, offering invaluable insights into the region's atmospheric conditions. This dataset is particularly relevant for climate research, weather forecasting, and various scientific and practical applications that require fine-scale climate information. CERRA, as well as ERA5 provides hourly data, enabling detailed analysis of diurnal cycles and short-term weather phenomena.

CERRA offers data at a fine 5.5 km horizontal resolution. This dataset achieves its detailed resolution by utilizing the global ERA5 reanalysis dataset, which provides both initial and boundary conditions. In addition to inputs from ERA5, CERRA's regional reanalysis incorporates higher-resolution observational data and physiographic datasets that describe surface characteristics. This reanalysis process is described in figure 3.1 and comprises three distinct stages:

1. The Global Reanalysis ERA5 will serve as the boundary condition;
2. The CERRA Regional Reanalysis;
3. A CERRA 2D Reanalysis for the near surface.

The number of arrows in figure indicate that the quantity of observational data incorporated into the reanalysis increases progressively from the global to the regional reanalysis.

The output of CERRA includes both forecasts and reanalyses. In the context of weather forecasting, the current state of the atmosphere and terrestrial and marine surfaces is analysed. These forecasts are generated using mathematical and physical numerical models that are based on this analysis. During the reanalysis phase, a weather forecasting model is initially used to estimate the atmospheric state at a specific time. This 'first guess' is then refined through a process known as 'data assimilation', which involves correcting the model based on observational data. This process ensures that the reanalysis provides an accurate historical record of weather conditions.

CERRA encompasses a comprehensive array of meteorological variables, including temperature, humidity, wind speed and direction, precipitation, and cloud cover. The data is available on a regular grid, while the temporal resolution differs from that of ERA5, with a distinction made between reanalysis and forecast

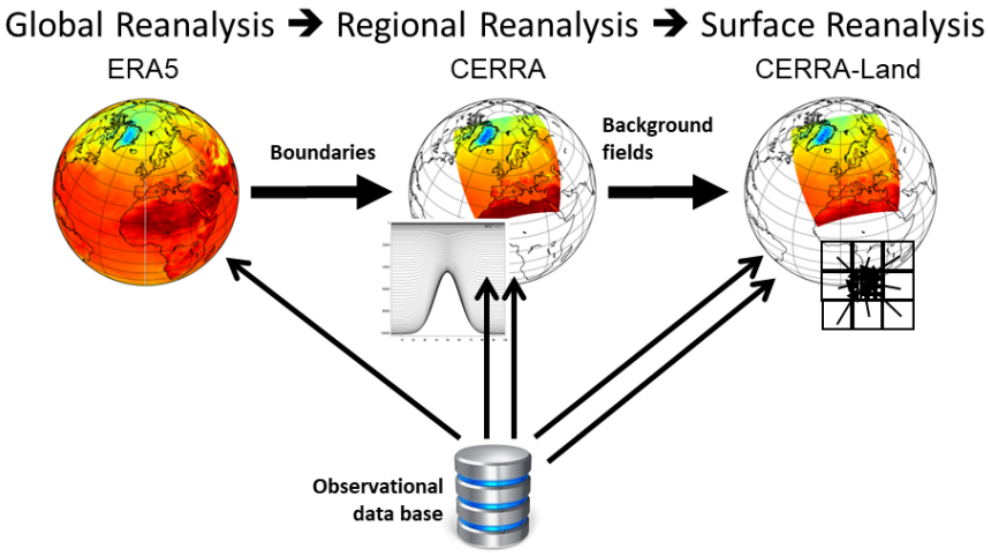


FIGURE 3.1: The reanalysis process of CERRA. Source: Copernicus-ECMWF CERRA Presentation, 2024. [15]

time. For each day, eight reanalyses are available, commencing at 00 UTC and concluding at 21 UTC, with a temporal resolution of three hours. It is commonly assumed that analysis data for these time periods exhibits superior quality compared to forecasts for the same time frames, as it is inherently more aligned with observations. The next six hours of each reanalysis data set are processed through forecasting. Furthermore, for reanalyses commencing at 00 UTC and 12 UTC, the subsequent 30 hours are processed, with a three-hour interval following the initial six hours. In this manner, users are afforded the opportunity to select the forecast they prefer, or, if available, the reanalysis result, for each hour. However, this implies that even if CERRA provides data for each hour, only the eight present in the reanalysis schedules guarantee a corrective step using observations, while the others are forecast results. Figures 3.2 and 3.3 provides a somewhat clearer view of this information, where the colour coding reflects analysis (red) and forecasts (blue).

The CERRA dataset represents a valuable resource for a wide range of applications, including the study of climate trends, renewable energy forecasting, water resource management, and risk assessment. In the context of climate research, CERRA provides a historical reconstruction of meteorological variables, which is essential for understanding long-term climate trends and the effects of climate change in Europe.

- In [21] the authors demonstrated that CERRA outperforms the ERA5-Land

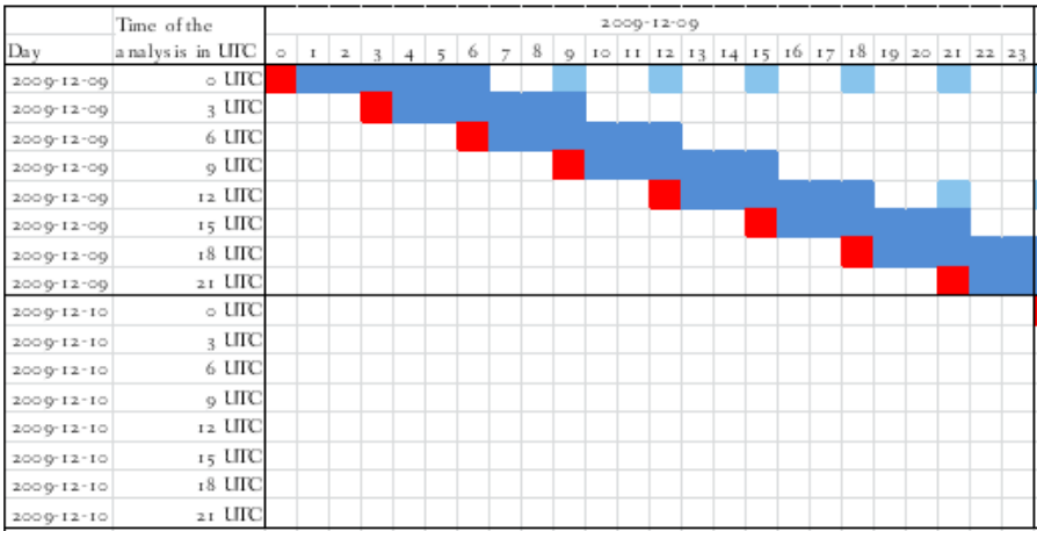


FIGURE 3.2: This table illustrates how different forecasts overlap and which options users have at a certain hour of the day. The availability of data is illustrated for the example date 2009/12/09. [15]

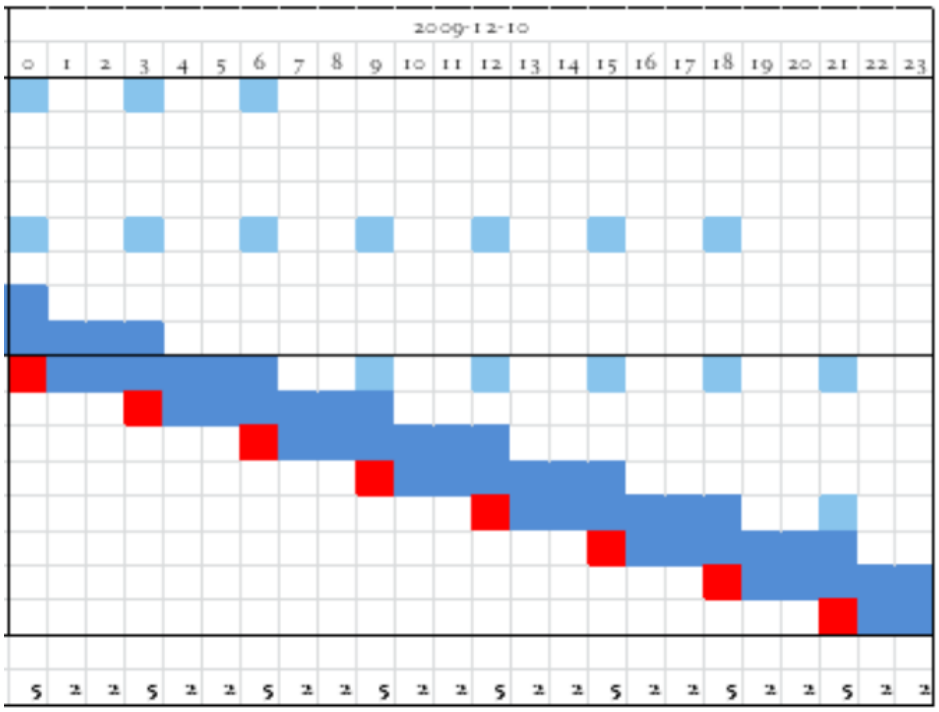


FIGURE 3.3: This table illustrates how different forecasts overlap and which options users have at a certain hour of the day. The availability of data is illustrated for the example date 2009/12/10. [15]

reanalysis in replicating temperature and other bio-meteorological variables, rendering it particularly useful for the assessment of heat-related health risks.

- In the field of renewable energy, CERRA's accurate weather observations are of great importance for the development of forecasting tools for wind and solar power, which are essential for the integration of these renewable sources into the power grid [47].
- The management of water resources also benefits from CERRA's ability to represent precipitation [69] and snow [42]. This also enables the assessment of the risks of droughts and floods.
- It has been demonstrated [49] that CERRA is a viable substitute for ground-based agro-meteorological measurements, capturing data on air temperature, actual vapor pressure, wind speed, and solar radiation.

Given the significance of CERRA in a multitude of applications, it would be optimal to have a real-time production of this data, enabling the effective study of recent events in a manner analogous to that of ERA5. Unfortunately, delays in the production and data gathering have resulted in a delay of more than two years at the time of writing, thus hindering the release of CERRA.

3.3.3 IGRA V2

The Integrated Global Radiosonde Archive version 2 (IGRA V2) [19], maintained by the National Centers for Environmental Information (NCEI), is a comprehensive dataset that plays a fundamental role in atmospheric research and climatology. IGRA V2 provides a meticulously curated collection of radiosonde and pilot balloon observations, offering invaluable vertical profiles of atmospheric parameters from around the globe. This dataset is of great importance for the understanding of atmospheric dynamics, the validation of climate models, and the support of a wide range of meteorological and climatological applications.

IGRA V2 is distinguished by its extensive temporal and spatial coverage. In fact, the dataset includes observations dating back to the early 20th century, providing a long-term record of atmospheric conditions. The spatial coverage of IGRA V2 is global, encompassing data from over 1,500 stations worldwide. The extensive geographical scope of the dataset ensures that it can be used to analyse atmospheric processes on both regional and global scales.

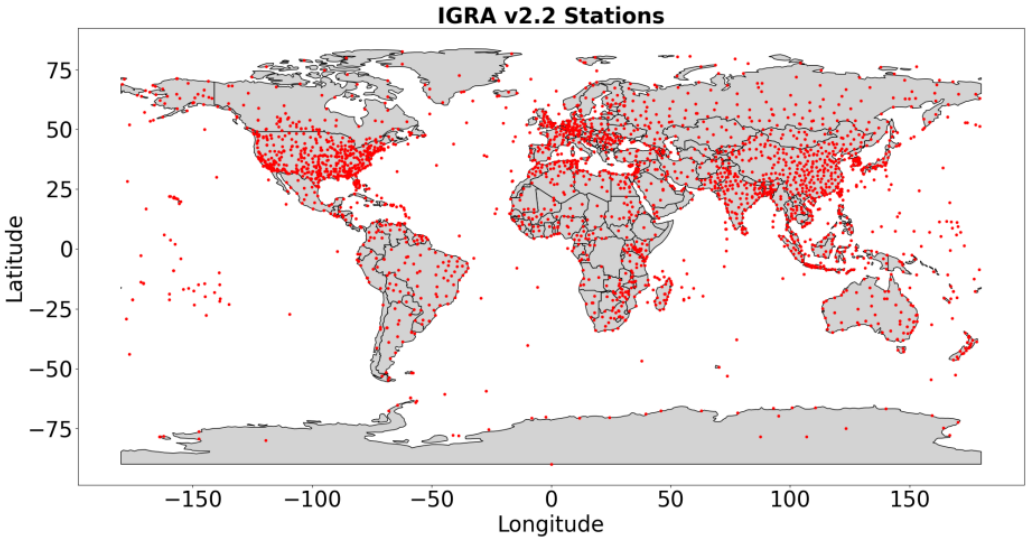


FIGURE 3.4: Map of the locations for all available IGRA v2.2 stations. [19]

The IGRA V2 undergoes a series of validation processes with the objective of ensuring the quality and reliability of the data. The validation process includes the quality control of radiosonde observations of temperature, humidity, and wind at stations. The data must be accompanied by sufficient documentation of its digital format, with only observed values included, while estimated values are excluded. This dataset thus represents a valuable source of information for the validation of results in the field of reanalysis and downscaling.

Chapter 4

Deep Learning Models for Downscaling

In the last decade, machine learning has received renewed attention in several fields, boosted by major breakthroughs achieved with deep learning (DL) models. The advantage of DL lies in its ability to extract high-level feature representations in a hierarchical manner, due to its (deep) layered structure. In particular, for spatio-temporal data, convolutional neural networks (CNNs) have gained much attention due to their ability to learn spatial features [36]. DL models allow high-dimensional problems to be handled automatically, thereby avoiding the use of conventional feature extraction techniques (e.g. principal components, PCs) that are commonly used in more classical approaches (e.g. linear models and traditional fully connected neural networks). In addition, new efficient learning methods (e.g. batch, stochastic and mini-batch gradient descent), regularisation options and computational frameworks have popularised the use of DL techniques, allowing neural networks to learn efficiently from data and avoid overfitting.

This chapter describes the basics of neural networks. We start by introducing the perceptron model in Section 4.1, followed by a brief explanation of the multi-layer perceptron in Section 4.2. We then move on to explain the principles underlying convolutional neural networks and other newer architectures, which are essential to the models developed in this thesis.

4.1 Neuron Model

A neuron model in the context of deep learning is a mathematical abstraction that mimics the behaviour of a biological neuron and serves as the fundamental building block of neural networks. The primary function of a neural model is to receive inputs, process those inputs through a set of weights, apply a non-linear

transformation, and produce an output. This process allows neural networks to learn and approximate complex functions from data.

In a typical neuron model, the following components and operations are involved:

- Inputs (Features): A neuron receives multiple inputs, which can be features from the data or outputs from other neurons. These inputs are represented as a vector $\mathbf{x} = [\, x_1, x_2, ..., x_n \,]$ where n is the number of inputs.

- Weights: Each input is associated with a weight $\omega = [\, \omega_1, \omega_2, ..., \omega_n \,]$, which represents the strength and direction of the input's influence on the neuron's output. These weights are initially set at random and are adjusted during the training process to minimise the error in the network's predictions.

