Development of 300-m gridded digital twins of precipitation over Delhi for 1980-2020

A Thesis

submitted to

Indian Institute of Science Education and Research Pune in partial fulfillment of the requirements for the BS-MS Dual Degree Programme

by

Vishal Choudhary

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May 2024

Supervisor: Dr Manmeet Singh(IITM Pune)

Certificate

This is to certify that this dissertation, **“Development of 300-m gridded digital twins of precipitation over Delhi for 1980-2020”** towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents the study/work carried out by Vishal Choudhary at Indian Institute of Science Education and Research under the supervision of Dr Manmeet Singh, Indian Institute Of Tropical Meteorology, Pune, during the academic year 2023-2024.

Manmeet Singh

Dr Manmeet Singh

Committee:

Supervisor: Dr Manmeet Singh (IITM Pune)

TAC Expert: Dr Joy Merwin Monterio (IISER Pune)

This thesis is dedicated to my family

Declaration

I hereby declare that the matter embodied in the report **entitled "Development of 300-m gridded digital twins of precipitation over Delhi for 1980-2020"** are the results of the work carried out by me at the Indian Institute of Tropical Meteorology, Pune, under the supervision of **Dr Manmeet Singh**. The same has not been submitted elsewhere for any other degree. Wherever others contribute, every effort is made to indicate this clearly, with due reference to the literature and acknowledgment of collaborative research and discussions.

Vishal

Vishal Choudhary

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr Manmeet Singh for his support, guidance and scholarly insights throughout the entire process of my thesis. I thank my TAC Expert, Dr. Joy Merwin Monterio for his constructive feedback, critical evaluation and suggestions that significantly enriched my work. I am deeply grateful to my Father, Mother, and Brother for their unconditional love and support. I thank and appreciate IISER Pune for providing the necessary resource, facilities, and opportunities that facilitated the completion of the thesis.

Abstract

The impacts of climate change are felt by most critical systems, such as infrastructure, ecological systems, and power plants. However, contemporary Earth System Models (ESM) are run at spatial resolutions too coarse for assessing effects this localized. High-resolution datasets are required for the planning, adaptation and furthering of urban climate science(NEELESH, 2022). Although there has been tremendous growth in climate science and weather forecasting in general, the development of gridded datasets of the order of sub 300 m or less than 500 m gridded scale is still challenging (DAVID, 2019). Deep learning has proven to be a potent tool in deciphering nonlinear mappings. It can be used as a powerful technology to develop high-resolution products from coarse-resolution available datasets (MARKUS, 2019). Here, we use Deep learning models like SRCNN(super-resolution convolutional neural network) and GAN(Generative Adversarial Network to try to find a solution to this problem.

Contents

Table of Figures

Chapter1

Introduction

Cities need climate information to develop the infrastructure and to adapt the information. We need a high-resolution dataset to understand urban climate. The spatial resolution tells us how detailed or representative the particular station\image is and the context of city planning. High-resolution datasets can provide many attributes and help city planners make better decisions and strategies. For example, urban climate data with high Resolution can be a good measure to understand the areas that flood or will have more heat. In addition, such information is necessary to develop high-quality infrastructure as part of a smart city. The desired information is in order of finer scales than what is available from climate analysis and future projections. (Kumar, 2021)

Downscaling approaches are categorized under two themes: statistical and dynamical. Dynamical downscaling uses a high-resolution numerical weather prediction model to simulate the weather over smaller domains at fine spacing and scaling. However, Dynamic downscaling is computationally intensive, and statistical downscaling is often used to improve spatial resolution when reference baseline data is available. It still depends on high-quality historical climate data, and there are always chances that it may not be able to capture the effects of climate change. More sophisticated approaches like Support Vector Machines, Probabilistic Global Search, and Artificial Intelligence have become available recently. These approaches have gained popularity in recent years because of their low computational need and relatively quickness compared to dynamical downscaling. (vandal, 2017)

Several studies conducted by Dong et al.(2015) have shown that a simple, lightweight convolutional neural network (SRCNN) can substantially outperform a widely used technique of dimensional bicubic interpolation for spatial downscaling. In this study, our interest is to develop high-resolution climate information or datasets over some cities of Delhi. Past research for generating high-resolution datasets focused on machine learning methods such as kriging, random forests, and support vector machines. However, recent approaches are more focused on Deep Learning. While most of the studies and research attempt to improve the urban dataset's spatial resolution, they do not employ Deep Learning for high-resolution (<500m) urban precipitation Downscaling. Also, past research has primarily focused on air pollution and temperature-related variables. Still, none of them have hardly focused on the generation of the High-Resolution Precipitation gridded dataset. (manmeet, 2022) High-resolution precipitation maps over a particular region, especially urban areas that are non-continuous and dynamic, are not popularly assessed by studies. The heterogeneous environment of urban regions has varying physical and thermodynamic properties that alter the flux of surface and atmospheric boundary layers, which ultimately translates into a shift in urban regime over the urban landscape. We expect that urban precipitation can be downscaled in very high resolution. Our model is expected to generate a fast high spatiotemporal rainfall or any other meteorological dataset with Super-Resolution.

This study uses CNN-based models SRCNN and SRGAN to downscale precipitation data. We have created a 300-meter gridded dataset by doing a sequence of operations and then unifying coarse resolution, High Resolution, and source station data in a single NetCDF file. (Bipin, 2023). In this study, we move away from all the previous efforts of downscaling and try to create 300m precipitation data with the help of randomly distributed observations of rainfall sequenced by date and time, which we call the Stations data, which is officially called Global Historical climatological network Daily (GHCND) which is directly obtained from the various observational land surface stations around a particular location(latitude-longitude) and CHIPRS 5km gridded data by Climate Hazard Center UC Santa Barbara. We accomplished this with the help of running an iterative SRCNN unifying and recovering that CHIRPS data as a 300m gridded image compilation. In this pursuit, we ignored all those inconsistent stations that did have long-term data over time. Then, we created and employed SRGAN-inspired architecture called MeteoGAN, which includes a Generator, a Discriminator, and a Target. Here, GHCND station data is used as a target dataset. The whole structure resembles architecture similar to SRGAN, which includes a generator and discriminator going forward and backward propagation and then trained all of this to generate a 300m precipitation dataset.

If we look at the previous research done by various organizations on generating High-resolution datasets or Downscaling high-resolution data from a particular available data, anything around 500m resolution is a good and ambitious goal. We tried it for a 300m gridded resolution and felt comfortable dealing with it. There is no particular reason why we chose this specific number of 300m, but in the future, we plan to work for a better resolution than 300m.

Chapter – 2

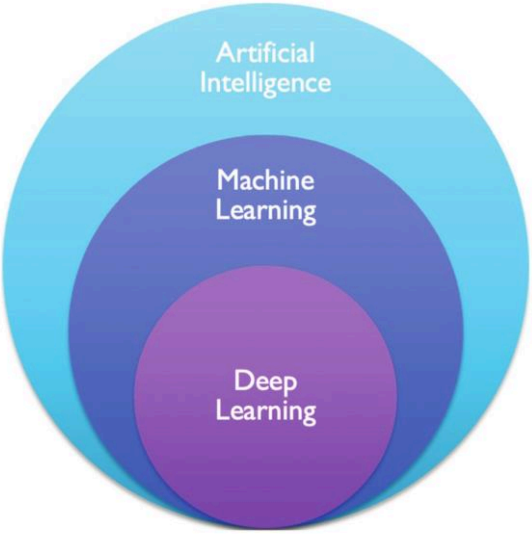
Theoretical background:

2.1 Artificial Intelligence:

Artificial intelligence is the intelligence of the software or a machine. It is a field of computer science that focuses on and develops intelligent systems. These machines are sometimes called AI.

This technology has many applications in agriculture, government, science, medicine, and other industries.

Alan Turing was the scientist who conducted any kind of research on intelligence-related machines. This research field has endured many decades of dormancy regarding funding and interest. Although it started as an effort to understand or mimic the human brain, it eventually went on a different path, which had the potential to become commercialized because of its vast industrial application.



.

Figure 1: AI, Machine Learning, Deep Learning

The funding and interest primarily increased after the coming of deep learning in 2012 when it surpassed all the previous Artificial intelligence techniques and then transformer architecture, which led to a boom in the overwhelming interest shown by companies, universities, and laboratories.

The growing use of artificial intelligence systems today is an effect of an economic shift towards automation and data-driven systems.

2.2 Machine learning:

Machine learning is a computer science and artificial intelligence subfield that uses data and algorithms to mimic how people learn. It is divided into two subcategories: supervised and unsupervised learning. In the former, a labeled dataset is used to train algorithms to classify or predict outcomes. At the same time, the latter contains unstructured data that can be analyzed on clustering using machine learning algorithms. The most common supervised learning task includes regression and classification. Dimensionality reduction and anomaly reduction can be made using unsupervised learning.

The machine learning models input the data and generate the predictions, which can be tested using metrics like RMSE in regression, accuracy in classification, etc.

If the predictions do not match, the model is repeatedly trained to minimize the defined cost function to improve itself.

There are two types of machine learning: supervised and unsupervised.

(a) Supervised learning:

When the input data is paired with the desired output and the machine is trained on a labeled data set, the machine will learn to predict the output for new input data. Supervised learning is frequently utilized for tasks like object detection, regression, and classification.