- Bias: In addition to the weighted inputs, a bias term $\mathbf{b}$ is included. The bias allows the neuron to shift the activation function, enabling the model to fit the data better by providing additional flexibility.

- Weighted Sum: The neuron computes a weighted sum of the inputs plus the bias, represented as:

$$\mathbf{z} = \sum_{i=1}^{n} \omega_i \, x_i + b \tag{4.1}$$

- Activation Function: The weighted sum $\mathbf{z}$ is passed through an activation function $\phi(\mathbf{z})$. The activation function introduces non-linearity into the model, allowing the network to capture complex patterns and relationships in the data. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).

- Output: The result of the activation function is the neuron's output $\mathbf{y} = \phi(\mathbf{z})$. This output can either be the final prediction of the network (in the case of an output layer neuron) or an intermediate value passed to other neurons (in hidden layers).

Formally, the output of a neuron can be expressed as:

$$\mathbf{y} = \phi(\sum_{i=1}^{n} \omega_i \, x_i + b) \tag{4.2}$$

The training of a neural network involves the adjustment of the weights and biases in order to minimise a loss function, which quantifies the discrepancy between the network's predictions and the actual target values. This training process

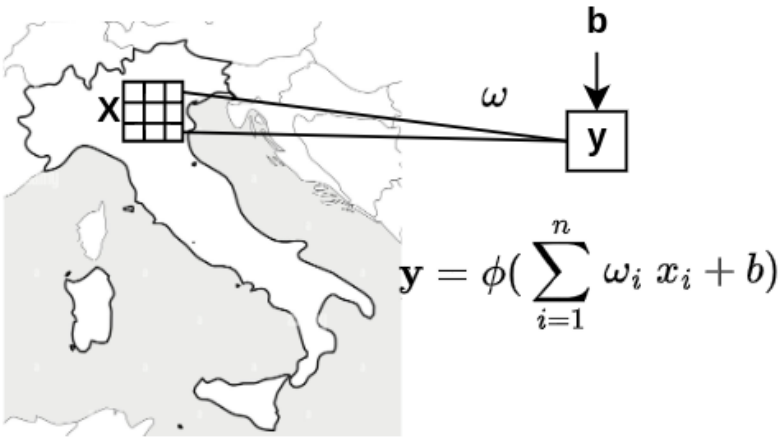


FIGURE 4.1: Diagram showing a neuron model working on a representative application.

comprises several steps. Initially, during forward propagation, the input data traverses the network, layer by layer. Each neuron computes its output by applying the weighted sum of its inputs, followed by an activation function, continuing this process until the final output is produced.

Subsequently, the loss function calculates the discrepancy between the predicted output and the true target. Commonly employed loss functions include mean squared error (MSE) for regression tasks and cross-entropy loss for classification tasks. Backpropagation, a fundamental component of the training process, entails computing the gradients of the loss function with respect to the weights and biases. By applying the chain rule of calculus, the error propagates in a backward direction through the network, from the output layer to the input layer. This backward propagation enables the network to discern the contribution of each weight to the overall error.

Utilising the gradients computed during backpropagation, the network updates its weights and biases. Optimization algorithms such as gradient descent or its variants (e.g., stochastic gradient descent, Adam) iteratively adjust these parameters to minimise the loss function. This iterative process continues until the network achieves satisfactory performance on the training data.

4.2 Dense Neural Network

A single neuron provides limited functionality, but when neurons are organized into layers, they form a neural network. This structured arrangement enables the network to perform complex tasks by learning hierarchical representations of the

input data. Neural networks comprise multiple layers, each serving a distinct purpose in the data processing pipeline.

Dense neural networks comprise three or more layers of neurons, with each neuron in a given layer activated by an affine transformation based on the neurons in the previous layer. This architectural approach was developed as a natural extension of the perceptron model and is commonly referred to as a Multi-Layer Perceptron (MLP).

The width and depth of the network influence the degree of non-linearity that can be achieved. The number of coefficients, parameters, or weights that must be adjusted by the network increases exponentially with both width and depth.

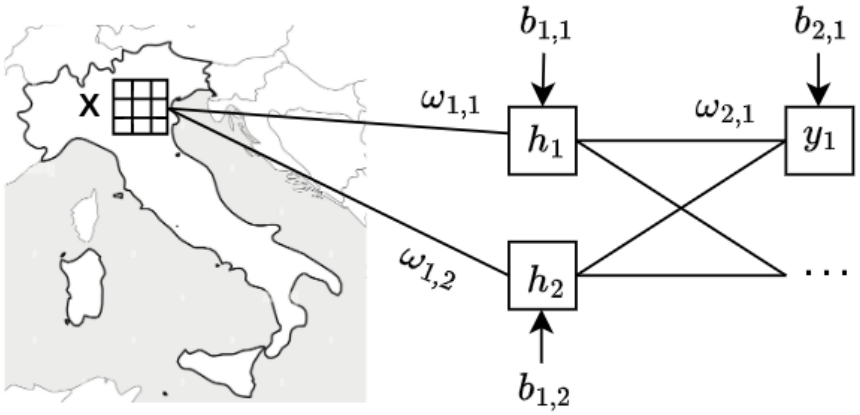


FIGURE 4.2: Diagram showing a DNN working on a representative application.

4.3 Convolutional Neural Networks

Within the broader landscape of neural networks outlined in previous sections, Convolutional Neural Networks (CNNs) [36] represent a highly specialised and effective class of models, particularly suited to the processing of grid-like data structures such as images. CNNs are designed to automatically and adaptively learn spatial hierarchies of features, making them an extremely powerful tool for visual recognition tasks.

The fundamental component of CNNs is the convolutional layer, which applies convolutional operations to the input data. In contrast to traditional neural networks, where each neuron is connected to every input, convolutional layers utilise a local connectivity pattern, whereby each neuron only processes data from a small receptive field. This approach enables the efficient capture of local spatial features.

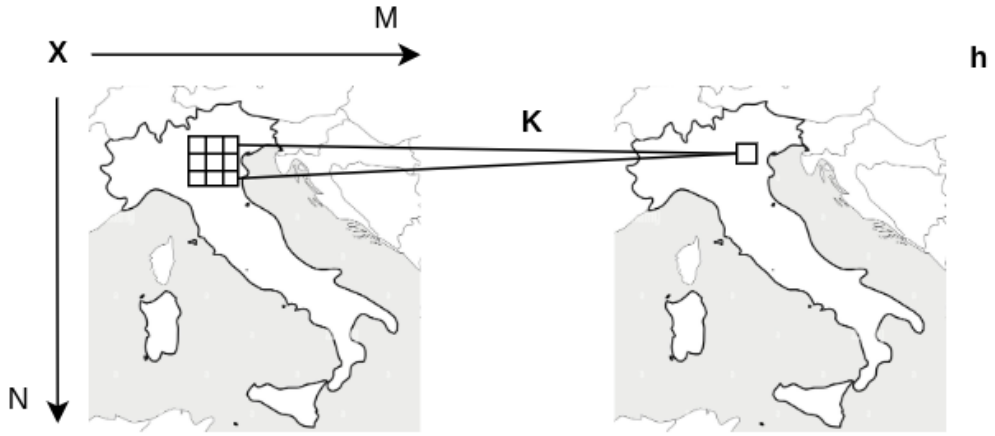


FIGURE 4.3: Diagram showing a CNN working on a representative application.

The convolution operation in a CNN applies a filter **K** to an input matrix **X** to produce a feature map **h**. An example is described in figure 4.3. For a 2D input, the convolution at a particular location can be described as:

$$h(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{X}(i+m, j+n) \cdot \mathbf{K}(m,n) \tag{4.3}$$

Convolutional layers are composed of several filters (or kernels), which are small, trainable matrices. These filters are slid over the input data, performing dot products between the filter entries and the input patch it overlaps. This operation produces a feature map, which highlights the presence of specific features such as edges, textures, or colours at different spatial locations within the input. The ability to learn multiple filters means that each convolutional layer can detect a diverse range of features from the input data. The depth of these layers allows the network to construct increasingly abstract and complex representations of the input, layer by layer. The initial layers may detect simple edges and textures, while the deeper layers can recognise parts of objects or entire object classes.

It should be noted that when the dimensions of the kernel are equal to the dimensions of the input space, the CNN is equivalent to the dense layer described in section 4.2.

4.3.1 U-Net

U-Nets [53] are a a specialised type of convolutional neural network architecture, specifically designed to excel in tasks requiring precise localisation and contextual

understanding, such as image segmentation.The distinctive symmetric encoder-decoder structure, augmented by skip connections, enables U-Nets to retain detailed spatial information while effectively processing high-resolution input data. The architecture of U-nets consists of two main components: an encoder and a decoder. These components work together to encode input data into a compressed representation and then decode it back to a desired resolution. Let us look at them in more detail:

- The encoder part of a U-Net is designed to capture the contextual information of the input through a series of convolutional operations followed by a pooling of layers. This process reduces the spatial dimensions of the input while increasing the depth of the feature maps. Each layer in the encoder typically includes two convolutional operations, followed by a rectified linear unit (ReLU) activation function, and then a max-pooling operation to downsample the feature map;
- The decoder is responsible for reconstructing the spatial dimensions while refining the details lost during the encoding phase. This is achieved through up-convolution (transposed convolution) layers that up-sample the feature maps. Skip connections link each layer in the encoder to its corresponding layer in the decoder, allowing the model to use high-resolution features from the encoder to improve the reconstruction process.

4.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [23] are a class of machine learning frameworks developed in 2014. They consist of two neural networks, the generator and the discriminator, which are trained simultaneously through adversarial processes. The generator tries to produce data that is indistinguishable from real data, while the discriminator tries to distinguish between real and generated data. The interaction between these two networks can be thought of as a game in which the generator tries to fool the discriminator and the discriminator tries to catch the generator producing false data. Here's a more detailed description of these two components:

- The generator network takes random noise as input and generates data that mimics the real data distribution. It essentially maps a latent space (noise vector) onto the data space, producing samples that resemble the real data.

The objective functions for the generator can be formulated as follows:

$$L_G = -\mathbb{E}_{x\sim p_{data}}[\log D(x)] - \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \tag{4.4}$$

- The discriminator network receives either real data or generated data as input and outputs a probability indicating whether the input data is real or generated. The discriminator is essentially a binary classifier. The objective functions for the discriminator can be formulated as follows:

$$L_D = -\mathbb{E}_{z\sim p_z}[\log D(G(z))] \tag{4.5}$$

where $D(x)$ is the discriminator's estimate of the probability that real data instance x is real, $G(z)$ is the generator's output when given noise z, and $D(G(z))$ is the discriminator's estimate of the probability that a generated instance is real. The training process alternates between updating the discriminator to maximize L_D and updating the generator to minimize L_G.

4.4.1 SRGAN

Super-Resolution Generative Adversarial Networks (SRGANs) [37] extend the GAN framework to the problem of image super-resolution. They aim to reconstruct high-resolution images from low-resolution inputs by exploiting the adversarial training mechanism of GANs to generate images with high perceptual detail.

SRGANs extend the basic GAN framework by incorporating specific architectural features tailored for super-resolution tasks in the generator and discriminator.

- The generator in SRGAN is typically a deep convolutional network designed to upsample the LR input image to the HR output image. It uses a series of residual blocks to capture high frequency detail, which is critical for generating realistic textures and edges;

- The discriminator in SRGAN is a convolutional network that attempts to distinguish between real HR images and generated HR images. It learns to evaluate the perceptual quality of the images produced by the generator.

Further improvements are incorporated in the loss function: in addition to the adversarial loss used in traditional GANs, SRGAN introduces a perceptual loss that includes both content loss and adversarial loss. The content loss is usually

defined as the pixel-wise mean squared error (MSE) between the generated HR image and the ground truth HR image.

$$L_{content} = \mathbb{E}_{x,y}[\| \phi(y) - \phi(G(x))\|^2] \tag{4.6}$$

where ϕ represents the feature maps obtained from a pre-trained deep network (e.g., VGG19), y is the ground truth HR image, and $G(x)$ is the generated HR image. The total loss for the generator combines the perceptual loss and the adversarial loss:

$$L_G = L_{content} + \lambda \mathbb{E}_x[\log(1 - D(G(x)))] \tag{4.7}$$

where λ is a weighting factor that balances the two loss components.

Super-Resolution Generative Adversarial Networks represent a significant advance in image super-resolution, using the adversarial training mechanism of GANs to produce high-quality, high-resolution images from low-resolution inputs. The combination of adversarial loss and perceptual loss enables SRGANs to produce images that are not only accurate in terms of pixel values, but also visually pleasing, capturing fine details and textures. This makes SRGANs a powerful tool for applications requiring high fidelity image reconstruction and enhancement.

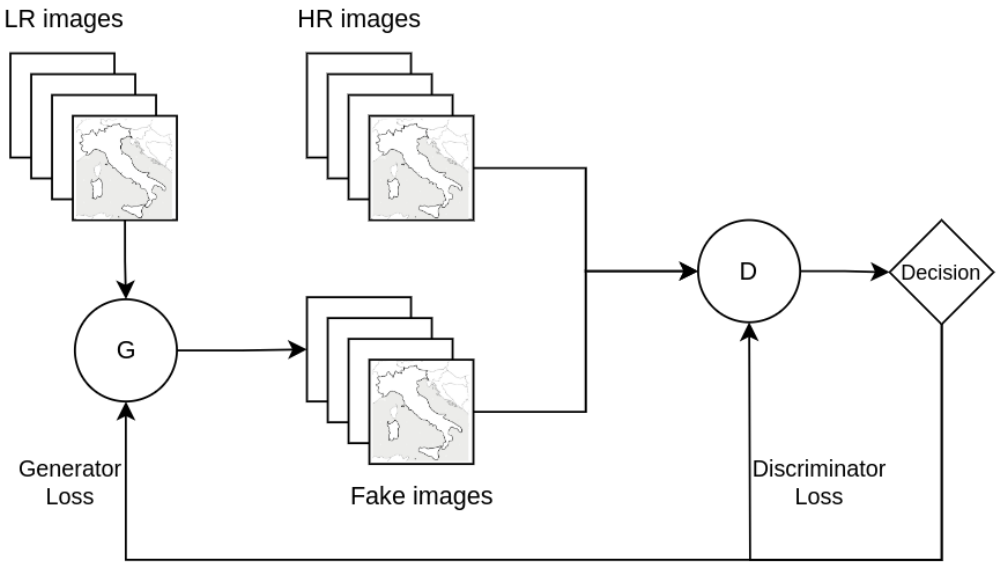


FIGURE 4.4: Diagram showing a SRGAN working on a representative application

4.5 Recurrent Neural Networks

Recurrent neural networks (RNNs) [54, 32] represent a class of artificial neural networks designed for processing sequences of data. In contrast to traditional feedforward neural networks, RNNs feature connections that form directed cycles, enabling the persistence of information. This renders them particularly well-suited for tasks where the context or state across time is crucial, such as time series forecasting, natural language processing, speech recognition, and, in our case, for the prediction of weather variables.

In an RNN, the input sequence is fed into the network one element at a time, and the output depends not only on the current input but also on the network's previous states. This is achieved through the hidden state, which acts as a memory of past inputs. Mathematically, the hidden state h_t at time step t is computed as:

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t \cdot b_h) \tag{4.8}$$

The hidden state is then used to produce the output y_t:

$$y_t = \phi(W_y \cdot h_t + b_y) \tag{4.9}$$

where:

- σ and ϕ are the activation functions (usually a tanh or ReLU function);
- W_h, W_x and W_y are the weight matrices for the hidden state the input and the output respectively;
- x_t is the input at time step t;
- b_h and b_y are bias terms.

One of the primary challenges with standard RNNs is their difficulty in learning long-term dependencies due to issues like vanishing and exploding gradients. These issues arise during the backpropagation process, where gradients can diminish or grow exponentially, making it hard for the network to learn and retain information over long sequences.

4.5.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks [27] are a type of RNN designed to overcome the limitations of standard RNNs incorporating a more complex architecture with additional gates to control the flow of information. An LSTM cell consists of three main gates:

- Forget Gate: Determines what information from the previous cell state should be discarded;
- Input Gate: Decides which new information should be added to the cell state;
- Output Gate: Controls what information from the cell state should be outputted.

A graphical representation of a LSTM cell is provided in Figure 4.5.

LSTM networks are particularly adept at capturing long-term dependencies due to their capacity to regulate the flow of information through gating mechanisms. This enables them to retain crucial information over extended sequences, thereby circumventing the challenges associated with vanishing and exploding gradients. Consequently, LSTMs have become a prevalent choice in a multitude of applications.

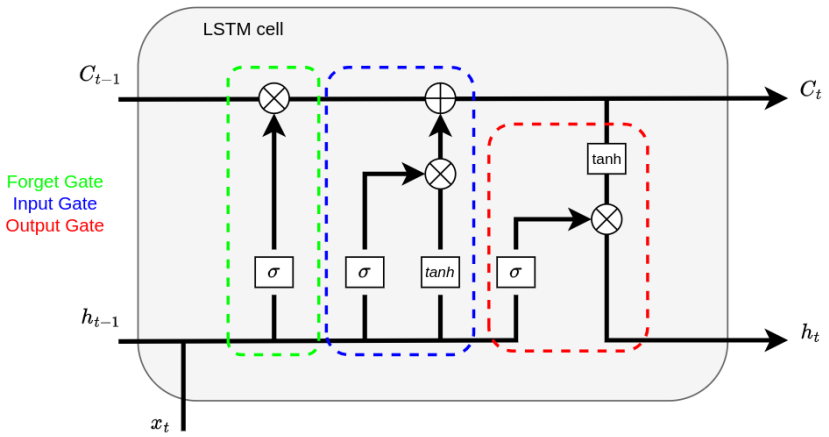


FIGURE 4.5: Graphical representation of a long short-term memory (LSTM) cell, with the three gates highlighted for clarity.

4.6 Diffusion Models

Diffusion models are a class of generative models that have recently garnered significant attention for their effectiveness in producing high-quality images and other types of data. This section will delve into diffusion models in greater detail than the rest of the chapter, as they represent a crucial aspect of the study undertaken in this thesis.

The fundamental concept underlying diffusion models is that of a diffusion process, a stochastic process describing the continuous random movement of particles over time. This process models the spread or diffusion of some quantity in space or time, with particles moving from regions of high concentration to regions of low concentration, resulting in a gradual blending or mixing of the quantity.

In the context of machine learning, diffusion models leverage the principles of diffusion processes to model the generation of data. They operate by gradually transforming a simple, often Gaussian, noise distribution into a complex data distribution through a series of iterative (diffusion) steps. This transformation is accomplished by incorporating Gaussian noise into the current data samples and iteratively refining them.

The diffusion process is divided into two distinct categories: forward and reverse diffusion processes. The forward diffusion process begins with a real image and gradually introduces Gaussian noise, resulting in a progressively noisier image over time. In contrast, the reverse diffusion process employs a neural network that learns to remove this noise in a step-by-step manner, thereby reconstructing the original image from the noisy version. This iterative refinement enables diffusion models to generate high-quality data samples from initial noise. Figure 4.6 illustrates an example of a forward and reverse diffusion process on an image.

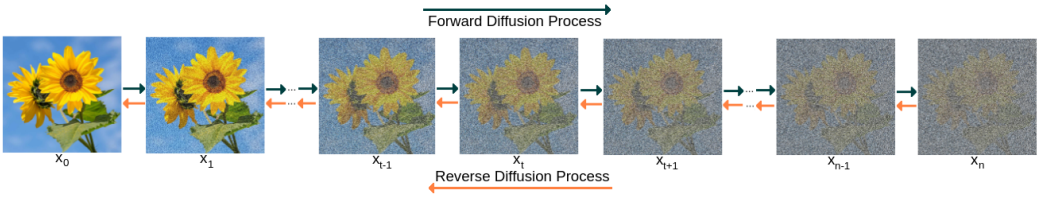


FIGURE 4.6: Forward (from left to right) and reverse (from right to left) diffusion process. [3]

From a mathematical perspective [43], considering a distribution $q(x_0)$ which generates the data, generative models aim to find a parameter vector θ such that the distribution $p_\theta(x_0)$ parameterized by a neural network approximates $q(x_0)$.

Denoising Diffusion Probabilistic Models (DDPM)[26] assume the generative distribution $p_\theta(x_0)$ to have the form

$$p_\theta(x_0)) = \int p_\theta(x_{0:T}) dx_{1:T} \tag{4.10}$$

given a time range horizon $T > 0$. where the Markov Chain formulation is :

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{4.11}$$

Training in this context is traditionally based on a variational lower bound of the negative log likelihood. It is important to note that the Kullback-Leibler Divergence $D_{\mathrm{KL}} = D_{\mathrm{KL}}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0))$ is positive, and so we obtain:

$$-\log p_\theta(x_0) \leq -\log p_\theta(x_0) + D_{\mathrm{KL}} \tag{4.12}$$

We can thus expand the second term to derive the training loss L_θ:

$$\begin{aligned}
&= -\log p_\theta(x_0) + \mathbb{E}_q\Big[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\Big] \\
&= -\log p_\theta(x_0) + \mathbb{E}_q\Big[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \log p_\theta(x_0)\Big] \\
&= \mathbb{E}_q\Big[\log q(x_{1:T}|x_0) - p_\theta(x_{0:T})\Big] = L_\theta
\end{aligned} \tag{4.13}$$

It is important to notice that in the case of diffusion models, the latent space has typically the same dimension of the visible space, so latent encodings can be visually inspected and compared with real images.

In the case of Denoising Diffusion Implicit Models (DDIM) [61], which are employed in this thesis, the diffusion process is non-Markovian and can be defined as:

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0) \tag{4.14}$$

where

$$q_\sigma(x_T|x_0) = \mathcal{N}(x_T|\sqrt{\alpha_T}x_0, (1-\alpha_T)\cdot I) \tag{4.15}$$

and

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\Big(x_{t-1}\Big|\mu_{\sigma_t}(x_0, \alpha_{t-1}); \sigma_t^2 \cdot I\Big) \tag{4.16}$$

with

$$\mu_{\sigma_t}(x_0, \alpha_{t-1}) = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}} \tag{4.17}$$

The definition of $q(x_{t-1}|x_t, x_0)$ is cleverly chosen in order to ensure two important aspects of the diffusion process of DDPM:

- the Gaussian nature of $q(x_{t-1}|x_t, x_0)$;
- the fact that the marginal distribution $q_\sigma(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\alpha_t}x_0; (1-\alpha_t)\cdot I)$ is the same as in DDPM.

Thanks to the latter property, x_t can be expressed as a linear combination of x_0 and a noise variable $\varepsilon_t \sim \mathcal{N}(\varepsilon_t|0; I)$:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\varepsilon_t \tag{4.18}$$

The subsequent step is to define a trainable generative process $p_\theta(x_{0:T})$ where $p_\theta(x_{t-1}|x_t)$ leverages the structure of $q_\sigma(x_{t-1}|x_t, x_0)$. The idea is that given a noisy observation x_t, one begins to make a prediction of x_0, and then use it to obtain x_{t-1} according to equation 4.16.

In practice, the idea is to train a neural network $\varepsilon_\theta^{(t)}(x_t, \alpha_t)$ to map a given x_t and a noise rate α_t to an estimate of the noise ε_t added to x_0 to construct x_t. As a consequence, $p_\theta(x_{t-1}|x_t)$ becomes a $\delta_{f_\theta^{(t)}}$, where

$$f_\theta^{(t)}(x_t, \alpha_t) = \frac{x_t - \sqrt{1-\alpha_t}\varepsilon_\theta(x_t, \alpha_t)}{\sqrt{\alpha_t}}. \tag{4.19}$$

Using $f_\theta^{(t)}(x_t, \alpha_t)$ as an approximation of x_0 at time step t, x_{t-1} is then obtained as follows:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \cdot f_\theta^{(t)}(x_t, \alpha_t) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \varepsilon_\theta(x_t, \alpha_t) \tag{4.20}$$

.

As for the loss function, the term in Equation 4.13 can be further refined expressing L_θ as the sum of the following terms [60]:

$$L_\theta = L_T + L_{t-1} + \cdots + L_0 \tag{4.21}$$

where

$$L_T = D_{\mathrm{KL}}(q(x_T|x_0) \parallel p_\theta(x_T))$$
$$L_t = D_{\mathrm{KL}}(q(x_t|x_{t+1},x_0)\|p_\theta(x_t|x_{t+1})) \quad \text{for } 1 \leq t \leq T-1$$
$$L_0 = -\log p_\theta(x_0|x_1)$$

All previous distributions are Gaussian and their KL divergences can be calculated in closed form using, obtaining the following formulation:

$$L_t = \mathbb{E}_{t\sim[1,T],x_0,\varepsilon_t}\left[\gamma_t\|\varepsilon_t - \varepsilon_\theta(x_t,t)\|^2\right] \tag{4.22}$$

This can be simply interpreted as the weighted Mean Squared Error between the predicted and the actual noise a time t.

In order to gain a deeper insight into the preceding explanation and to provide a more practical illustration of the functioning of diffusion models, pseudocodes of the training and sampling algorithms are presented below:

Algorithm 1 Training

repeat
  $x_0 \sim q(x_0)$
  $t \sim$Uniform(1,..,T)
  $\varepsilon \sim \mathcal{N}(0;I)$
  $x_t = \sqrt{\alpha_t}x_b + \sqrt{1-\alpha_t}\varepsilon$
  Backpropagate on $||\varepsilon - \varepsilon_\theta(x_t,\alpha_t)||^2$
until converged

Sampling is an iterative process, starting from a purely noisy image $x_T \sim \mathcal{N}(0,I)$. The denoised version of the image at time step t is obtained using equation 4.20.

Algorithm 2 Sampling

$x_T \sim \mathcal{N}(0,I)$
for $t = T,...,1$ **do**
  $\varepsilon = \varepsilon_\theta(x_a,x_t,\alpha_t)$
  $\tilde{x}_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\varepsilon)$
  $x_{t-1} = \sqrt{\alpha_{t-1}}\tilde{x}_0 + \sqrt{1-\alpha_{t-1}}\varepsilon$
end for

4.7 Transformers

The introduction of Transformers [68] in 2017 has led to a revolution in the field of machine learning, particularly in the area of natural language processing (NLP). The key innovation of the Transformer model is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence when encoding a particular word. This mechanism enables the efficient processing of long-range dependencies, making Transformers highly effective for tasks such as translation, text summarisation, and more.

4.7.1 Vision Transformers

Vision Transformers (ViTs) represent an extension of the Transformer architecture to the domain of computer vision. Introduced in 2020 [17], ViTs have demonstrated that pure Transformer architectures can achieve state-of-the-art performance on image recognition tasks, which have historically been dominated by convolutional neural networks (CNNs).

ViTs introduce a novel approach to image processing by treating images as sequences of patches, analogous to how traditional transformers handle text as sequences of words. This innovative method begins with the division of an image into fixed-size patches. Each patch is then flattened and linearly embedded into a higher-dimensional space, similar to the token embeddings used in natural language processing. Following the embedding step, the sequence of patch embeddings is augmented with a learnable positional encoding. This encoding provides information about the relative positions of the patches, compensating for the lack of inherent order in the image patches, in a manner analogous to positional encodings in text processing, which provide information about the order of words.

Once the patches are embedded and positionally encoded, they are passed through a series of transformer layers. These layers comprise multi-head self-attention mechanisms and feed-forward neural networks, which permit the model to dynamically focus on various parts of the image and learn complex contextual relationships between the patches. The self-attention mechanism enables each patch to attend to every other patch, facilitating the capture of both local and global dependencies within the image. This ability to model long-range interactions represents a significant improvement from the localized receptive fields characteristic of CNNs.

The final step involves the reconstruction of the image from these processed patches. This reconstruction structure ensures that the output image retains the spatial coherence and high-level features learned during the attention and embedding stages. The combination of patch-wise processing and the reconstruction phase allows Vision Transformers to effectively manage and manipulate image data, leveraging the strengths of the transformer architecture.

Shifted Windows Transformers

The traditional Vision Transformer framework treats an image as a sequence of patches, utilising self-attention mechanisms to model the relationships between these patches. While this methodology is effective at capturing global context, it often comes at a high computational cost and struggles with modelling fine-grained local details due to the absence of the localized inductive biases that are naturally present in CNNs. Shifted Windows Transformers were developed to address these challenges by introducing a structured approach to self-attention that efficiently balances the processing of local and global features.

The architecture of Shifted Windows Transformers is founded on two principal steps: window-based self-attention and shifted window-based self-attention.

- In window-based self-attention the image is divided into a series of non-overlapping windows of fixed size and within each of them self-attention is applied independently. This localized attention allows the model to concentrate on the intricate details within each window, facilitating a more granular understanding of local features.

- The Shifted Window-Based Self-Attention mechanism is employed to guarantee that the model captures interactions between distinct windows. This is achieved by shifting the windows by a fixed number of patches in subsequent layers. This shifting creates overlap with neighbouring windows, which enables the model to integrate information from adjacent windows and thereby capture cross-window dependencies.

Subsequently, the outputs from the window-based and shifted window-based self-attention layers are aggregated. This aggregation step enables the model to synthesise both local and global features in an effective manner, utilising the complementary strengths of each attention mechanism.

Chapter 5

Experiments

This chapter outlines the experimental framework employed in the analysis conducted for the elaboration of this thesis. The objective of this work is to downscale the meteorological variable under consideration from the resolution of the ERA5 dataset to that of CERRA, treating it as an image super-resolution task. In order to assess the efficacy of the proposed methodology, several state-of-the-art models will be compared. The foundations of these models were introduced in Chapter 4. Finally, the results will be validated against in-situ observations of the meteorological variable provided by the IGRA V2 dataset. This will enable an assessment of the extent to which the CERRA dataset deviates from the reality.