(b) Unsupervised learning:

The system is trained on a collection of unlabeled data in unsupervised learning, which implies that the input data does not match the intended output.
After that, the machine picks up on the relationships and patterns in the data.
Unsupervised learning is frequently applied to anomaly detection, dimensionality reduction, and grouping.

(c) Reinforcement Learning:

Through trial and error, an agent learns to make the best judgments possible in an environment through the intriguing field of reinforcement learning (RL) in machine learning. Consider an agent that is navigating a maze. Reinforcement learning (RL) enables it to learn by interacting with the environment, earning rewards for good deeds and punishments for bad ones. Like humans, the agent learns from its experiences and adjusts its plan over time to maximize its cumulative reward. This method enables RL models to handle complex tasks in various fields, including robotics, gaming, healthcare, and finance, where defining a precise set of instructions may be challenging. The reward function, which directs the agent's learning process, must be carefully designed for reinforcement learning (RL) algorithms, which can be computationally costly.

2.2.1 Linear regression:

Linear regression comes under the supervised machine learning technique. The model finds its best fit and captures the linear relationship between the dependent and independent variables. Let $X1, X2 \ldots\ldots Xn$ be the independent variables in our dataset and corresponding, each has labeled dependent feature Y, so our task is to find the best fit linear line which maps from X to Y. The linear equation has weight b1, b2,…….bn and bias b0 to get the least MSE.

$$Y = b_0 + b_1X_1 + b_2X_2 + \ldots\ldots\ldots\ b_nX_n \qquad (2.1)$$

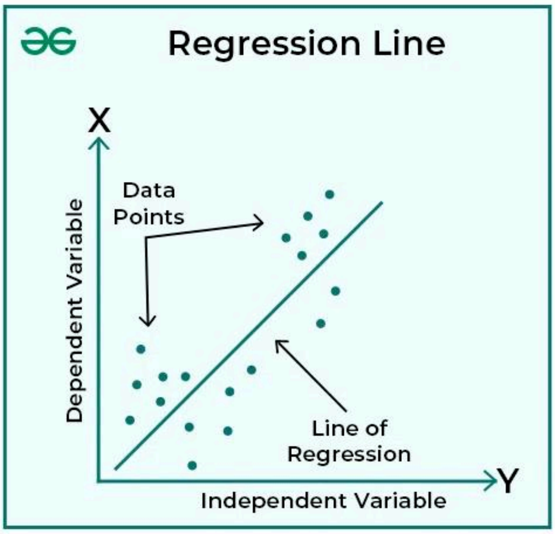


Figure 2:Linear Regression (gfg, GFG)

The mean squared error is used as a cost function to adjust the weights and the biases. Here y_i is the ground truth value, and $\hat{y}_i$ is the predicted one.

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

(2.3)

The above cost function is minimized using some optimization algorithm. Since this is the cost function with only one minimum, we can reduce the gradient descent algorithm to reduce the loss.

To check the performance, the model can be further evaluated using metrics like MSe, RMSE,   etc..

$$R^2 = 1 - \frac{\Sigma_i (y_i - \hat{y}_i)^2}{\Sigma_i \left(y_i - \underline{y}_i\right)^2} \qquad (2.3)$$

2.2.2 Gradient Decent:

It is used to find the local maxima of a differentiable function. Imagine the above model (2.1) for only one independent variable (n=1). So, we have $Y_{pred} = b_0 + b_1 X$ the predicted output, which is interpreted, the slope, and the input sample here. The task is to reduce the cost function (2.2) using some optimization. This can be achieved using the gradient descent algorithm on J(w,b), where J is the cost function. Then, some weights b_1 *and* b_0 are updated using the steps below, where α the learning rate generally lies between 0 and 1. The smaller the learning rate, the slower the step will be, and the bigger the learning rate, the larger the step will be.

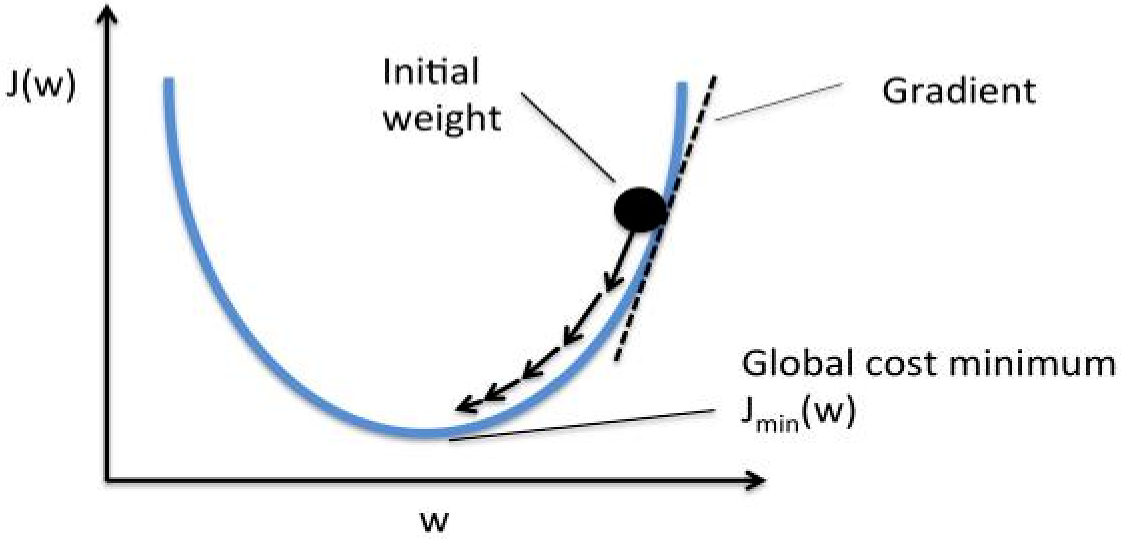


Figure 3:Grdient Decent (analyticsarora, 2023)

Repeat until convergence.

The model is trained on biases found for the minimum of the cost function. The figure below shows how the algorithm works.

2.2.3 XGBoost:

A tree-based machine learning model called gradient boosting uses gradient descent for optimization.

When extended, it is called extreme gradient descent, which can avoid overfitting and have higher performance and accuracy.

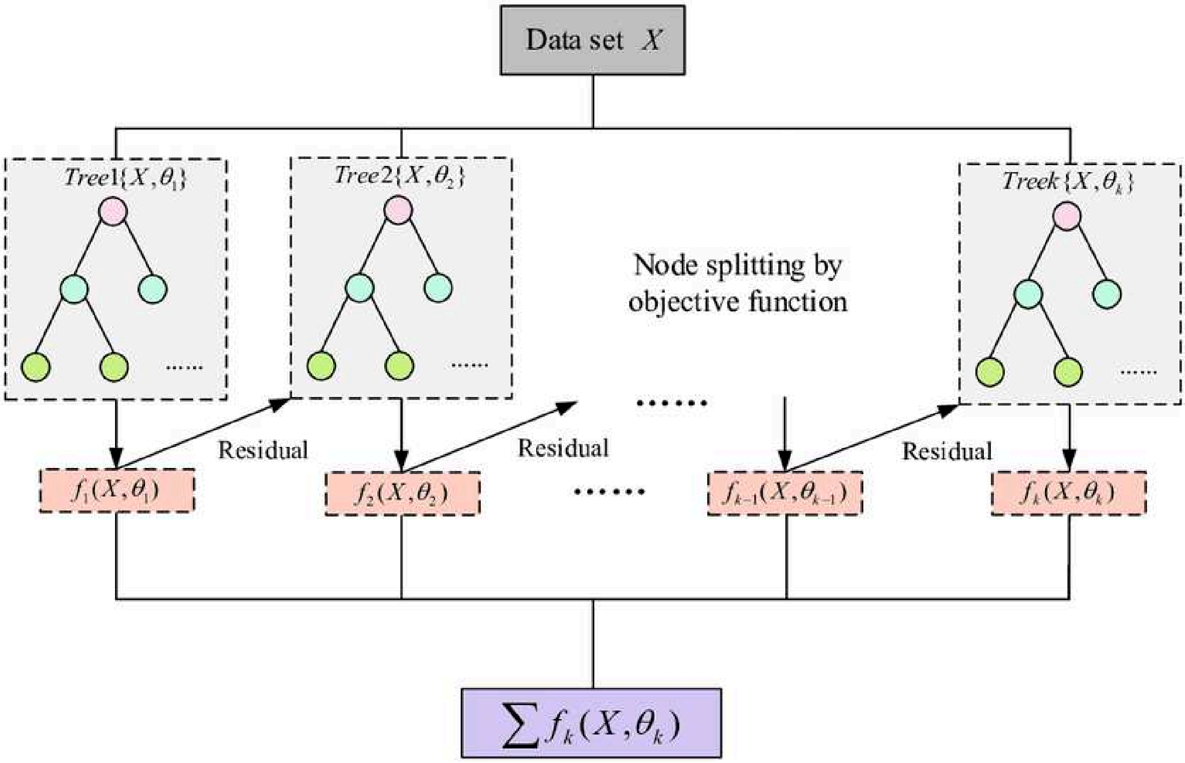


Figure 4:Flowchart of XGBoot (et.al G. , 2020

2.3 Deep Learning:

Artificial neural networks are the foundation of the machine learning subfield known as deep learning. It can recognize intricate links and patterns in data. We don't have to program everything in deep learning explicitly. Because of the availability of enormous datasets and the advancements in computing power, it has grown in popularity in recent years because deep neural networks, or artificial neural networks (ANNs), are its foundation (DNNs). Multilayer artificial neural networks are used to handle and analyze data. It does this by constructing multi-layered models, where each layer pulls out more abstract elements from the input. This method has transformed domains such as object detection, image analysis, and speech recognition. Deep learning even affects scientific discovery outside these domains, such as genetics and drug development. Backpropagation is a method used in deep learning. Through error propagation analysis, this approach assists the model in optimizing its internal parameters. The model learns to better precisely represent data across each layer by modifying these parameters. Various varieties of deep learning architectures are better at particular tasks. Convolutional neural networks, or ConvNets, are highly skilled at processing audio and visual input, including speech, movies, and images. Conversely, recurrent neural networks, or RNets, are particularly good at processing sequential input, such as spoken language and text.