5.1 Datasets description and Preprocessing

In order to conduct the experiments, a range of ten years, from 2010 to 2019, has been selected for the training phase of the models. Two distinct years, 2009 and 2020, have been identified as testing periods. The choice of non-contiguous test years is motivated by aiming to widen the evaluation range and to match the availability of in-situ observations for validation. The lower resolution ERA5 dataset was employed as a conditioning factor in conjunction with the CERRA dataset, which served as the high-resolution ground truth. It is important to note that in order to align these datasets, only the ERA5 measurements corresponding to 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00 and 21:00, which correspond to CERRA time, were used.

5.1.1 Area Selection

The initial step was to select the area of interest on which to operate. It was decided to limit the study to a predominantly Italian area, with the following coordinates:

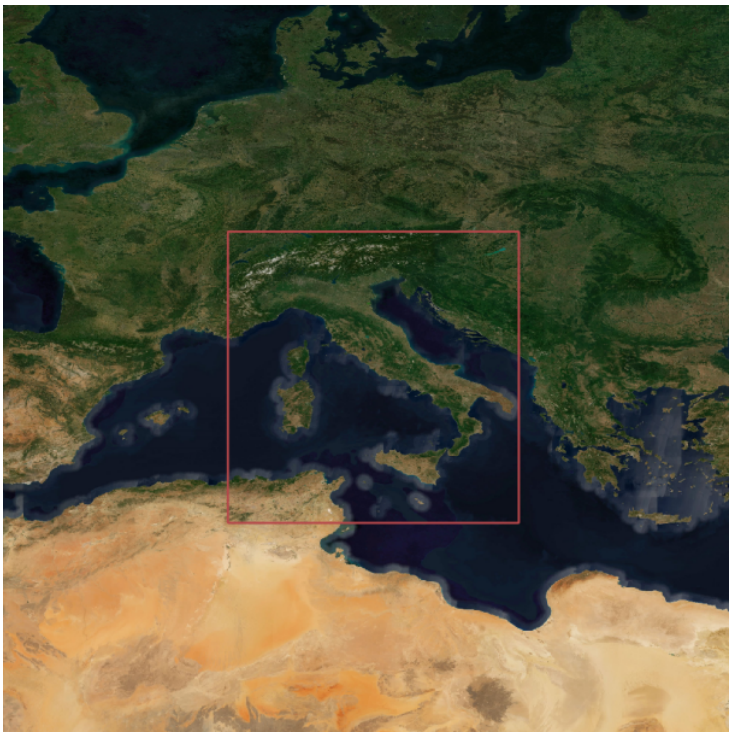


FIGURE 5.1: Selected area for the study.

- North at 47.75°;
- South at 35°;
- East at 18.75°;
- West at 6°.

The area of interest can be seen in Figure 5.1.

This area was chosen for two reasons in particular: firstly, since the research team is composed only of Italians and located in Italy, the choice fell on this country. Additionally, other countries are included, specifically Switzerland, Slovenia, and Tunisia, Algeria, France, Austria, Croatia, Hungary, Bosnia & Herzegovina, and Montenegro. The second reason is that selecting an area that is too small would have made the study less interesting, and a larger area (e.g. the entire European area covered by CERRA) would have made it unfeasible due to the computational resources required. Furthermore, using these exact coordinates allows for a target image of exactly 256x256 pixels, which we know is an excellent value to work with, as it is a power of 2.

5.1.2 Data availability

Both ERA5 and CERRA are freely available from the Copernicus Climate Change Service (C3S) website, which is part of the European Union's Copernicus Programme [14] . The Copernicus Programme represents a comprehensive initiative designed to provide accurate, timely, and easily accessible information to improve the management of the environment, understand and mitigate the effects of climate change, and ensure civil security. The Copernicus Data Store (CDS) provides a user-friendly interface and a wealth of climate data, tools, and applications, facilitating the exploration and extraction of data for a wide range of scientific and practical applications.

5.1.3 Variable selection

The focus of this study is on the meteorological variable of wind speed. Several factors influenced this decision. Firstly, the aim is to simplify the experimental framework by focusing on a single variable. Secondly, the selection of wind speed is due to the fact that, compared to variables like rainfall, it is more physically self-contained and thus easier to analyse in isolation. Furthermore, wind speed is of significant societal relevance, particularly in the context of extreme weather events and their implications for energy generation in wind farms.

In particular, the value of wind speed at 10 metres above the surface has been utilised. In the case of CERRA, it is already included as a primary variable. In contrast, for ERA5, it is possible to compute it from both the zonal u and the meridional v wind components by the the formula $ws = \sqrt{u^2 + v^2}$. The utilized values are instantaneous.

5.1.4 Area Projection

The two datasets employed utilise disparate projection mechanisms, with ERA5 employing a cylindrical projection and CERRA a Lambert Conformal Conic projection. Given that the selected study area is situated at mid-latitudes, where distortions in cylindrical projections are relatively minimal, it is recommended that the two projections be equalised by re-projecting CERRA using a cylindrical projection consistent with ERA5.

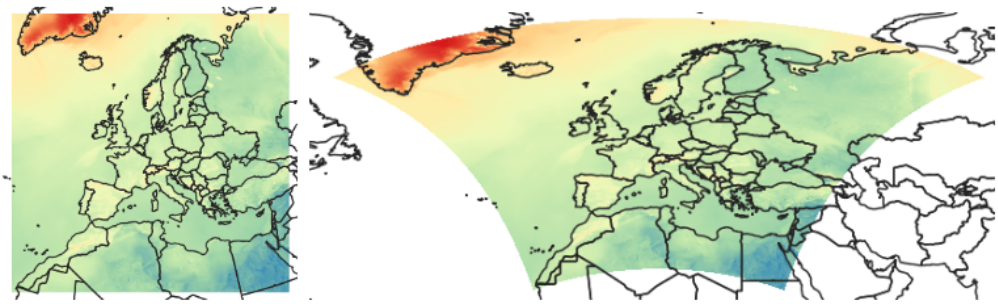


FIGURE 5.2: An example of a CERRA-Land GRIB file opened in QGIS. To the left in the Lambert Conformal Conic projection, to the right transformed to WGS 84. [15]

The projection can be implemented using the CDO library [56], in accordance with the following steps:

- Download the CERRA dataset;
- Install CDO through the command *"sudo apt-get install cdo"* or by source.
- Prepare a configuration file, *cyl.txt*, with the following parameters:
  - gridtype = lonlat
  - xsize = 256
  - ysize = 256
  - xfirst = 6
  - xinc = 0.05
  - yfirst = 35
  - yinc = 0.05
- Then is possible to convert directly the GRIB file of the dataset with the comand *"cdo remapbil,cyl.txt <inputfile.grib> <outputfile.grib>"*

5.1.5 SR framework

The application of super-resolution (SR) to neural networks necessitates the utilisation of some form of upsampling, which may be implemented in a variety of ways and at different points within the network. In general, two frameworks for SR can be delineated:

- pre-upsampling, where the upsampling is conducted in the initial section of the network;

- post-upsampling, where the upsampling component is integrated at the conclusion of the network.

If the upsampling is performed at the start, the spatial dimension of the features is increased, which typically results in improved overall performance at the expense of computational complexity.

The upsampling process can be executed using various methods. Some methods employ learnable parameters, such as transposed convolution, while others, like 2D upsampling or bilinear and bicubic upsampling, do not use learnable parameters. In this study, we opted for pre-emptive bilinear upsampling, due to its practicality. Additionally, the experiments with methods involving upsampling with learnable parameters yielded similar results.

The preemptive bilinear upsampling can be computed in a straightforward manner, directly within the data generator using the *cv2* library. In particular, the function *cv2.resize()* with interpolation *cv2.INTER_LINEAR* can be employed for this purpose. This represents the transition of the data from ERA5, which has a size of 52x52 pixels, to the size of the CERRA data, which instead is 256x256 pixels.

5.2 Models Definition

This section defines in detail all the models and methods used in the experiments carried out in this study, from a baseline to the best performing model. In order to optimise the performance of the models, it is necessary to condition them in order to guide them towards a forecast defined by the known previous weather conditions. In this study, conditioning is applied in a classifier-free manner, by concatenating the conditioning frames to the noisy images alongside the channel axis.

The input data fed into the models consists of four pre-emptively upsampled ERA5 images, along with a single image representing the noisy input. Each model then outputs a single super-resolved image. Given that the image size is set at 256×256 pixels, the input dimension is structured as (batch_size, 256, 256, 4), while the output dimension is formulated as (batch_size, 256, 256). The only exception is the diffusion model, which takes in input also the noise. In order to predict the high-resolution image at time t_0, the conditioning information includes the low-resolution images at times t_{-6}, t_{-3}, t_0 and t_{+3}. A more detailed understanding of this workflow can be gained from analysing image 5.3.

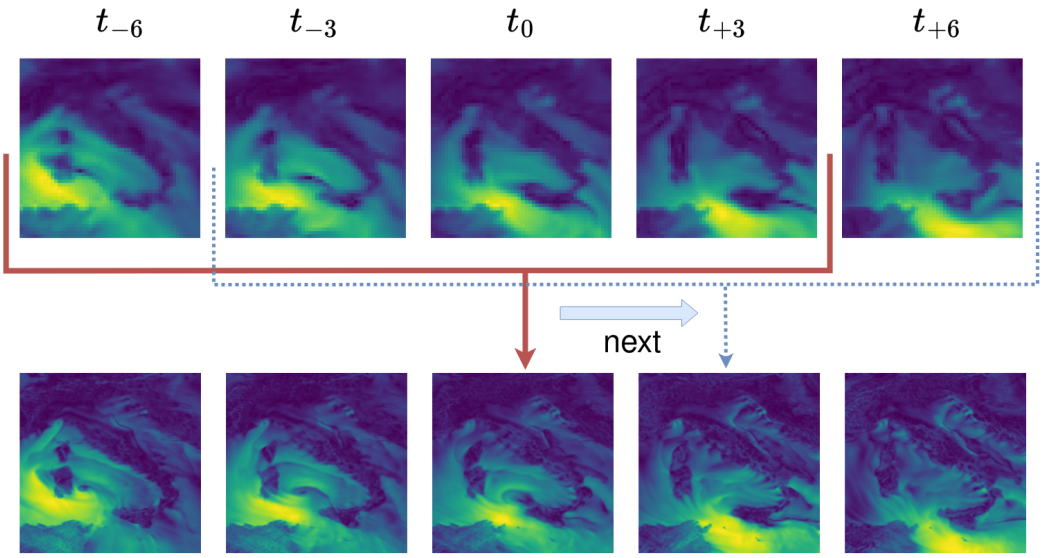


FIGURE 5.3: This image illustrates the selection of LR data for the generation of a HR image at time t_0. The images represent the wind speed over the selected territory.[43]

5.2.1 Baseline

The baseline selected for this task is simply the result of the bilinear upsampling used in the preprocessing phase. This operation ensures that the data reaches the same size as the CERRA data, allowing for comparison using the available metrics.

Bilinear interpolation is a widely used method in image processing for resizing images, particularly for scaling up or down. The fundamental principle behind bilinear interpolation is to use the values of the four nearest pixels to estimate the value of a new pixel. This method provides a balance between computational simplicity and the quality of the resulting image, making it a popular choice for many practical applications despite the existence of more advanced methods. The procedure comprises a series of methodical steps that finally result in the interpolation of pixel values, thus ensuring a smooth transition and continuity across the image.:

1. Identification of Surrounding Pixels: to begin with, consider a point (x,y) in the output image where interpolation is required. The first step involves identifying the four nearest pixel values from the input image that surround this point. These pixels are located at the integer coordinates (i, j), $(i+1, j)$, $(i, j+1)$, $(i+1, j+1)$. These coordinates correspond to the top-left, top-right, bottom-left, and bottom-right pixels relative to the point (x,y).

2. Calculation of Distance Fractions: compute the fractional distances from the point (x, y) to the integer coordinates of the surrounding pixels. Let x_d denote the fractional part of x and d_y denote the fractional part of y. These distances are given by:

$$x_d = x - \lfloor x \rfloor \quad \text{and} \quad y_d = y - \lfloor y \rfloor \tag{5.1}$$

where $\lfloor x \rfloor$ and $\lfloor y \rfloor$ represent the greatest integers less than or equal to x and y respectively.

3. Weighted Averaging: the core of bilinear interpolation involves computing a weighted average of the four surrounding pixel values. Let $Q_{1,1}$, $Q_{2,1}$, $Q_{1,2}$ and $Q_{2,2}$ denote the pixel values at coordinates (i, j), $(i+1, j)$, $(i, j+1)$ and $(i+1, j+1)$, respectively. The interpolated pixel value $P(x, y)$ at the point (x, y) is then determined by the following formula:

$$\begin{aligned} P(x,y) = Q_{1,1} \cdot (1 - x_d) \cdot (1 - y_d) + \\ Q_{2,1} \cdot x_d \cdot (1 - y_d) + \\ Q_{1,2} \cdot (1 - x_d) \cdot y_d + \\ Q_{2,2} \cdot x_d \cdot y_d \end{aligned} \tag{5.2}$$

In this expression, each term represents the contribution of one of the four surrounding pixels, weighted by the distances to the point (x, y). The weights are proportional to the inverse distances, ensuring that closer pixels have a greater influence on the interpolated value.

4. Finally, the computed interpolated value $P(x, y)$ is assigned to the corresponding location in the output image. This process is repeated for each pixel in the output image, resulting in a resized image where the new pixel values are smoothly interpolated from the original image.

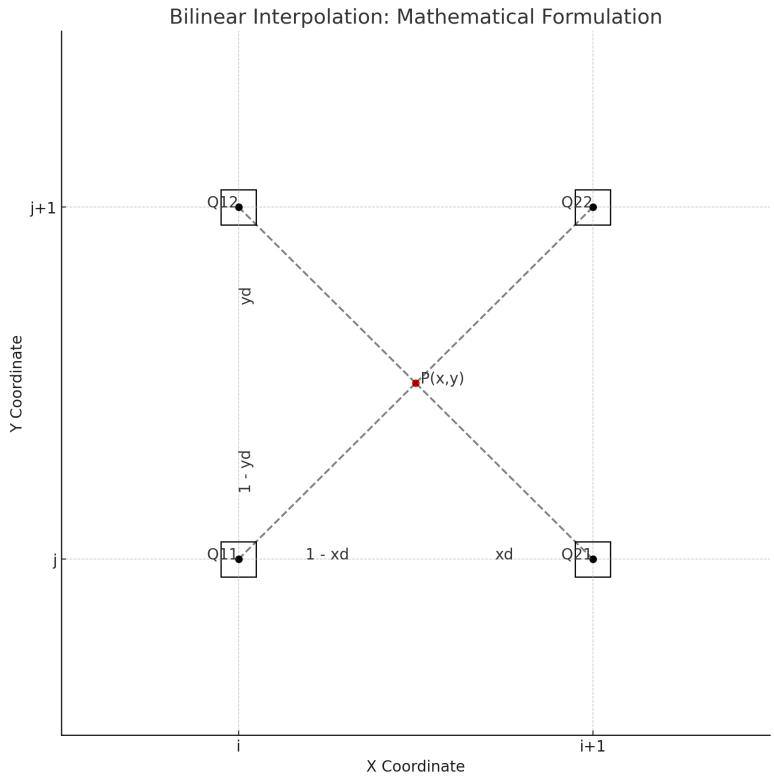


FIGURE 5.4: The image arepresenting the mathematical formulation of bilinear interpolation for pixel intensities and positions.

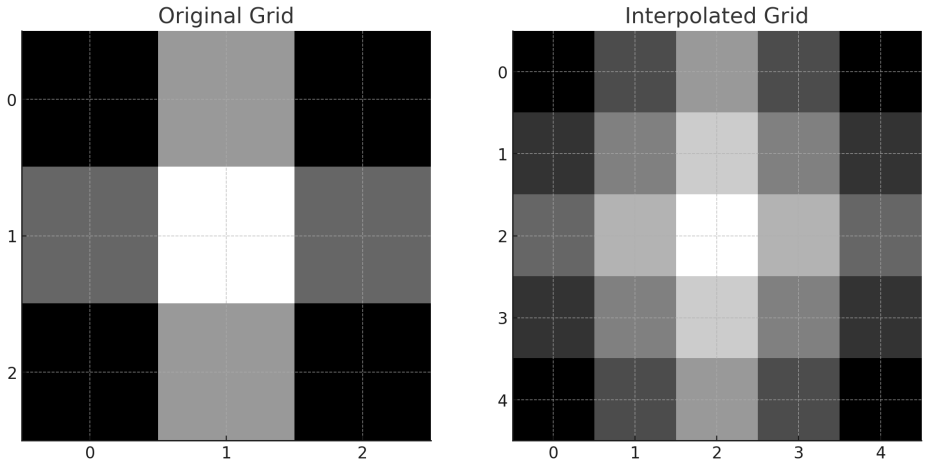


FIGURE 5.5: Visual representation of bilinear interpolation applied to a 2D image grid.

5.2.2 ESPCNN Model

The Efficient Sub-Pixel Convolutional Neural Network (ESPCNN) represents a significant advancement in the field of image super-resolution, having been introduced in 2016 [59]. This model is renowned for its innovative approach to

upscaling images, offering a more efficient and effective method compared to traditional super-resolution techniques. The fundamental concept of ESPCNN is based on the sub-pixel convolution operation, which represents a compelling alternative to conventional deconvolution or transposed convolution layers. One of the primary advantages of ESPCNN is its computational efficiency. By deferring the upscaling operation to the end of the network, ESPCNN processes lower-resolution images throughout most of the layers. This reduces the computational load and accelerates the training and inference processes. Furthermore, the architecture demonstrated the capacity to achieve high-quality super-resolution results.

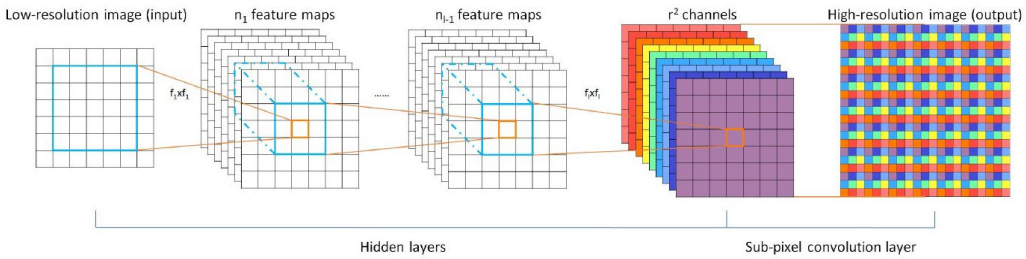


FIGURE 5.6: ESPCNN original architecture, with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step. [59]

Table 5.1 presents the hyperparameters employed to train the ESPCNN model utilised in this study.

Hyper-parameter	**Value**
Learning rate	1e-4
Epochs	200
Steps per epoch	500
Weight decay	1e-5
Optimizer	AdamW
Batch size	32

TABLE 5.1: Hyper-parameters for model ESPCNN

5.2.3 EDSR Model

The introduction of Enhanced Deep Residual Networks for Single Image Super-Resolution (EDSR) represents a significant advance in the field of image super-resolution. Introduced in 2017 [40], EDSR builds upon the concept of residual learning to deliver state-of-the-art performance in image enhancement tasks. The

model is designed to address the limitations of previous approaches, offering substantial improvements in both accuracy and computational efficiency. One of the notable innovations in EDSR is the removal of batch normalization layers. While batch normalization is commonly used to stabilize and accelerate training, the authors found that it can adversely affect the performance of super-resolution models. This architecture introduces also the concept of scale-specific models to handle different upscaling factors (e.g., 2x, 3x, 4x). Instead of using a single model for all scales, EDSR trains separate models optimised for each specific upscaling factor. This specialisation allows the model to better capture the nuances and details required for different levels of image enhancement. EDSR has continued to influence and inspire subsequent research and development in the domain of image enhancement, thereby solidifying its position as a cornerstone in modern super-resolution techniques.

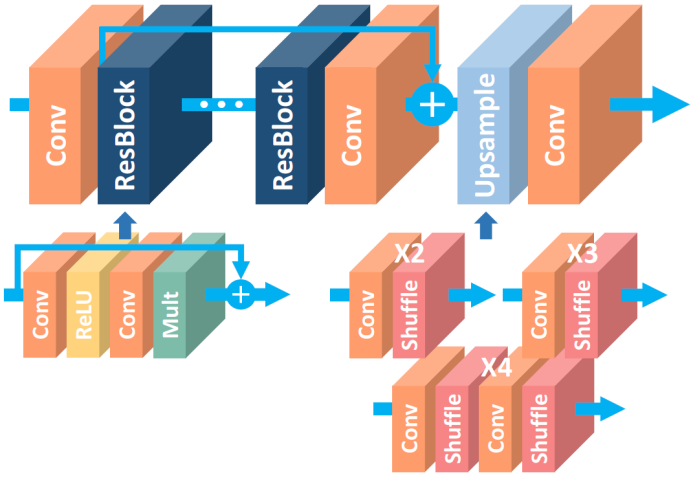


FIGURE 5.7: The architecture of single-scale SR network (EDSR). [40]

Table 5.2 presents the hyperparameters employed to train the ESPCNN model utilised in this study.

5.2.4 SwinLSTM Model

The SwinLSTM model [62] represents a sophisticated hybrid approach in the realm of image super-resolution, combining the strengths of Swin Transformers and Long Short-Term Memory (LSTM) networks. This innovative model leverages the powerful feature extraction capabilities of Swin Transformers with the temporal dynamic modeling of LSTM, providing a robust framework for high-quality image enhancement. The Swin Transformer introduced in section 4.7.1

Hyper-parameter	Value
Learning rate	1e-4
Epochs	200
Steps per epoch	500
Weight decay	1e-5
Optimizer	AdamW
Batch size	32
Filters	128
Residual blocks	5

TABLE 5.2: Hyper-parameters for model EDSR

serves as the primary feature extractor in the SwinLSTM architecture, while the LSTM component (explained in Section 4.5.1) is utilised to model temporal dependencies and dynamic changes in image sequences. While traditional super-resolution models primarily focus on spatial relationships, LSTM networks are adept at capturing sequential information, rendering them suitable for handling image sequences or video frames. This model is thus able to maximise the information derived from the images passed as conditioning for the final result and to utilise them for the improvement of the SR image in output.

The authors presented two possible versions of this architecture, a Base one and a more sophisticated Deep one. In order to achieve the greatest possible performance in this study, the Deep version of the model has been employed. A more detailed description of the architecture is provided in Figure 5.8.

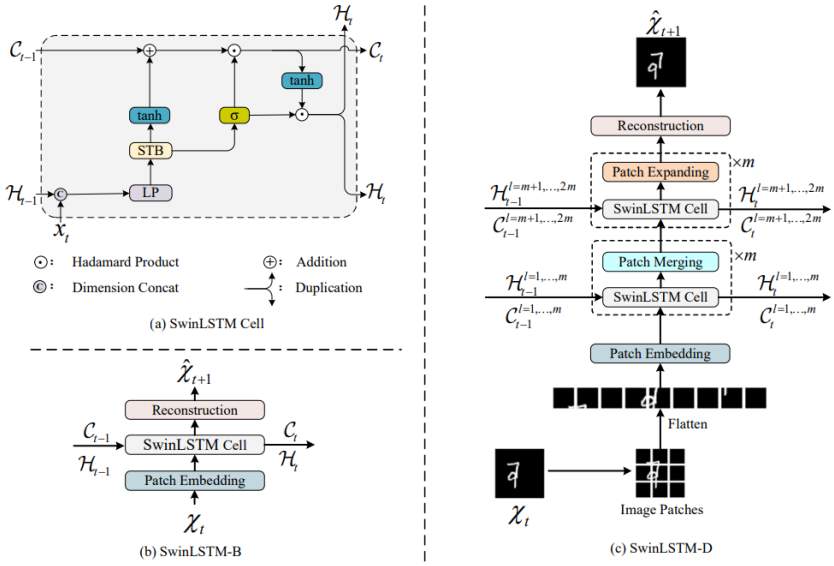


FIGURE 5.8: The two SwinLSTM architectures: base model with a single SwinLSTM cell (on the left). And the deep model with multiple SwinLSTM cells (on the right). [62]

Table 5.3 presents the hyperparameters employed to train the SwinLSTM-D model utilised in this study.

Hyper-parameter	Value
Learning rate	1e-4
Epochs	20
Weight decay	1e-5
Optimizer	AdamW
Batch size	32
Depths downsample	[2, 2, 2, 2]
Depths upsample	[2, 2, 2, 2]
Patch size	2
Heads	[8, 8, 8, 8]
Window size	2

TABLE 5.3: Hyper-parameters for model SwinLSTM

5.2.5 SRGAN Model

The functioning of SRGAN models [37] has already been explained in Section 4.4.1, where the study of their architecture also detailed a broad spectrum of decisions and the prevalence of use of the perceptual loss component into the generator's loss function. However, our experiments indicated that utilising a mean

squared error (MSE)-based non-perceptual loss, in conjunction with a Patch-GAN [29] discriminator, proved to be more efficacious than its perceptual counterpart. This approach yielded optimal performance outcomes.

Table 5.4 presents the hyperparameters employed to train the SRGAN model utilised in this study.

Hyper-parameter	**Value**
Learning rate	2e-4
Epochs	160
Steps per epoch	500
Optimizer	Adam
Beta1	0.5
Batch size	2

TABLE 5.4: Hyper-parameters for model SRGAN

5.2.6 Residual U-Net Model

The residual U-net is a U-net implementation that is employed by the diffusion model to perform the denoising component. The network utilises residual blocks and implements downscaling via average pooling and upsampling via interpolative resizing. The experiments conducted demonstrated that it also functions as a standalone super-resolution model, with notable effectiveness in this setting, providing an effective benchmark against the diffusion model. This comparison is of particular significance due to the architectural parallels and similarities in parameter count between the two models. The shared characteristics allow for a more accurate understanding of their respective strengths and capabilities in the field of super-resolution.

Table 5.5 presents the hyperparameters employed to train the Residual U-Net model utilised in this study.

Hyper-parameter	Value
Learning rate	1e-4
Epochs	110
Steps per epoch	500
Weight decay	1e-5
Optimizer	AdamW
Batch size	8
Widths	[64, 128, 256, 384]
Block depth	3

TABLE 5.5: Hyper-parameters for model Residual U-net

5.2.7 ViT Model

Furthermore, the functioning of the Vision Transformer [17] models has already been elucidated in the previous chapter (Section 4.7.1). The implementation involved in this study is a tuned version of the base architecture of ViTs with the reconstruction process implemented via transposed convolution upsampling alternated with residual blocks.

Table 5.6 presents the hyperparameters employed to train the ViT model utilised in this study.

Hyper-parameter	Value
Learning rate	1e-4
Epochs	220
Steps per epoch	500
Weight decay	1e-5
Optimizer	AdamW
Batch size	8
Patch size	8
Heads	12
Transformer layers	6

TABLE 5.6: Hyper-parameters for model ViT

5.2.8 Diffusion Model

Diffusion models, previously introduced in Section 4.6, operate as iterative denoising algorithms. Their main trainable component is the denoising network. The training of this underlying denoising network is conducted in accordance with conventional procedures, and we have elected to employ the U-net architecture described in Section 5.2.6.

Table 5.7 presents the hyperparameters employed to train the Diffusion model utilised in this study.

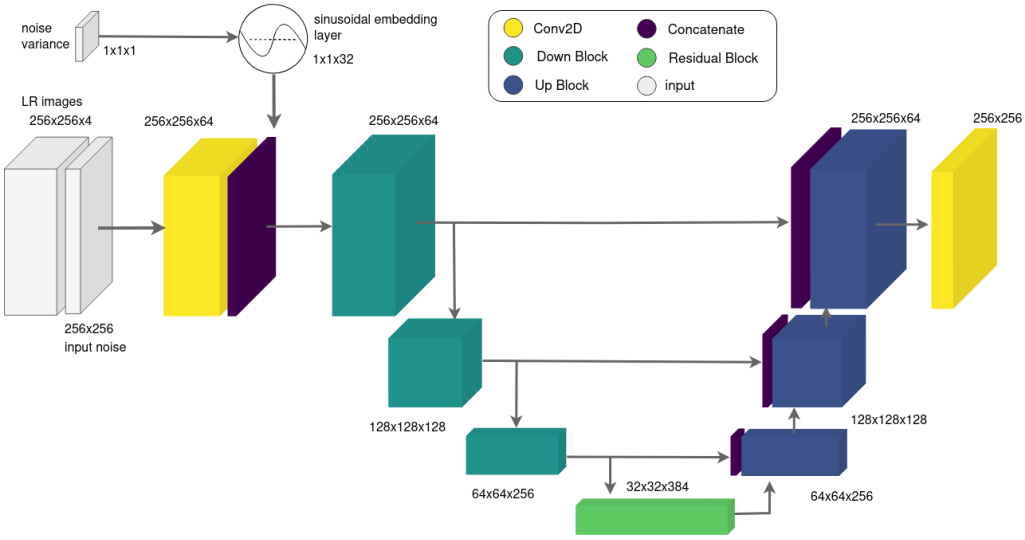


FIGURE 5.9: Architectural design of the denoising U-Net employed in the Diffusion Model.

Generative Ensemble Diffusion

In a similar manner to a previous study [3], this investigation employs the ensemble properties of diffusion models. The objective of diffusion models is to approximate the distribution of the training data. Consequently, the generative outcome is expected to be a probabilistic point within the data distribution. It is common for diffusion processes with identical conditioning to yield disparate results across different executions and this behaviour of may impede the prediction process, as a single generation may diverge considerably from the mean of the distribution, resulting in a convincing yet highly unlikely outcome within the distribution. One potential solution to this issue is to compute the mean of a set of diffusion generations, which would move the generated image closer to the mean value of the distribution and therefore generate a more probable outcome.