2.3.1 Deep Neural Networks:

Artificial neural networks are the foundation of deep learning (ANNs). The biological neurons found in the brain serve as a loose model for these ANNs. They are made up of layers of connected nodes or artificial neurons.

Data is fed into the network's input layer, from where it moves through hidden layers to the output layer, where it is finally output. As the data moves through the network, each layer applies particular mathematical operations to extract higher-level properties. The word "deep" describes several hidden layers that enable the network to decipher intricate patterns from the input.

Benefits and Uses:
The power of deep learning resides in its capacity to automatically extract intricate patterns from data, frequently without the need for explicit feature engineering, which involves working with humans to describe the characteristics the model should recognize. This qualifies it for jobs like

***Image Recognition**:* Deep learning is excellent at identifying objects and situations in photographs. Examples of applications for this type of work include traffic sign identification by self-driving cars and smartphone unlocking by facial recognition systems.

Natural Language Processing (NLP)**:** Deep learning is the engine behind applications such as text sentiment analysis, machine translation, and chatbots that can comprehend and converse in human languages.

Speech Recognition: Voice assistants such as Siri or Alexa may interpret your commands by using deep learning algorithms that translate spoken language into text.

2.3.2 Artificial Neural Network:

Neuron is the most fundamental unit of any deep learning model. After receiving one or more inputs, it sums up after giving up some weights and biases randomly. Then, it applies nonlinear transformation using some activation function and produces the output. The errors are compared using some cost function like MSE. The gradient updates the weights decently during backpropagation.

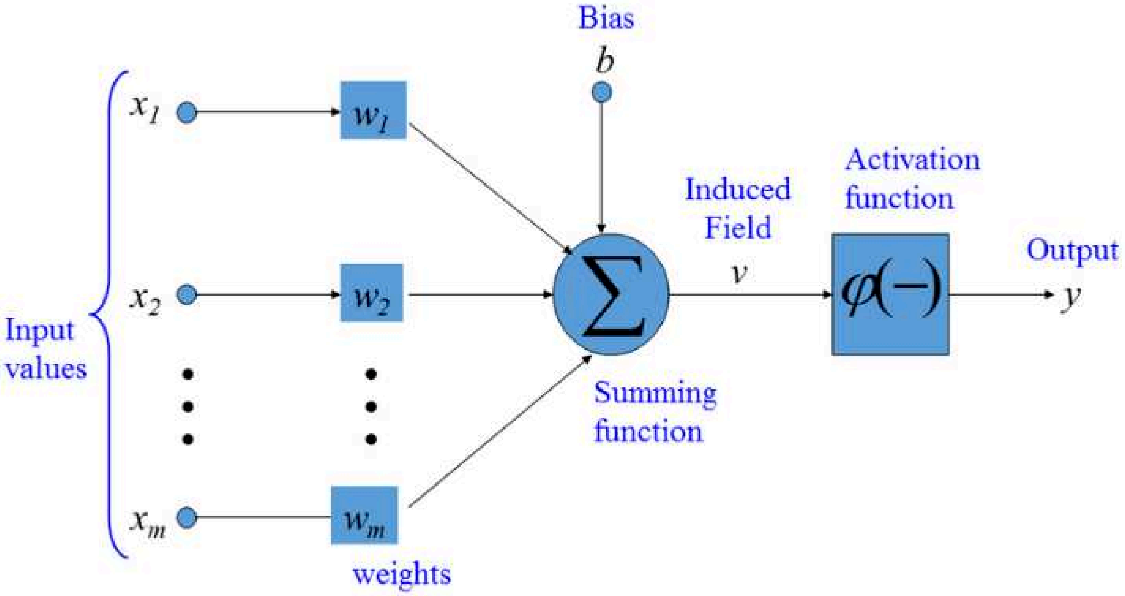


Figure 5:Artificial Nueron (Gabormelli, 2024)

More such neurons are added layer-by-layer to extract features from the inputs. Finally, all these layers are connected to the output layer. And then, we see the whole creation of an artificial neural network.

Because of the many parameters, these neural networks require much training time. We call it a Deep Neural Network if it has more layers. As we can see in the figure, the number of neurons are connected, and a hidden layer is attached. The neurons use activation functions to learn non-linearity in data.

Some joint activation functions are sigmoid, ReLU, etc.

The sigmoid activation function scales the values between the range[0,1]. It is represented by $\sigma(x)$ as given in the equation (2.6)

$$\sigma(x) = \frac{e^x}{1+e^x} \qquad (2.6)$$

Rectified Linear Unit (ReLU) is used to scale the values in the range[0,∞). For values less than zero, it replaces them with zero and the same for values greater than zero.

$$g(x) = \max(0,x) \qquad (2.7)$$

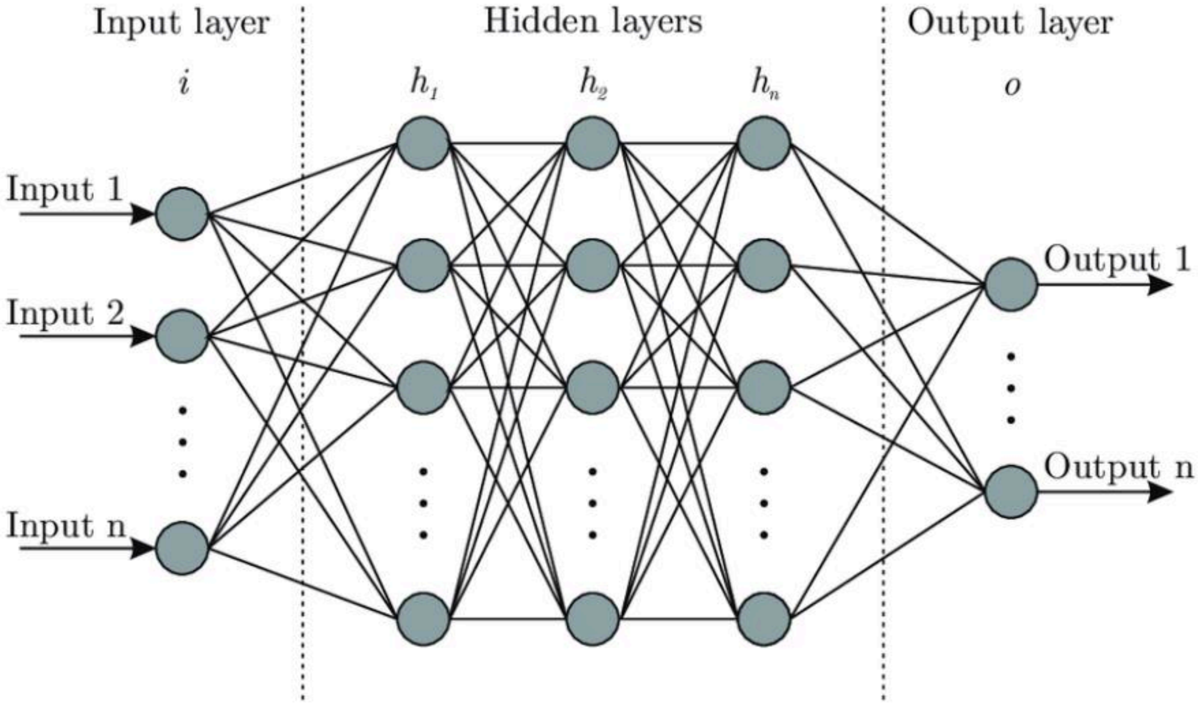
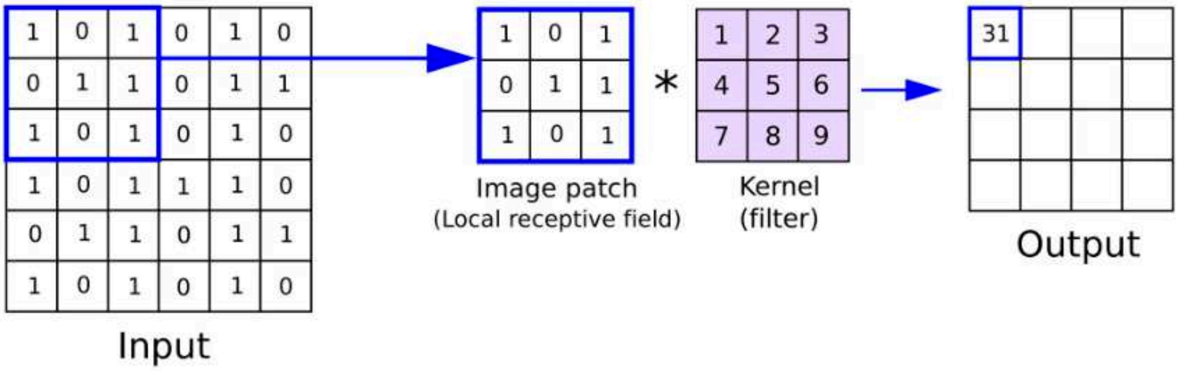


Figure 6:Deep Neural Network (Bre, 2017)

The Neural network uses the cost function to calculate the difference between predictions of the network and actual truth, and the model learns by minimizing the cost function. Generally used cost functions include 'mean squared error,' 'cross-entropy,' 'mean absolute error,' etc. During the forward propagation, it generates some output, and the error gets backpropagated such that weights and biases are adjusted using optimizers. The training data is usually provided in batches. Finally, the model can be evaluated using metrics like 'RMSE' for regression or 'accuracy' for classification problems.