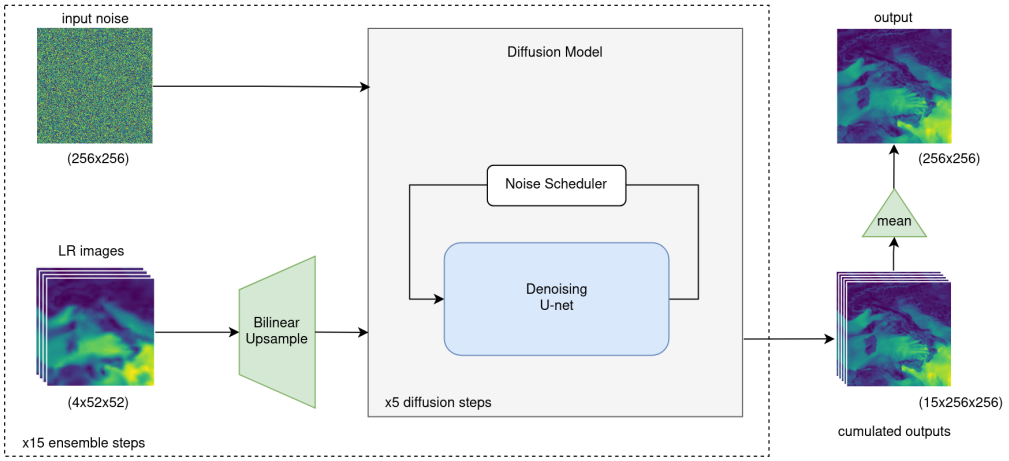


FIGURE 5.10: Diagram explaining how the Diffusion Model works and how Generative Ensemble Diffusion is integrated. [43]

Differently from the approach used in [3], the methodology employed simply calculates the mean between the predicted instances, rather than utilising a post-processing model. This more straightforward approach yielded comparable results with a reduced computational load. In our experiments, we opted for an ensemble comprising 15 executions of the single diffusion model, with each diffusion comprising 5 steps. This combination proved to be the optimal configuration with respect to our testing years.

Figure 5.10 provides a general overview of the ensemble architecture. Starting from the left, it can be seen that the inputs are the noise and four low-resolution conditioning images. Pre-emptive bilinear upsampling is performed on the low-resolution images before their insertion into the model. The model then performs five diffusion steps of sequential denoising, with a noise scheduler managing the re-addition of the noise and the computation of noise rates. The noise scheduling process requires the generation of a linear schedule, with denoising operations alternating with the re-addition of noise at a lower rate. A pseudocode of the algorithm is reported in Algorithm 3.

This operation is performed 15 times as part of the ensemble effort, using the same conditioning information but different input noise. The cumulative outputs are finally combined via a mean operation to produce the downscaled output. This operation is effectively performed in a parallelised manner with a batch size of 32.

Algorithm 3 Noise Scheduler

```
next_noisy_images ← initial_noise
for step in diffusion_steps do
    noisy_images ← next_noisy_images
    noisy_variance ← diffusion_schedule(step)
    pred_noise, pred_image ← denoising_unet(noisy_images, noisy_variance)
    next_variance ← diffusion_schedule(next(step))
    next_noisy_images ← pred_image + next_variance * pred_noise
end for
```

Hyper-parameter	Value
Learning rate	1e-4
Epochs	220
Steps per epoch	500
Weight decay	1e-5
Optimizer	AdamW
Batch size	8
Widths	[64, 128, 256, 384]
Block depth	3
Diffusion Steps	5

TABLE 5.7: Hyper-parameters for model Single Diffusion. It is important to note that the model represented by these parameters is the same model used to generate the ensemble diffusion results.

5.3 Training and Evaluation

All training was conducted on an Nvidia RTX 4000 and all the models were developed utilizing the TensorFlow/Keras and PyTorch frameworks, which are renowned for their robustness and flexibility in deep learning applications. The training dataset encompassed data from 2010 to 2019, while the testing dataset included data from the years 2009 and 2020. This strategic partitioning was designed to rigorously evaluate the models' performance across temporal boundaries that both precede and follow the training period, thereby ensuring a comprehensive assessment of their generalization capabilities.

To manage the substantial volume of data efficiently, a custom data generator was employed. This generator played a crucial role in the dynamic creation of data sequences and the efficient handling of training batches. Specifically, the generator utilized memory mapping techniques to load only the necessary data into RAM, significantly optimizing the memory usage and computational efficiency. Data batches were randomly selected across the ten-year training span, enhancing the model's ability to generalize by preventing overfitting to specific time periods.

The training process was conducted in mini-batches of 8 samples, iterating over a large number of epochs to ensure thorough learning. The AdamW optimization algorithm, which combines the benefits of the Adam optimizer with weight decay regularization, was utilized to update the model parameters. This approach helps in mitigating overfitting by incorporating regularization directly into the optimizer.

To further refine the training process, a sophisticated callback function was implemented to adjust the learning rate and weight decay dynamically. Initially, the learning rate was set at 1×10^{-4} and gradually reduced to 1×10^{-5} as training progressed. Similarly, the weight decay parameter was initialized at 1×10^{-5} and subsequently decreased to 1×10^{-6}.

During the evaluation phase, a separate generator was employed to produce continuous sequences tailored for each testing year. This generator was designed to handle a batch size of 32, ensuring efficient processing of the test data. Given the three-hour temporal resolution of the dataset, each batch represented a 96-hour (four-day) period. This approach facilitated a detailed and continuous assessment of the models' performance over extended timeframes, providing insights into their ability to maintain accuracy and robustness over varying temporal conditions.

5.3.1 Validation

The validation process employs in-situ observations from weather stations as the ground truth against which the model predictions are evaluated. This step is crucial not only for assessing the models' performance but also for evaluating the quality of the reanalysis datasets provided by ERA5 and CERRA. These in-situ observations offer a reliable benchmark, enabling a thorough comparison between observed and predicted values. The IGRA V2 dataset, which provides comprehensive meteorological data, was selected as the primary source of validation data. The dataset comprises wind speed measurements at various pressure levels, with the maximum value corresponding to the altitude of the weather station. To ensure consistency with the ERA5 and CERRA datasets, only the observation with the highest air pressure value was selected for each measurement. This criterion aligns the in-situ observations with the structure of the reanalysis data, facilitating a coherent validation process.

Given the temporal characteristics of the datasets from ERA5 and CERRA, which report measurements at hourly intervals with specific data points recorded every three hours, it was imperative to structure the in-situ data accordingly. The IGRA data, characterized by an irregular distribution of timestamps with concentrations around midnight and midday, necessitated a preprocessing step to standardize the temporal resolution. Measurements not occurring at these key times were excluded. Additionally, for instances where multiple measurements were recorded by the same station within the hour preceding these slots, the values were averaged, and the corresponding timestamp was rounded to the start of the subsequent hour. This approach ensured temporal alignment across all datasets, facilitating a direct and fair comparison.

To enable a spatially coherent comparison, we mapped all values to the spatial grid of the in-situ data. This decision allowed us to include only those data points that had corresponding values in the ERA5 and CERRA datasets, as well as in the model predictions. By ensuring that each comparison involved data present in all three sources, we significantly reduced the potential for spatial discrepancies, thus enhancing the reliability of the validation process. This strategy also minimized the storage requirements by focusing only on relevant data points, streamlining the computational resources needed for validation.

The procedure for assembling the validation dataset is outlined in Figure 5.11. This diagram provides a comprehensive overview of the steps involved in preparing the validation data, ensuring clarity and reproducibility of the process. The resultant dataset is then curated to facilitate the calculation of differences between

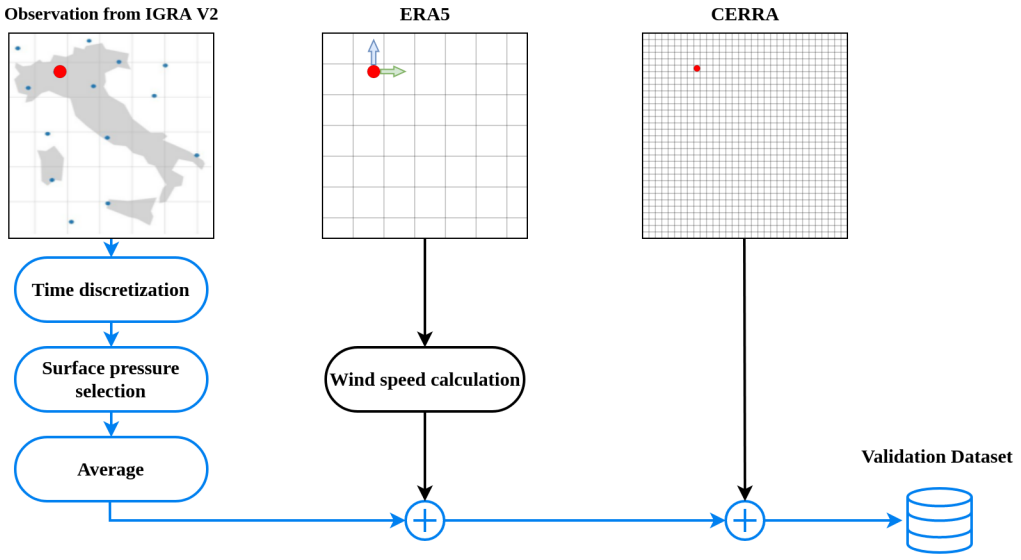


FIGURE 5.11: The image illustrates the construction process of the dataset employed for the validation of ERA5, CERRA and the generated data, which exploits the in-situ observations from IGRA V2. [43]

the model predictions and the ground truth observations. This comparison is pivotal for assessing the accuracy and performance of the various models employed to approximate CERRA.

Although the primary objective of this analysis is to evaluate the models' predictive capabilities, this process also yields valuable insights into the performance of the ERA5 and CERRA reanalysis datasets themselves. By comparing the model outputs against the in-situ observations, we can discern the degree of alignment between the reanalysis datasets and actual meteorological conditions.

5.4 Results and Discussion

This section details the experiments conducted using the models and data previously outlined, along with an analysis of the results obtained. The primary objective of the experiment is to evaluate the effectiveness of a neural-network-based super-resolution model in replacing a traditional downscaling model, such as the physics-inspired HARMONIE model currently employed for the generation of CERRA. One key advantage of the neural model is its independence from supplementary data requirements, along with significantly reduced computational demands. The comparative analysis encompasses eight competing neural models: ESPCNN, EDSR, SwinLSTM, SRGAN, U-net, ViT, Single Diffusion, and

Ensemble Diffusion. This general comparison includes a variety of models, ranging from small, efficient networks to state-of-the-art, high-computational-demand approaches. The assessment aims to determine the feasibility and relative performance of these neural models in the context of super-resolution tasks for weather variables. In order to facilitate the aforementioned analysis, it is also possible to consider the table 5.8, which presents a comparative analysis of the tested models, including the number of parameters and the execution time for the computation of one year of data.

Parameters and execution times		
Model	**Parameters**	**Time**
ESPCNN [59]	0.72	23
EDSR [40]	2.17	29
SRGAN [37]	4.31	63
ViT [17]	23.23	69
SwinLSTM [62]	3.41	134
U-net [53]	19.97	246
Single Diffusion	19.97	385
Ensemble Diffusion	19.97	7281

TABLE 5.8: The table above presents a comparative analysis of the tested models, focusing on the number of parameters (in millions) and the execution time (in seconds) for the computation of one year of data.

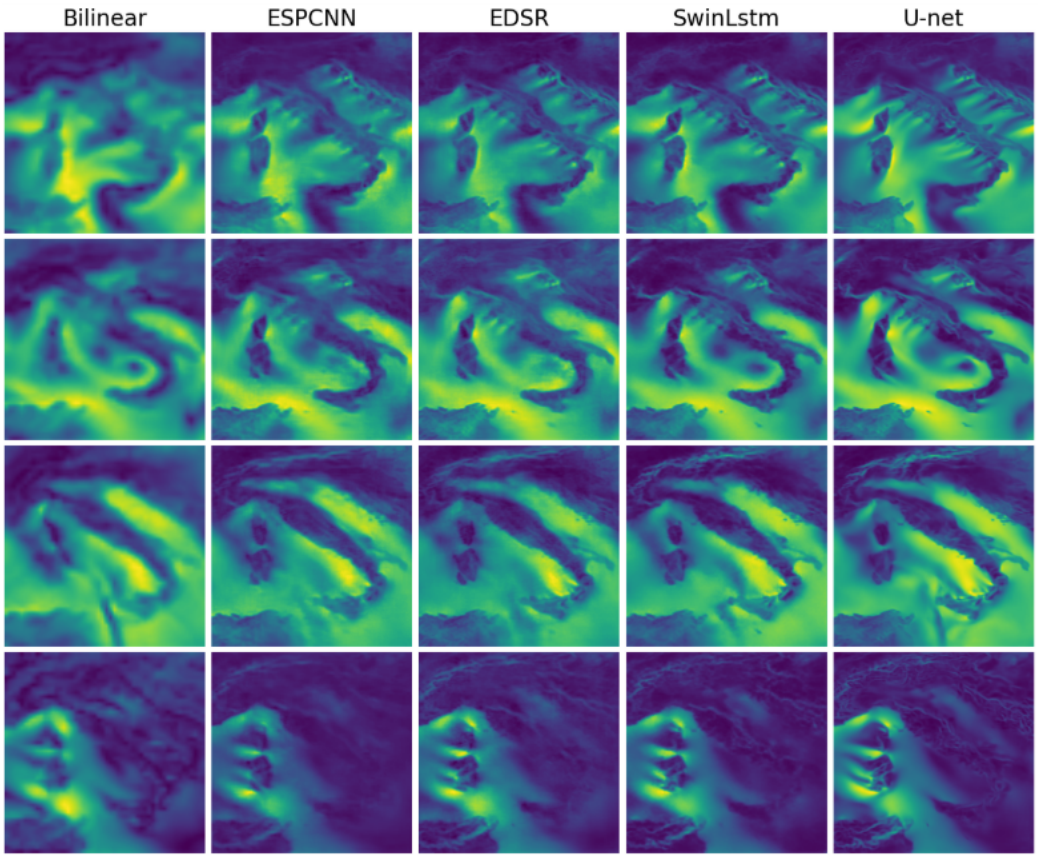


FIGURE 5.12: This figure presents a visual evaluation of super-resolution methods for ten selected samples from the years 2009 and 2020, showcasing different weather conditions, including a Mediterranean cyclone. In this first image, the Baseline model and the four least performant models in terms of SSIM metric are reported, namely ESPCNN, EDSR, SwinLSTM and U-Net.

As previously indicated, the models are trained on a 10-year period with a sequence-to-one approach, utilising four conditioning low-resolution time frames, and they are then evaluated on the two different years. A summary of the obtained results is presented in Table 5.9, where it can be observed that all neural models exhibit considerably superior performance with respect to our traditional downscaling baseline of bilinear interpolation. The model that achieves the best result, in terms of all three metrics, is Ensemble Diffusion. In comparison to bilinear interpolation, Ensemble Diffusion improves the MSE value from 2.50e-03 to 1.02e-03, and provides a 15% improvement in PSNR and a 20% improvement in SSIM. A set of results for visual inspection is reported in Figures 5.12 and 5.13.

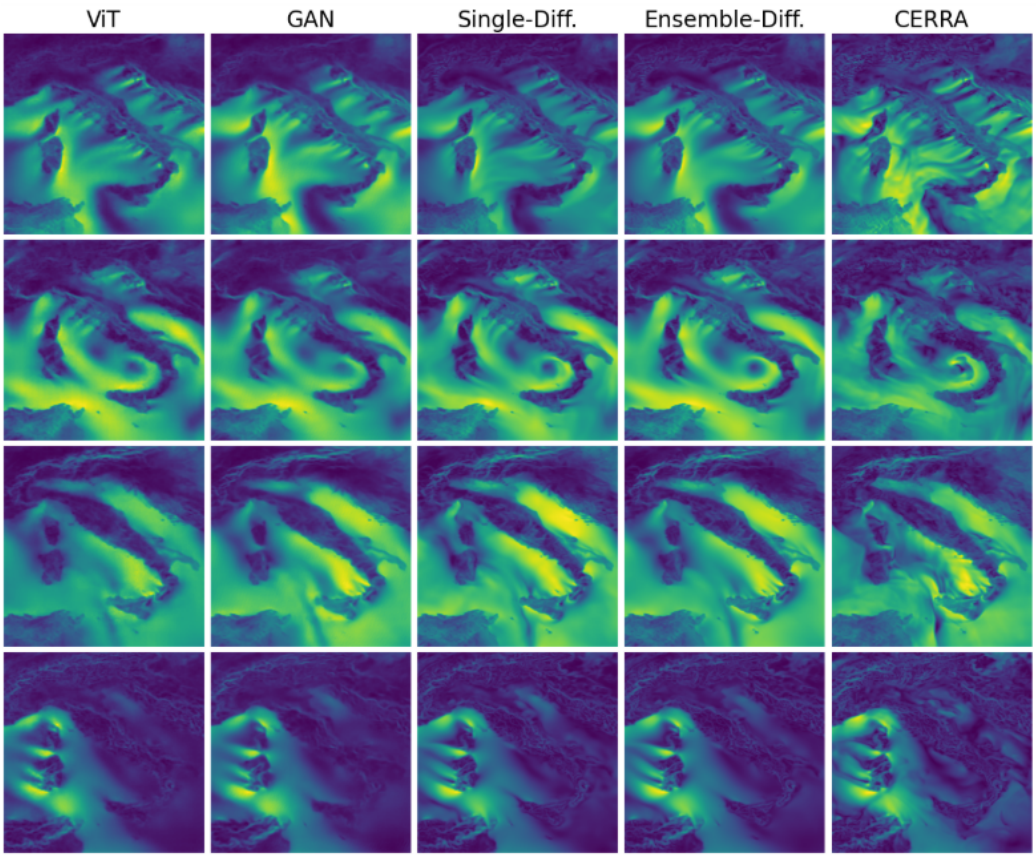


FIGURE 5.13: This figure presents a visual evaluation of the four most performant models in terms of the SSIM metric. These are ViT, SRGAN, Single Diffusion and Ensemble Diffusion. The ground truth represented by CERRA is also reported. The comparison is made on the same data presented in Figure 5.12.

Wind speed ERA5 to CERRA Downscaling			
Model	**MSE ↓**	**PSNR ↑**	**SSIM ↑**
	2009		
Bilinear	2.50e-03	26.36	0.708
ESPCNN [59]	1.31e-03	29.11	0.773
EDSR [40]	1.20e-03	29.48	0.796
SwinLSTM[62]	1.14e-03	29.70	0.802
SRGAN [37]	1.10e-03	29.85	0.813
U-net [53]	1.16e-03	29.65	0.819
ViT[17]	1.09e-03	29.94	0.822
Single Diffusion	1.18e-03	29.62	0.829
Ensemble Diffusion	1.06e-03	30.11	0.844

TABLE 5.9: A comparison of the results of all the tested models across the year 2009

Wind speed ERA5 to CERRA Downscaling			
Model	**MSE ↓**	**PSNR ↑**	**SSIM ↑**
	2020		
Bilinear	2.38e-03	26.67	0.712
ESPCNN[59]	1.25e-03	29.31	0.775
EDSR[40]	1.14e-03	29.70	0.797
SwinLSTM[62]	1.10e-03	29.99	0.804
SRGAN[37]	1.05e-03	30.07	0.814
U-net [53]	1.10e-03	29.86	0.820
ViT[17]	1.05e-03	30.18	0.824
Single Diffusion	1.13e-03	29.84	0.831
Ensemble Diffusion	1.02e-03	30.32	0.845

TABLE 5.10: A comparison of the results of all the tested models across the year 2020.

The data generated through our process was validated against in-situ measurements, providing a crucial comparison to determine if the downscaled version offered tangible improvements over real-world observations. As indicated in Table 5.11, the validation confirmed that our downscaled data aligned more closely

with the actual wind speed measurements in the area, demonstrating enhanced fidelity. In order to evaluate the performance of the models, we selected the Mean Absolute Error (MAE) as our preferred metric for assessing errors. This was due to the tangible physical interpretation of the metric, which is measured in meters per second. This provides a clear indication of the magnitude of the error.

Our validation analysis revealed that CERRA outperforms ERA5 in terms of real-world data correlation. This was achieved through a 8.82% improvement in Mean Absolute Error (MAE). Among the downscaling models tested, the Ensemble Diffusion, SRGAN, and ViT methods demonstrated the most effective performance, with an improvement of 8.82% over ERA5, which is the same level of performance demonstrated by CERRA.

Wind speed validation for the year 2009	
Model	**MAE** $\downarrow$
ERA5	2.04
CERRA	<u>1.86</u>
Bilinear	1.96
ESPCNN[59]	1.90
EDSR[40]	1.92
SwinLSTM[62]	1.90
SRGAN[37]	<u>1.86</u>
U-net [53]	1.88
ViT[17]	<u>1.86</u>
Single Diffusion	1.90
Ensemble Diffusion	<u>1.86</u>

TABLE 5.11: Validation of the results of all the tested models, in addition to ERA5 and CERRA, for the year 2009.

These results demonstrate the efficacy of neural models in approximating the inner mechanics of state-of-the-art physics-inspired models, offering both reduced computational costs and the avoidance of the need for additional information, which is often difficult to obtain, with common delays in the publication of current data. Our validation was limited to the year 2009, as validation data was not available for the year 2020.

Figure 5.14 presents a comparative analysis of SSIM metrics across the various downscaling methods for the year 2020. It can be observed that there is a significant fluctuation in performance throughout the year. These variations, which are consistently observed across all models, including bilinear interpolation, suggest that temporal weather variations significantly influence the efficacy

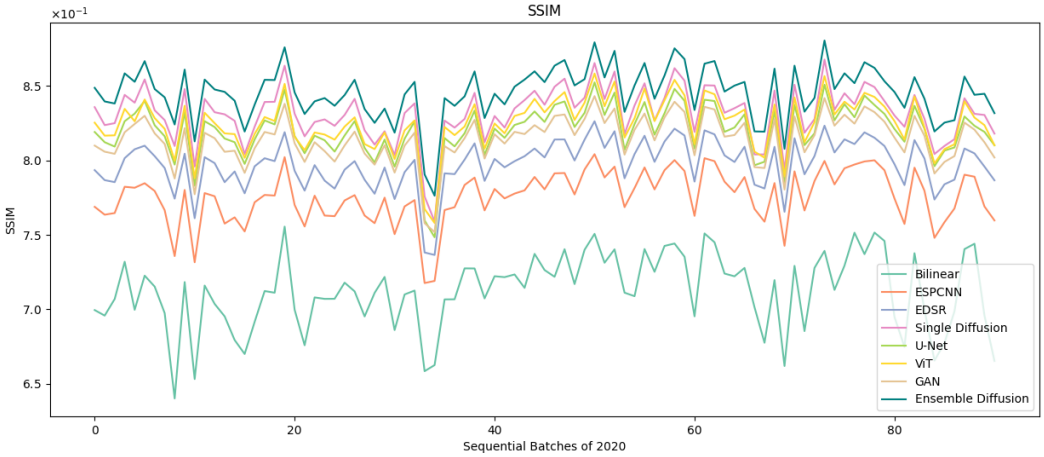


FIGURE 5.14: A graph comparison of the SSIM metric for the different tested models is presented. SSIM is calculated for each batch sequentially throughout the testing year of 2020, thereby highlighting the variability of performance at different times of the year.

of the downscaling models. This consistent trend across different models implies that the performance changes are predominantly driven by weather-related factors rather than the intrinsic characteristics of the models themselves.

An additional analysis is presented in Figure 5.15, which depicts the values predicted by the models and those available in ERA5 and CERRA, along with those expressed by the observations of a meteorological station over the course of the year. This image illustrates two key points: firstly, that each line follows a similar pattern, demonstrating that even when working with reanalysis data, it is possible to capture real-world patterns in the variables under consideration. The second observation is that, in general, the reanalysis appears to underestimate the value of wind speed. It should be noted, however, that this second point may depend on the specific meteorological stations used for the study or may be related to the wind speed variable itself.

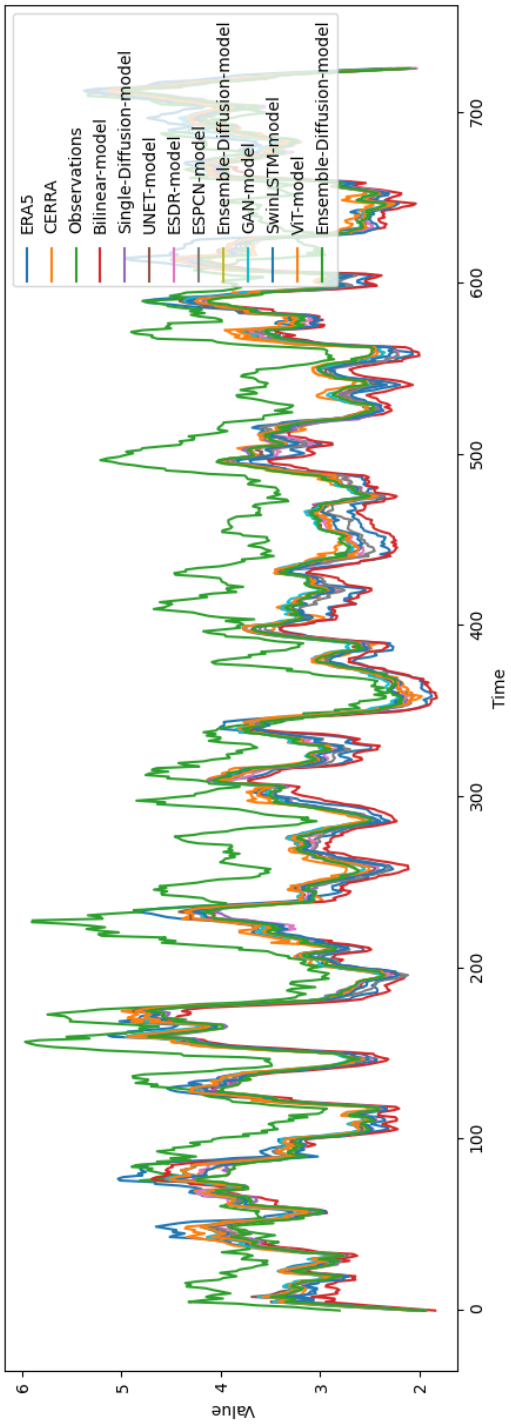


FIGURE 5.15: The figure represents the value of wind speed as given by all the models tested, the two reanalysis models ERA5 and CERRA, and a single meteorological station, over the entire year 2009.

In conclusion, a third analysis could be conducted: an in-depth geographical analysis of SSIM performance could provide insights into which areas are harder to effectively downscale. This is possible thanks to a characteristic of SSIM, which is calculated over patches, and so it is feasible to compile a comprehensive SSIM output. This output takes the form of a matrix, where each pixel value denotes the SSIM computed over a window centred on that corresponding pixel in the input images. Figure 5.16 presents mean images derived from averaging the full SSIM matrices for the entire testing years of 2009 and 2020, respectively. These images demonstrate that coastal regions exhibit higher error rates, with the most pronounced discrepancies observed in the high Adriatic Sea, the Strait of Messina, and the Ligurian Sea. The results also indicate that the geographical errors are remarkably similar across the two testing years, suggesting that there has been no significant change in the error distribution between these comparative periods.

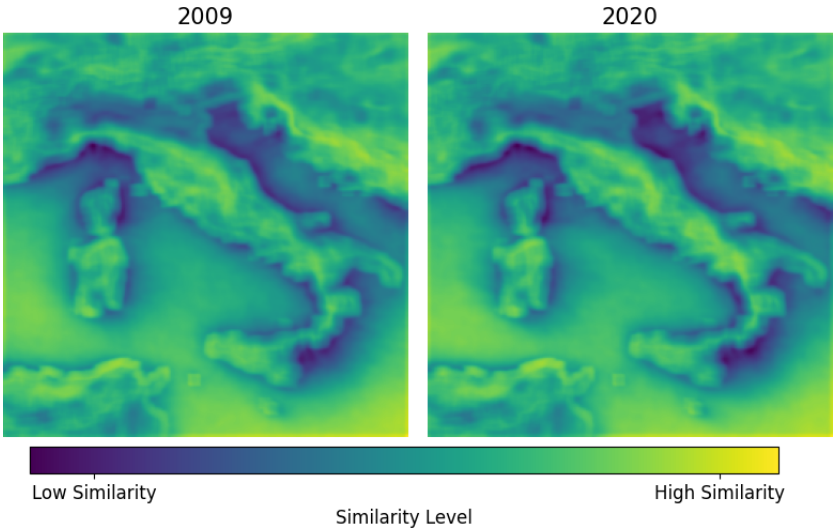


FIGURE 5.16: The image depicts the spatial SSIM error for both the 2009 and 2020 testing years. It illustrates the downscaling performance relative to different geographical areas. The similarity in the results indicates that the errors made by the model are consistent across the two years used for the test.

As a final experiment, given the unavailability of CERRA for the years ranging from 2021 to 2024, we also make publicly available for further studies our computed approximation of CERRA, computed via Ensemble Diffusion, including the complete code used for the experiments.

Chapter 6

Conclusions

This thesis has investigated the potential of advanced deep learning techniques for the super-resolution (SR) downscaling of climate data. The implementation and evaluation of several state-of-the-art models, including convolutional neural networks (CNN), generative adversarial networks (GANs), diffusion models, and vision transformers, has demonstrated the capacity of these methods to significantly enhance the spatial resolution of reanalysis datasets.