2.2.3 Convolutional Neural Network:

The complexity of ANN increases with an increase in parameters and image size. Then, convolutional neural networks were introduced, which can extract relevant features from the images and reduce the parameter number to learn from the data.
The convolutional is passing a filter over the image to extract the features.
The dimensionality and size of the filter can be adjusted. The greyscale images use Conv2D, and the RGB channel uses Conv2D and Conv3D. 'Same' and 'Valid' are two common types of padding are applied. In contrast, the input array size is smaller when the value is 'valid.'



1	0	1	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	0	1	1	1	0
0	1	1	0	1	1
1	0	1	0	1	0

1	0	1
0	1	1
1	0	1

1	2	3
4	5	6
7	8	9

31			

Figure 7:Convoltion operation (superannote, 2024)

Spatial size of the image is decreased by applying additional procedures like "max pooling" and "average pooling."

The maximum value of the items in the pooling window is taken in the max pooling operation. All the components within the pooling window are averaged in average pooling. This is very useful when dealing with large-sized photos, where the loss of information won't be substantial.

2.2.4 Recurrent Neural Network:

Artificial Neural networks are not productive when it comes to processing sequential information like time series, natural language, and speech; therefore, recurrent neural network is created.

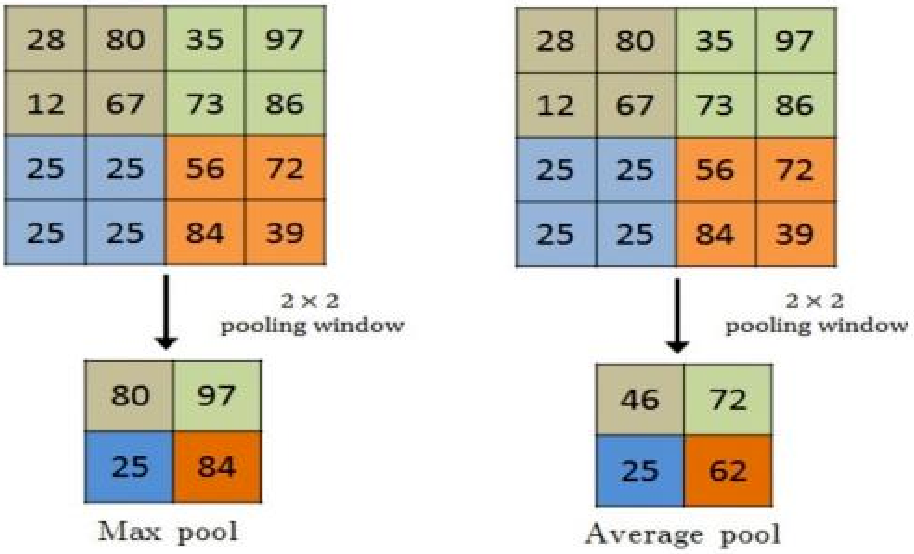


Figure 8:max pooling and avg. pooling (et.al Y. , 2019)

It can process input sequences of arbitrary length sequentially. For the network to learn the sequential information, the output of the preceding cell—referred to as the hidden state—is sent on with the subsequent input. The figure shows the unrolled ’RNN’ that takes input Xt inputs, producing a hidden state ht at each timestep passed on with the following input in the sequence. In this way, the network learns to retain the sequential information of previous timesteps.

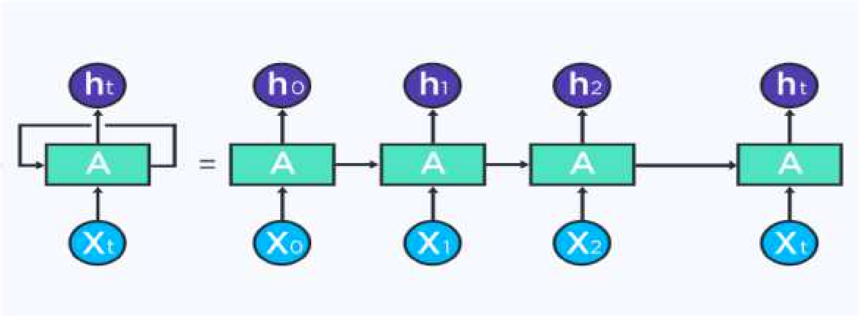


Figure 9:unrolled recurrent neural network

2.2.5 Image Super Resolution:

Image super-resolution aims to take this lower-quality image and turn it into a high-resolution one. It entails filling in the blanks and producing a crisper, more informative image.

2.2.5.1 Deep CNN for image Super-Resolution:

Because deep CNNs can learn intricate, non-linear correlations between low-resolution (LR) and high-resolution (HR) picture patches, they are an effective tool for image SR.

The input layer receives the LR image, and then the convolutional layers use learnable filters to extract features from the LR image. At various abstraction levels, these features catch essential information and patterns. Upsampling Layer enlarges the feature maps spatial dimensions to correspond with the required HR output size. There are various upsampling methods that can be used, such as transposed convolutions, bilinear interpolation, and nearest neighbor. Reconstruction Layer creates the final HR image and processes the upsampled features more. This layer could use extra convolutions or other processes to improve detail and eliminate them. The output layer produces the reconstructed HR image.

2.2.5.1 SRGAN:

A deep learning model called *SRGAN (Super-Resolution Generative Adversarial Network)* is made especially for super-resolution images. Taking a low-resolution (LR) image, it attempts to produce a high-resolution (HR) version almost identical to the original high-resolution image from which the LR image may have been taken.

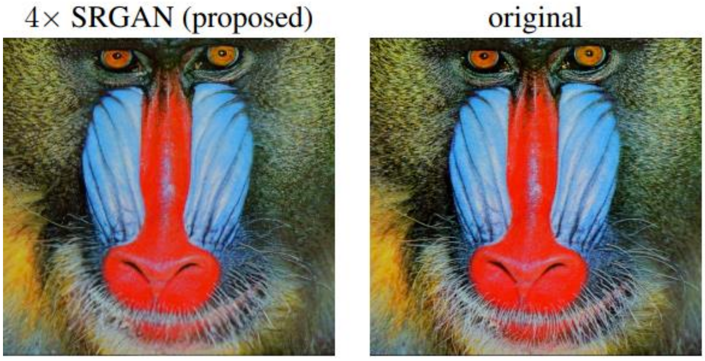


Figure 10:super-resolution image in left (al, 2018)

The idea of Generative Adversarial Networks (GANs) contains two neural networks competing with each other. They are called Generator and Discriminator.

Generator: Given a low-resolution input, this network seeks to produce realistic, high-resolution visuals.

Discriminator: This network serves as a critic, attempting to discern between the false HR images produced by the generator and the actual HR images found in the training dataset.

The generator gradually gains the ability to produce increasingly intricate and realistic HR images that can trick the discriminator through this continual competition. SRGAN is designed with super-resolution tasks in mind. SRGAN uses low-resolution inputs to generate realistic-looking, high-quality images with improved details. (HUANG, 2022)

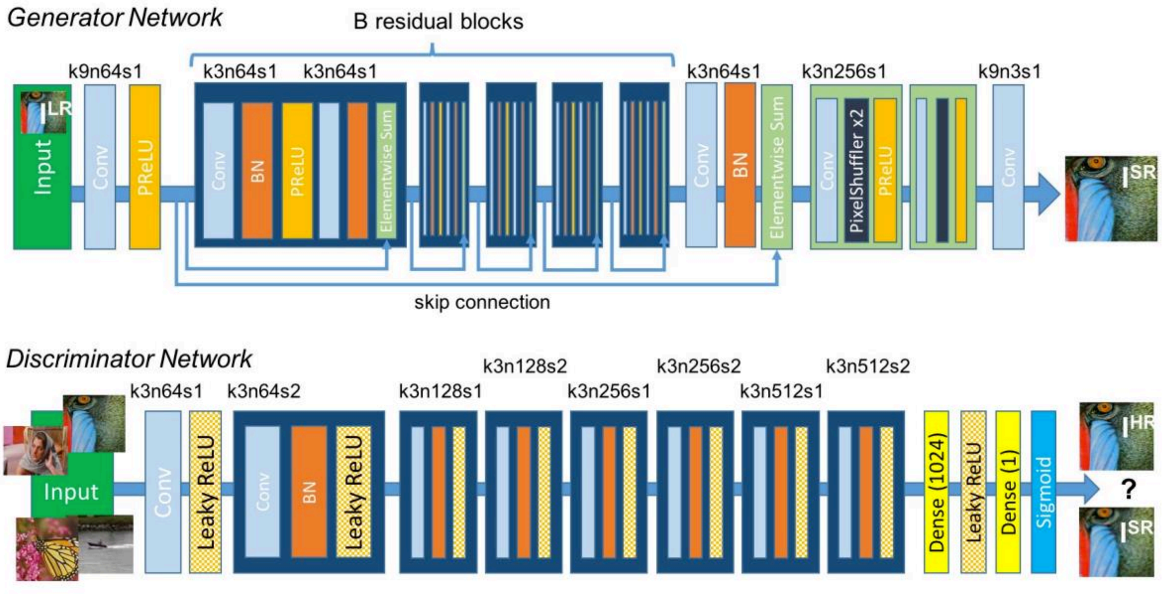


Figure 11:Architecture of SRGAN (Dong, 2014)