The results of the experiments, conducted over two distinct testing years, demonstrated that the downscaling computed by our method was in close alignment with the outcomes of traditional physics-based models. This alignment was evident in both the similarity of the results and their concordance with measurements from ground-based in-situ stations. Among the models tested, the Ensemble Diffusion approach demonstrated the most accurate results. Nevertheless, it is notable that even less complex super-resolution models consistently demonstrated superior performance compared to basic methods such as bilinear interpolation.

The results of this study illustrate the effectiveness of employing super-resolution deep neural networks in providing timely and informative downscaled data for meteorological research. In light of the considerable delays frequently encountered in downscaled reanalyses, for instance, CERRA's data lagging over two years behind the current date at the time of this work, our findings are of particular relevance to the scientific community. These delays typically arise from the unavailability of additional information needed by physics-based models, which depend on various sources and are subject to time-consuming computational processes. Our approach, which leverages a neural model trained on previously generated data, offers a timely alternative for accessing current downscaled results with a method that not only reduces computational costs but also maintains a high quality of output. It is of significant importance to note that the model in question produces data that is more closely aligned with actual measurements than the original low-resolution data, thereby demonstrating the practical potential of the

model in enhancing real-time weather analysis.

A further significant contribution arises from the disparate temporal scales between the high-resolution dataset and the low-resolution dataset. In fact, ERA5 offers hourly temporal resolution, whereas CERRA provides data at a 3-hour resolution. As our model is capable of generating sequences conditioned solely on ERA5 data, it can be employed to enhance the temporal resolution of the downscaled data, aligning it with that of ERA5.

It is important to note that this data-driven approach is limited by the necessity of pre-existing downscaled data for training, which relies on already established models. Furthermore, the experiments conducted in this study represent a preliminary phase, focusing solely on a single weather variable and a relatively limited geographical area. Future developments could encompass a broader range of weather variables and potentially cover the entire region addressed by CERRA. This more comprehensive understanding of the weather system could lead to improved results, as the model would gain access to a wider range of meteorological data. It is also important to note that the limited training dataset of "only" 10 years is noteworthy given the high quality of the results, but in future research, it would be beneficial to utilise a more extensive dataset, potentially spanning the full 40 years available from CERRA, to enhance the model's accuracy and predictive capabilities.

In conclusion, this thesis contributes to the field of climate data downscaling by demonstrating the potential of deep neural models, and in particular diffusion models, to generate high-resolution datasets that approximate the processes of traditional downscaling models. Furthermore, our findings suggest a promising future for reanalysis models, potentially shifting towards neural network-based approaches that leverage the advantages identified in this study. This shift could represent a significant advancement in the field, offering more efficient and accurate ways to handle meteorological data, ultimately contributing to better climate monitoring and prediction.

Bibliography

[1] Ali Salem Al-Sakkaf et al. “Assessing exposure to climate extremes over the Arabian Peninsula using ERA5 reanalysis data: Spatial distribution and temporal trends”. In: *Atmospheric Research* (2024), p. 107224.

[2] Saeed Anwar, Salman Khan, and Nick Barnes. “A Deep Journey into Super-Resolution: A Survey”. In: *ACM Comput. Surv.* 53.3 (2020). ISSN: 0360-0300.

[3] Andrea Asperti et al. “Precipitation nowcasting with generative diffusion models”. In: *arXiv preprint arXiv:2308.06733* (2023).

[4] Vivek Bannore. “Introduction to Super-Resolution”. In: *Iterative-Interpolation Super-Resolution Image Reconstruction: A Computationally Efficient Technique*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–8.

[5] Jorge Baño-Medina, Rodrigo Manzanas, and José Manuel Gutiérrez. “Configuration and intercomparison of deep learning neural models for statistical downscaling”. In: *Geoscientific Model Development* 13.4 (2020), pp. 2109–2124.

[6] Syed Muhammad Arsalan Bashir et al. “A comprehensive review of deep learning-based single image super-resolution”. In: *PeerJ Comput. Sci.* 7 (2021), e621. DOI: 10.7717/PEERJ-CS.621. URL: https://doi.org/10.7717/peerj-cs.621.

[7] Lisa Bengtsson et al. “The HARMONIE–AROME model configuration in the ALADIN–HIRLAM NWP system”. In: *Monthly Weather Review* 145.5 (2017), pp. 1919–1935.

[8] Yiling Cai and François-Marie Bréon. “Wind power potential and intermittency issues in the context of climate change”. In: *Energy Conversion and Management* 240 (2021), p. 114276.

[9] Francesco Cavalleri et al. “Inter-comparison and validation of high-resolution surface air temperature reanalysis fields over Italy”. In: (2023).

[10] C. Chaudhuri and Colin Robertson. "CliGAN: A Structurally Sensitive Convolutional Neural Network Model for Statistical Downscaling of Precipitation from Multi-Model Ensembles". In: *Water* (2020).

[11] Liuyi Chen et al. "Machine learning methods in weather and climate applications: A survey". In: *Applied Sciences* 13.21 (2023), p. 12019.

[12] Chia-Hao Chiang et al. "Climate Downscaling: A Deep-Learning Based Super-resolution Model of Precipitation Data with Attention Block and Skip Connections". In: *arXiv preprint arXiv:2403.17847* (2024).

[13] Francesco Chiaravalloti, Tommaso Caloiero, and Roberto Coscarelli. "The long-term ERA5 data series for trend analysis of rainfall in Italy". In: *Hydrology* 9.2 (2022), p. 18.

[14] Copernicus Climate Change Service (C3S). *ERA5 hourly data on single levels from 1940 to present*. Accessed on 22-11-2023. 2023.

[15] *Copernicus regional reanalysis for Europe (CERRA)*. URL: `https://climate.copernicus.eu/copernicus-regional-reanalysis-europe-cerra`.

[16] Tianyun Dong and Wenjie Dong. "Evaluation of extreme precipitation over Asia in CMIP6 models". In: *Climate Dynamics* 57.7-8 (2021), pp. 1751–1769.

[17] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: `https://openreview.net/forum?id=YicbFdNTTy`.

[18] Job CM Dullaart et al. "Advancing global storm surge modelling using the new ERA5 climate reanalysis". In: *Climate Dynamics* 54 (2020), pp. 1007–1021.

[19] Imke Durre et al. *Integrated Global Radiosonde Archive (IGRA), Version 2*. NOAA National Centers for Environmental Information. 2016.

[20] Marie Ekström, Michael R Grose, and Penny H Whetton. "An appraisal of downscaling methods used in climate change research". In: *Wiley Interdisciplinary Reviews: Climate Change* 6.3 (2015), pp. 301–319.

[21] Elissavet Galanaki et al. "Validating the Copernicus European Regional Reanalysis (CERRA) Dataset for Human-Biometeorological Applications". In: *Environmental Sciences Proceedings* 26.1 (2023), p. 111.

[22] Subhrendu Gangopadhyay, Martyn Clark, and Balaji Rajagopalan. "Statistical downscaling using K-nearest neighbors". In: *Water Resources Research* 41.2 (2005).

[23] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[24] Xiaogang He et al. "Spatial downscaling of precipitation using adaptable random forests". In: *Water resources research* 52.10 (2016), pp. 8217–8237.

[25] Hans Hersbach et al. "The ERA5 global reanalysis". In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049.

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[27] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[28] Alain Horé and Djemel Ziou. "Image Quality Metrics: PSNR vs. SSIM". In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2366–2369.

[29] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[30] Yan Ji et al. "Downscaling of Precipitation Forecasts Based on Single Image Super-Resolution". In: *EGU General Assembly Conference Abstracts*. 2020, p. 8533.

[31] Donglai Jiao et al. "Evaluation of spatial-temporal variation performance of ERA5 precipitation data in China". In: *Scientific Reports* 11 (2021).

[32] Michael I Jordan. "Serial order: A parallel distributed processing approach". In: *Advances in psychology*. Vol. 121. Elsevier, 1997, pp. 471–495.

[33] Mohammad Sajjad Khan, Paulin Coulibaly, and Yonas Dibike. "Uncertainty analysis of statistical downscaling methods". In: *Journal of Hydrology* 319.1-4 (2006), pp. 357–382.

[34] Rupa Kurinchi-Vendhan et al. "Wisosuper: Benchmarking super-resolution methods on wind and solar data". In: *arXiv preprint arXiv:2109.08770* (2021).

[35] David A Lavers et al. “An evaluation of ERA5 precipitation for climate monitoring”. In: *Quarterly Journal of the Royal Meteorological Society* 148.748 (2022), pp. 3152–3165.

[36] Yann LeCun, Yoshua Bengio, et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[37] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[38] Jangho Lee and Andrew E Dessler. “Improved Surface Urban Heat Impact Assessment Using GOES Satellite Data: A Comparative Study With ERA-5”. In: *Geophysical Research Letters* 51.1 (2024), e2023GL107364.

[39] Dawa Chyophel Lepcha et al. “Image super-resolution: A comprehensive review, recent trends, challenges and applications”. In: *Information Fusion* 91 (2023), pp. 230–260. ISSN: 1566-2535.

[40] Bee Lim et al. “Enhanced deep residual networks for single image super-resolution”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[41] Jiang Liu, Daniel Fiifi Tawia Hagan, and Yi Liu. “Global land surface temperature change (2003–2017) and its relationship with climate drivers: AIRS, MODIS, and ERA5-land based analysis”. In: *Remote Sensing* 13.1 (2020), p. 44.

[42] Ioannis Masloumidis et al. “Snow Depth Trends of European Ski Resorts”. In: *Environmental Sciences Proceedings* 26.1 (2023). ISSN: 2673-4931.

[43] Fabio Merizzi, Andrea Asperti, and Stefano Colamonaco. “Wind speed super-resolution and validation: from ERA5 to CERRA via diffusion models”. In: *arXiv preprint arXiv:2401.15469* (2024).

[44] S. Misra, S. Sarkar, and Pabitra Mitra. “Statistical downscaling of precipitation using long short-term memory recurrent neural networks”. In: *Theoretical and Applied Climatology* 134 (2018), pp. 1179–1196.

[45] Diego Monteiro and Samuel Morin. “Multi-decadal analysis of past winter temperature, precipitation and snow cover data in the European Alps from reanalyses, climate models and observational datasets”. In: *The Cryosphere* 17.8 (2023), pp. 3617–3660.

[46] Tung Nguyen et al. “Climatelearn: Benchmarking machine learning for weather and climate modeling”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[47] Gianfranco Nicodemo et al. “Remote Sensing Meteorological and DInSAR Historical Data to Analyse the Kinematic Behaviour of Slow-Moving Landslides at Municipal Scale”. In: *National Conference of the Researchers of Geotechnical Engineering*. Springer. 2023, pp. 242–250.

[48] Jon Olauson. “ERA5: The new champion of wind power modelling?” In: *Renewable energy* 126 (2018), pp. 322–331.

[49] A. Pelosi. “Performance of the Copernicus European Regional Reanalysis (CERRA) dataset as proxy of ground-based agrometeorological data”. In: *Agricultural Water Management* 289 (2023), p. 108556. ISSN: 0378-3774.

[50] Quoc Bao Pham et al. “Combing random forest and least square support vector regression for improving extreme rainfall downscaling”. In: *Water* 11.3 (2019), p. 451.

[51] Sahar Hadi Pour, Shamsuddin Shahid, and Eun-Sung Chung. “A hybrid model for statistical downscaling of daily rainfall”. In: *Procedia Engineering* 154 (2016), pp. 1424–1430.

[52] M. Rolle, S. Tamea, and P. Claps. “Improved large-scale crop water requirement estimation through new high-resolution reanalysis dataset”. In: (2020), p. 19289.

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[54] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, ed. DE Rumelhart and J. McClelland. Vol. 1. 1986”. In: *Biometrika* 71 (1986), pp. 599–607.

[55] S. Schimanke et al. “CERRA sub-daily regional reanalysis data for Europe on single levels from 1984 to present”. In: *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)* (2021). Accessed on 22-11-2023.

[56] Uwe Schulzweida. *CDO User Guide*. Version 2.3.0. Oct. 2023.

[57] Yingkai Sha et al. “Deep-Learning-Based Gridded Downscaling of Surface Meteorological Variables in Complex Terrain. Part II: Daily Precipitation”. In: *Journal of Applied Meteorology and Climatology* (2020).

[58] Liucheng Shen et al. “Changes in population exposure to extreme precipitation in the Yangtze River Delta, China”. In: *Climate Services* 27 (2022), p. 100317.

[59] Wenzhe Shi et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883.

[60] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: JMLR Workshop and Conference Proceedings 37 (2015), pp. 2256–2265. URL: http://proceedings.mlr.press/v37/sohl-dickstein15.html.

[61] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=St1giarCHLP.

[62] Song Tang et al. “Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 13470–13479.

[63] M. Tarek, F. Brissette, and R. Arsenault. “Evaluation of the ERA5 reanalysis as a potential reference dataset for hydrological modelling over North America”. In: *Hydrology and Earth System Sciences* 24 (2020), pp. 2527–2544.

[64] P. Termonia et al. “The ALADIN System and its canonical model configurations AROME CY41T1 and ALARO CY40T1”. In: *Geoscientific Model Development* 11.1 (2018), pp. 257–281.

[65] Ruian Tie et al. “CLDASSD: reconstructing fine textures of the temperature field using super-resolution technology”. In: *Advances in Atmospheric Sciences* 39.1 (2022), pp. 117–130.

[66] Thomas Vandal et al. “DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM,

2017, pp. 1663–1672. DOI: 10.1145/3097983.3098004. URL: https://doi.org/10.1145/3097983.3098004.

[67] Thomas Vandal et al. "Generating high resolution climate change projections through single image super-resolution: An abridged version". In: *International joint conferences on artificial intelligence organization*. 2018.

[68] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[69] A. Verrelle et al. "CERRA-Land : A new land surface reanalysis at 5.5 km resolution over Europe". In: (2021).

[70] Wei Wei et al. "Spatiotemporal variability in extreme precipitation and associated large-scale climate mechanisms in Central Asia from 1950 to 2019". In: *Journal of Hydrology* 620 (2023), p. 129417.

[71] Robert L Wilby and Christian W Dawson. "The statistical downscaling model: insights from one decade of application". In: *International Journal of Climatology* 33.7 (2013), pp. 1707–1719.

[72] Meric Yilmaz. "Accuracy assessment of temperature trends from ERA5 and ERA5-Land". In: *Science of The Total Environment* 856 (2023), p. 159182.

[73] Peng Yuan et al. "Feasibility of ERA5 integrated water vapor trends for climate change analysis in continental Europe: An evaluation with GPS (1994–2019) by considering statistical significance". In: *Remote Sensing of Environment* 260 (2021), p. 112416.

[74] Ruqing Zhang et al. "Assessment of Agricultural Drought Using Soil Water Deficit Index Based on ERA5-Land Soil Moisture Data in Four Southern Provinces of China". In: *Agriculture* 11 (2021), p. 411.

[75] Shuangyi Zhang and Xichen Li. "Future projections of offshore wind energy resources in China using CMIP6 simulations and a deep learning-based downscaling method". In: *Energy* 217 (2021), p. 119321.