SRGAN implements a perceptual loss based on features extracted from VGGNet, a pre-trained deep convolutional neural network (CNN). This loss function appraises the perceptual likeness between generated high-resolution images and their ground truth. (ENTHALPPY, n.d.).The incorporation of the perceptual losses ensures that SRGAN not only produces high-fidelity images but also captures crucial visual characteristics present in the reference images. SRGAN usually involves other architectural features or methods to enhance the quality of generated images, such as having residual blocks. In particular, SRGAN may use residual blocks to make learning the residual mappings between low-resolution and high-resolution pictures easier. On top of this, generators often adopt methods such as pixel shuffle or sub-pixel convolution to upsample low-resolution feature maps to an expected high-resolution output.

Bicubic Interpolation:

Interpolation is a vital technology in the field of computer graphics and image processing that improves the quality of digital images. Among sophisticated methods, cubic interpolation is noteworthy for its ability to approximate values between known data points by using mathematical rigour. It entails calculating a function's value at intervals between known data points. This extrapolation requires a technique that maintains key aspects of the original data while also guaranteeing smoothness. The fundamental idea behind bicubic interpolation is cubic interpolation, which is based on polynomial approximation. Cubic interpolation ensures continuity in both function and derivative values by building a cubic polynomial that goes through each data point given to it. Let us denote the known data points as (x_i, y_i), where i = 1, 2, ..., n. The cubic polynomial can be expressed as:

$$\boldsymbol{P(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3}$$

Where a_i, b_i, c_i and d_i coefficient specific to each interval $[x_i , x_{i+1}]$

By using cubic polynomials in both the horizontal and vertical directions, bicubic interpolation overcomes the drawbacks of linear and cubic interpolation. By improving the interpolated surface's smoothness, this method guarantees a more accurate depiction of the underlying data. Mathematically, the bicubic interpolation function is expressed as follows:

$$\boldsymbol{f(x,y) = \sum_{i=0}^{3} . \sum_{j=0}^{3} \mathbf{a_{ij}x^iy^i}}$$

f(*x*,*y*) represents the interpolated value at coordinates (x, y).
aij denotes the coefficients determined through a process known as convolution.

Convolution must be used in order to determine the coefficients in bicubic interpolation. Convolution is the process of integrating a kernel function across the whole original image domain. The generation of coefficients necessary for rebuilding the interpolated surface is made easier by this approach. Convolution using a bicubic kernel function yields the coefficients a_{ij}. This kernel function guarantees that the resultant surface retains the key elements of the original data because of its compact support and interpolation properties. Bicubic interpolation has advantage that it not only produces better images than its predecessors, but it also reduces problems like jagged edges and aliasing that are frequently present in lower-order interpolation techniques.

Chapter 3

Data and Methodology

3.1Global Historical Climatology Network Daily (GHCNd):

Daily climate summaries from land-based weather stations throughout the world are included in this database.

It is an enormous compilation of daily weather reports from multiple sources that NOAA's National Centres for Environmental Information (NCEI) has combined and implemented quality control inspections.

GHCND provides a variety of data:

- Maximum Temperature(degree Celsius)
- Minimum Temperature(degree Celsius)
- Total Precipitation(mm)
- Snowfall(mm)
- Snow Depth on Ground(mm)

GHCNd has information from stations in more than 180 nations and territories, giving it a tremendous worldwide reach. There are a significant number of stations—more than 100,000—that provide a wealth of data for climate studies. This data constantly gets updated and arrives from various sources, and NOAA's National Centers for Environmental Information (NCEI) performs quality checks on it.

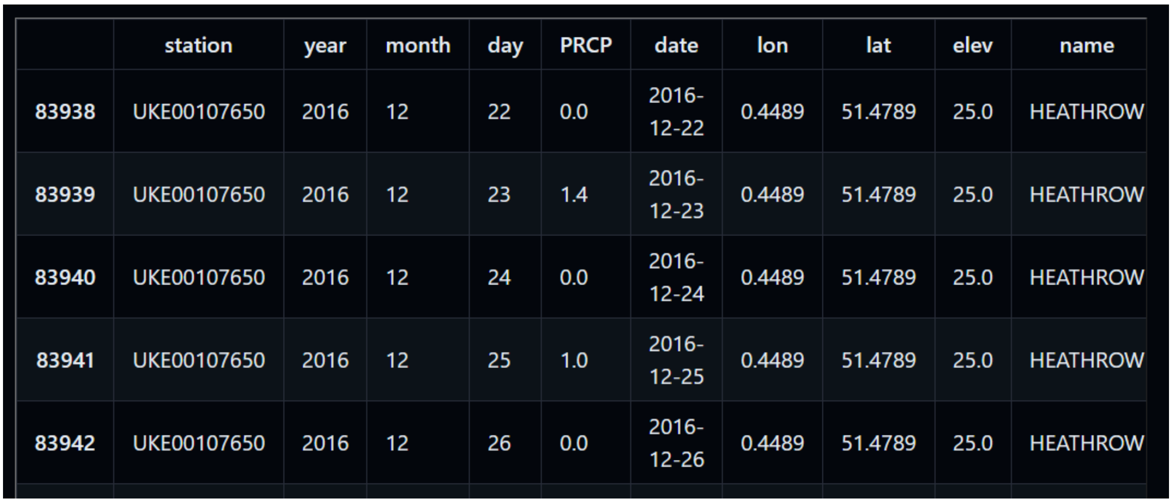
Metadata used here:

- Stations: Station ID, latitude, longitude, elevation, State (if applicable), and Station name

For extracting and usage purposes, we did not download this data. Instead, we cloned it from a GitHub repository so that we could work on it like a Python library.

```
git clone https://github.com/scotthosking/get-station-data.git
cd /path/to/my/get-station-data .
pip install -v -e .
```

(SCOTTOTHOSKING, n.d.)



	station	year	month	day	PRCP	date	lon	lat	elev	name
83938	UKE00107650	2016	12	22	0.0	2016-12-22	0.4489	51.4789	25.0	HEATHROW
83939	UKE00107650	2016	12	23	1.4	2016-12-23	0.4489	51.4789	25.0	HEATHROW
83940	UKE00107650	2016	12	24	0.0	2016-12-24	0.4489	51.4789	25.0	HEATHROW
83941	UKE00107650	2016	12	25	1.0	2016-12-25	0.4489	51.4789	25.0	HEATHROW
83942	UKE00107650	2016	12	26	0.0	2016-12-26	0.4489	51.4789	25.0	HEATHROW

Figure 12:An example image of GHCND data as a pandas data frame is shown above.

3.2 CHIRPS(Rainfall Estimates from Rain Gauge and Satellite Observations):

An essential component of environmental monitoring and drought early warning systems is estimating fluctuations in rainfall over time and space. It is necessary to contextualize a developing drier-than-normal season historically to assess the severity of rainfall deficiencies promptly. In contrast, more rural areas with fewer rain-gauge stations have worse precipitation grids created from station data. Working with scientists at the USGS Earth Resources Observation and Science (EROS) Centre, CHIRPS was developed to provide comprehensive, dependable, and current data sets for various early warning goals, such as trend analysis and seasonal drought monitoring. (devolopers, n.d.)

CHIRPS produces these precipitation estimates by combining information from many sources:

Satellite imagery: Cold cloud tops, frequently linked to precipitation, can be identified in areas using satellite infrared data.

Ground-Based Weather Stations: Real-time precipitation readings are obtained from in-situ observations made by weather stations. CHIRPS seeks to provide a more complete view of global precipitation patterns by combining data from both sources, especially in areas with little data.

Here are some benefits of CHIRPS Data:

CHIRPS is helpful for large-scale studies of precipitation patterns since it provides quasi-global coverage. The data provides a historical view of precipitation trends, going all the way back to 1981. Users can customize their analyses to meet particular requirements by utilizing available data at various temporal and spatial resolutions. A broad spectrum of users can obtain CHIRPS data since it is made available to the public by several sources.

The data was downloaded from **UCSB CHG Data Catalog:** The University of California, Santa Barbara's Climate Hazards Group maintains a data catalog entry for CHIRPS: https://developers.google.com/earth-engine/datasets/catalog/UCSB-CHG_CHIRPS_DAILY

downloaded in the range of year (1984,2020). After the download, we sliced it over a particular region i.e.- Delhi, and converted it into x-array for further use. (CENTRE, n.d.)

3.3 Data Information and processing :

3.3.1Global Historical Climatology Network Daily (GHCNd):

This integrated database of daily climate summaries, which derives from land surface stations across the globe, is quality assured. So, the official website suggests that we download the data from there and use it. However, the data files were too large to be used in the local system, so we used a GitHub repository instead. It was a Python library tool to extract data on daily weather stations. We cloned the given below repository along with its dependencies:

```
!git clone https://github.com/scotthosking/get-station-data.git

!mv get-station-data/* .

!pip install -v -e.
```

Then we chose a location by putting the longitude-latitude (lon/lat) value. The lat/lon value was given for Delhi and 100 nearest neighbors were also chosen as part of the procedure that came with instructions.

```
delhi_lon_lat = 77.1025, 28.7041

my_stns = nearest_stn(stn_md,

                        delhi_lon_lat[0], delhi_lon_lat[1],

                        n_neighbours=100)
```

However, in doing so, many Zeros and NAN values came from the many stations chosen as the 100 nearest neighbors. So we tried to find stations with the lowest NAN values to remove the complications and sophistication, and we found our only reliable stations following that condition. Those stations were ‘New Delhi/Palam’ and ‘New Delhi/ Safdarjun.’

And then we created both a pandas data frame and CSV files out of that extracted daily stations data with long-term terms to use it further. This data is used as a source input data.

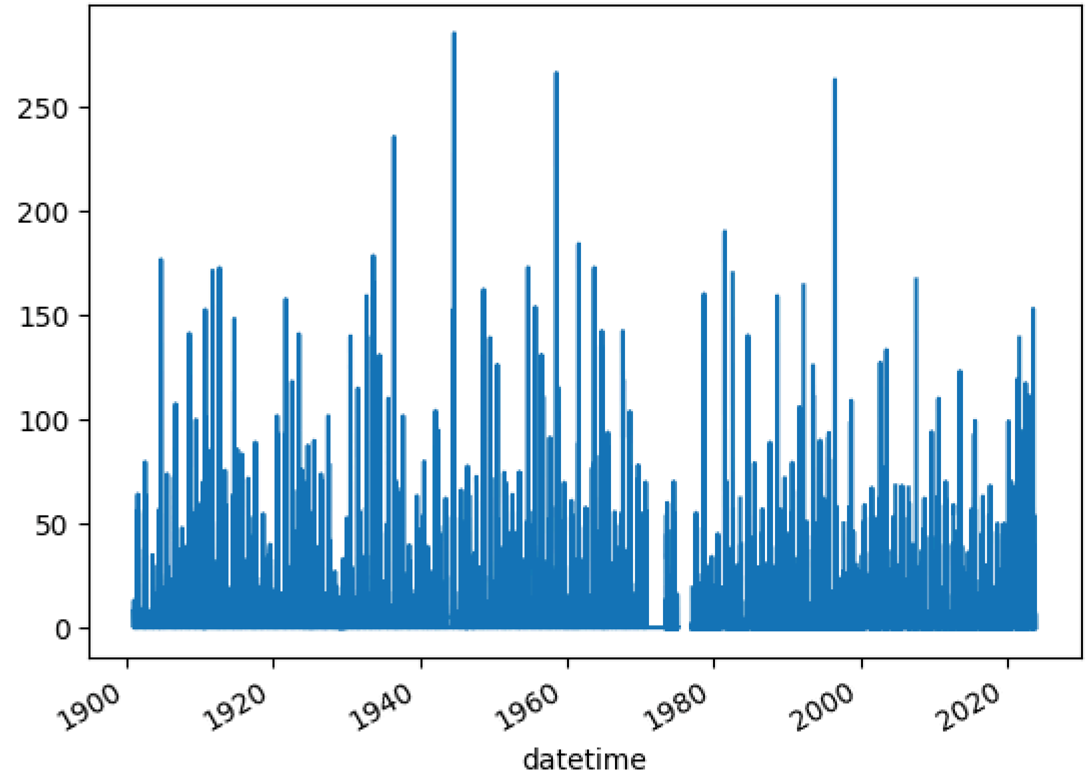
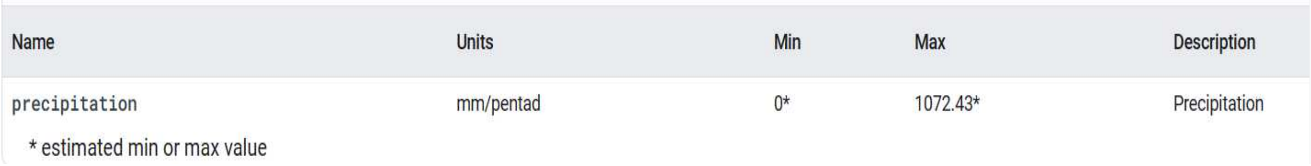


Figure 13:GENERATED PLOT OF PRECIPITATION OVER TIME (NEWDELHI/SAFDARJUN)

3.3.2 CHIRPS(Rainfall Estimates from Rain Gauge and Satellite Observations):

after generating a 300m grigged dataset with MeteoGAN, we need something to compare with a reliable existing dataset which is CHIRPS. It is also a high-resolution data of 0.05-degree precipitation climatologies.

```
ee.ImageCollection("UCSB-CHG/CHIRPS/PENTAD")
```



Name	Units	Min	Max	Description
precipitation	mm/pentad	0*	1072.43*	Precipitation

* estimated min or max value

Figure(Chirps Google Devp.)

after downloading by following the instructions, we used it as netcdf4 file for easy execution because the file size was smaller and easily could be handled from a local computer. we downloaded it for for 1983 to 2020. (CENTREE, n.d.)

3.3.3 Normalization:

In Data Normalization, we preprocess to standardize features within the dataset to ensure all features are on a similar scale, making them comparable during analysis or used in machine learning models that are sensitive to feature scale. Depending on the requirements, different normalizations are used. The two most popular normalizations are min-max normalization and Z-Score normalization.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

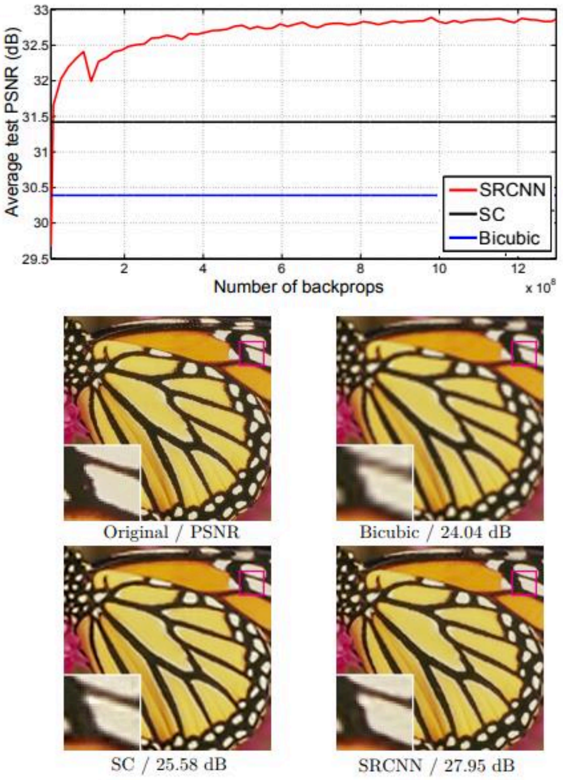
$X_{min} = minimum\ value\ of\ feature\ x$ $X_{max} = maximum\ value\ of\ feature\ x$

Mim-max normalization scales each feature up to a range between 0-1. Before training, we did min-max normalization, and the normalization was based on CHIRPS data. It does not matter what baseline we use for normalization.

3.4Models:

3.4.1 SRCNN

The 2014 introduction of SRCNN (Super-Resolution Convolutional Neural Network) marked a significant advancement in deep learning for super-resolution (SR) imaging.



Original / PSNR

Bicubic / 24.04 dB

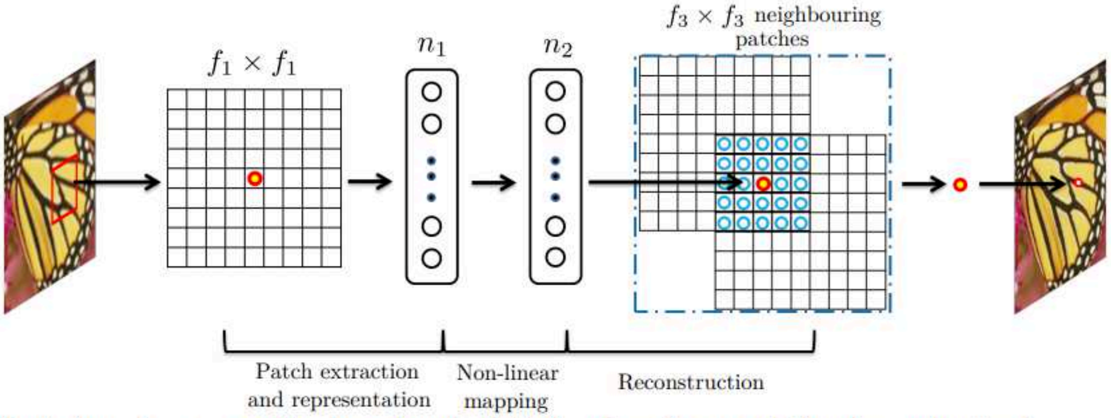
SC / 25.58 dB

SRCNN / 27.95 dB

Figure 14 (Park, 2021)

Working of SRCNN:

SRCNN employs a very straightforward 3-layer deep convolutional neural network (CNN) architecture to achieve picture super-resolution.


$f_1 \times f_1$
n_1
n_2
$f_3 \times f_3$ neighbouring patches
Patch extraction and representation
Non-linear mapping
Reconstruction

An illustration of sparse-coding-based methods in the view of a convolutional neural network.

Figure 15 (Park, 2021)

The low-resolution (LR) image is fed into the network for feature extraction. The first convolutional layer extracts the LR image's features. The critical information and patterns needed to rebuild a high-resolution (HR) image are captured by these features. To add non-linearity to the model, the extracted features are run through a non-linear activation layer, usually a ReLU layer. As a result, the network can understand more intricate connections between LR and HR images. From the processed features, a final convolutional layer reconstructs the HR image. This layer mapped the retrieved features back to a high-resolution representation.

Training Process:

Data Preparation and Training: a dataset with matched LR and HR images is needed. High-resolution cameras, downsampled high-resolution photos, and generative models can all produce the HR images. Preprocessing is done on the data to enhance model performance. And the SRCNN Data is trained iteratively over the paired LR-HR data

Evaluation:

After training, a different dataset of previously unseen LR and HR images is used to assess the model. When evaluating the model's ability to reconstruct high-resolution images, metrics such as the Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR) are employed.

The SRCNN has a relatively simple architecture, making it computationally efficient and more accessible to train. And made better improvements in image super resolution than traditional methods.

3.4.2 MeteoGAN:

MeteoGAN represents an innovative leap forward in meteorological data analysis, particularly in the downscaling process where coarse-resolution maps are refined into high-resolution information. This process is crucial for enhancing the quality of weather forecasting, climate, and agriculture models by bridging the gap between large-scale atmospheric phenomena and local weather patterns.

The foundational concern in developing MeteoGAN was the selection of initial datasets for both coarse and high-resolution maps. The initial idea of adopting an autoencoder-like generator was eventually surpassed by a more dynamic approach—an iterative Generator or an iterative SRCNN (Super-Resolution Convolutional Neural Network). This iterative process generates inputs and targets, after which the targets are fed to a discriminator alongside station data. The discriminator's task is to ensure the generated high-resolution outputs are realistic by comparing them with actual station data.

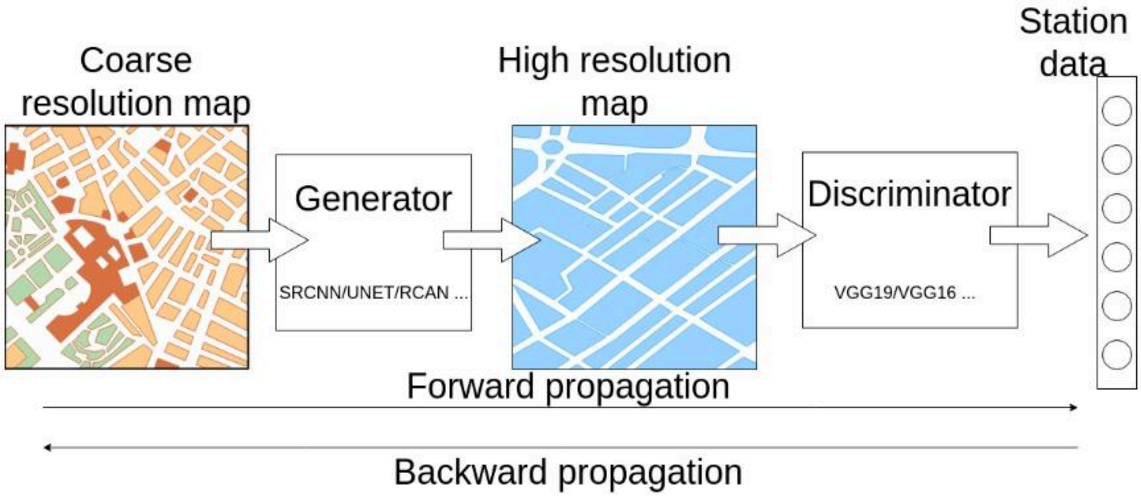


Figure 16: Architecture of MeteoGAN (Manmeet, 2023)

In the development of MeteoGAN was whether the discriminator requires explicit spatial information to match generated data with corresponding locations within the domain accurately. The resolution to this dilemma lies in the network's capacity to autonomously adjust its weights to optimally represent data at the correct locations, negating the need to manually input spatial information. This autonomous adjustment underscores the importance of a well-structured training dataset, suggesting the creation of NetCDF files that encapsulate the meteorological data and the associated station information for each timestamp.

This approach not only aids in accurately generating high-resolution maps but also in strategizing the placement and number of observational stations across a region for comprehensive coverage.

Traditionally, downscaling in meteorology through deep learning borrowed heavily from super-resolution algorithms developed in computer vision. However, MeteoGAN introduces a novel value proposition by integrating station data directly into the training process of deep learning models, thereby enhancing the accuracy and reliability of the downscaled outputs. Built upon the foundation laid by Singh et al. in 2023 with the iterative SRCNN for generating high-resolution data (300 m), MeteoGAN utilizes a generative adversarial network (GAN) framework. This GAN leverages the output of an iterative super-resolution algorithm, incorporating station information to produce highly accurate meteorological data.

MeteoGAN's efficacy was demonstrated through its application to precipitation data over Delhi, showcasing significant improvements in the quality and reliability of the generated high-resolution outputs. By training the model with selected station data and testing it against high-resolution output, MeteoGAN presents a compelling case for adopting advanced deep-learning techniques in meteorological downscaling. This approach enhances the precision of weather forecasts and contributes valuable insights for climate research, potentially leading to more informed decision-making in response to climate variability and change."

Model Training:

We aimed to train the MetoGAN and generate 300m resolution gridded data. First, we created and trained the architecture of SRCNN. Trained it with CHIRPS at 5 km to iteratively come to generate 300 m grids. Then we did bicubic interpolation on CHIRPS at 5km and gave it as the input to SRCNN for initial training of the MeteoGAN generator. We used this Bicubic interpolated data at 5 km to generate inputs and Targets of the Generator part of MeteoGAN. We treated this dataset as our coarse input and took the original GHCND stations dataset as the target for the Discriminator. Using this base SRCNN at 5km, we iteratively go to 300m. At the same time, we also do Bicubic interpolation on CHIRPS to make it reach 300m for our comparison.

Then, we gave this 300m generated SRCNN data to our created deep learning architecture, similar to SRGAN(Single Image Super Resolution Network), which we call MeteoGAN for further training of it. In MeteoGAN, we took Generator as SRCNN. We could also take another model, but we try to use SRCNN only. Its discriminator part contains no specific model but presently its almost similar to VGG19 (Symonion, 2015) convolutional neural network, but it works as usual. Then, we took Targat for our Discriminator part as our GHCND stations dataset, as shown in the schematic Figure 14.

Chapter4

Results and Discussion

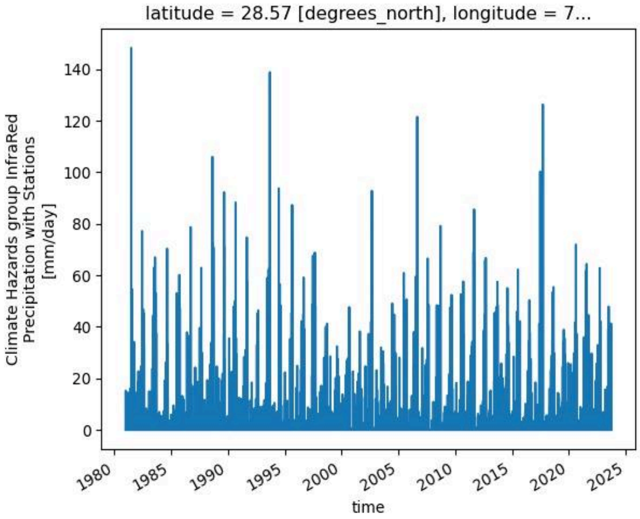


Figure 17: CHIRPS time series

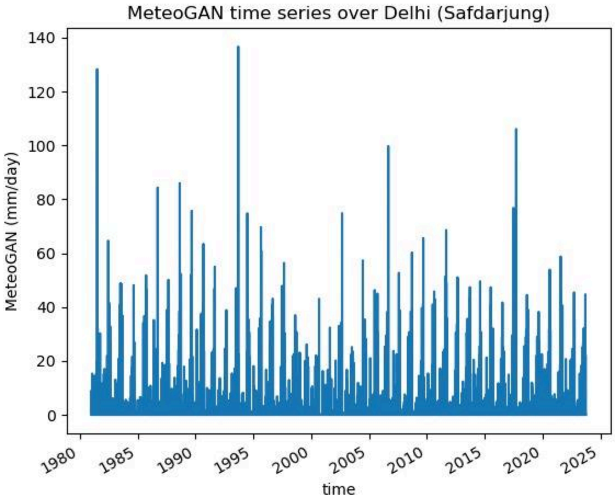


Figure 18: MeteoGAN times series

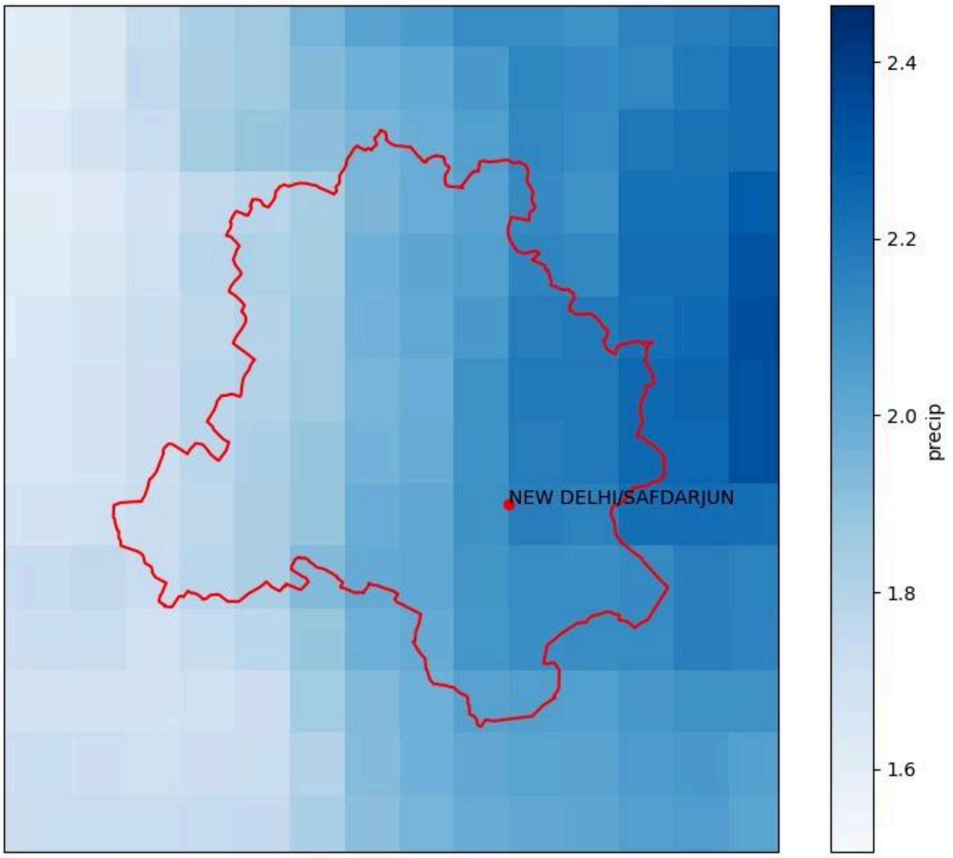


Figure 19:Boundary covered for our precipitation data(GHCND-stations)

Pearson correlation coefficient between GHCND stations data & CHIRPS 300m gridded data	**0.297**
Pearson correlation coefficient between GHCND stations & MeteoGAN generated 300m gridded data	**0.302**

The Pearson Correlation Coefficient values of GHCND stations -Chirps and GHCND-MeteoGAN are close and slightly different. But here, we are looking at a time span of around 40 years (1980-2020), and these correlation values represent the daily variations.

Given the station's data from GHCND (climate hazard group infrared precipitation data), as shown in Figure 17, we generated the data from MeteoGAN architecture at 300m. Both CHIPRS and MeteoGAN data are at 300m resolution received through iterative SRCNN. The above plots are for a particular region of Delhi – Safdarjung, which spans almost 40 years(1980 to 2020). It means that we worked on nearly 40×365 values of precipitation.

We generated plots for precipitation on a particular date(time) on 1993-09-10. We plotted for our trusted original CHIRPS data, bicubic interpolation applied data, and MeteGAN generated precipitation data. According to all these plots, if we compare all three of them(CHIRPS, Bicubic, MeteGAN), we can see that bicubic interpolation just stretches out the resolution. Whereas MeteoGAN gives a slightly better resolution. and CHIRPS at 300m cannot give fine resolution as the images are getting ripped at 300m resolution for CHIRPS. But if we take a look at MetoGAN(srcnn_300m) we can see that there is some slight improvement in resolution ie- variety of different precipitation values over that same region covered which implies that the better resolution.

Date: 1993-09-10

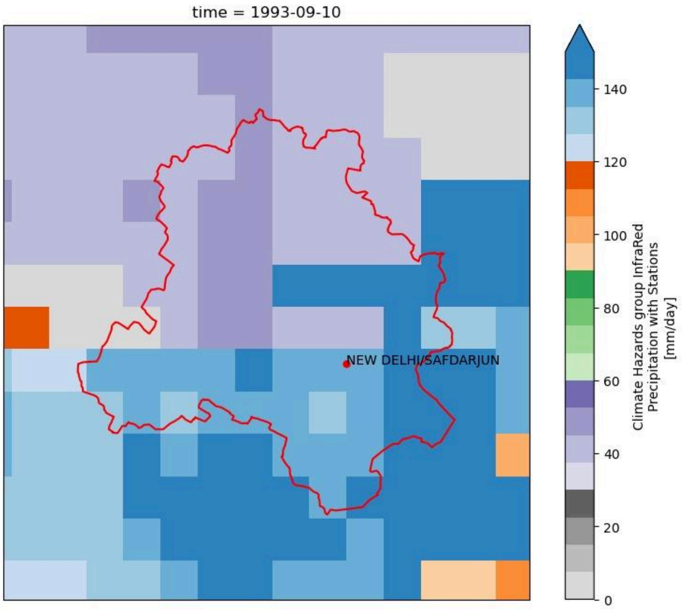

time = 1993-09-10
NEW DELHI/SAFDARJUN
140
120
100
80
60
40
20
0
Climate Hazards group InfraRed Precipitation with Stations [mm/day]

Figure 20: precipitaion over Safdarjung on 1993-09-10(CHIRPS)

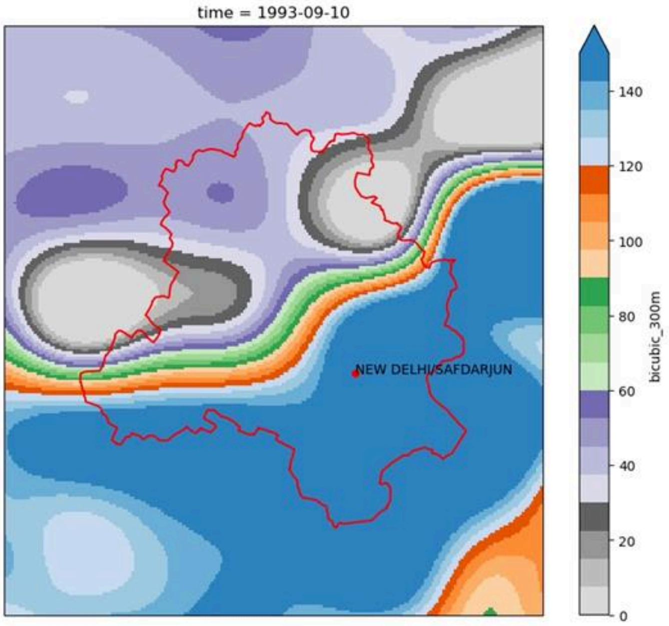


Figure 21: precipitation over safdarjung on 1993-09-10 (Bicubic)

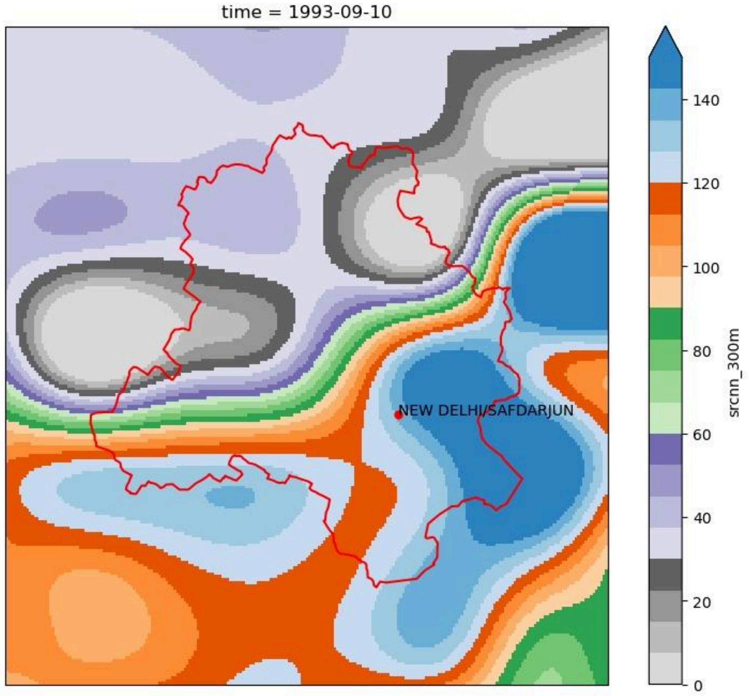


Figure 22: precipitation over Safdarjung on 1993-09-10(MeteoGAN generated 300m)

Date: 2017-09-22:

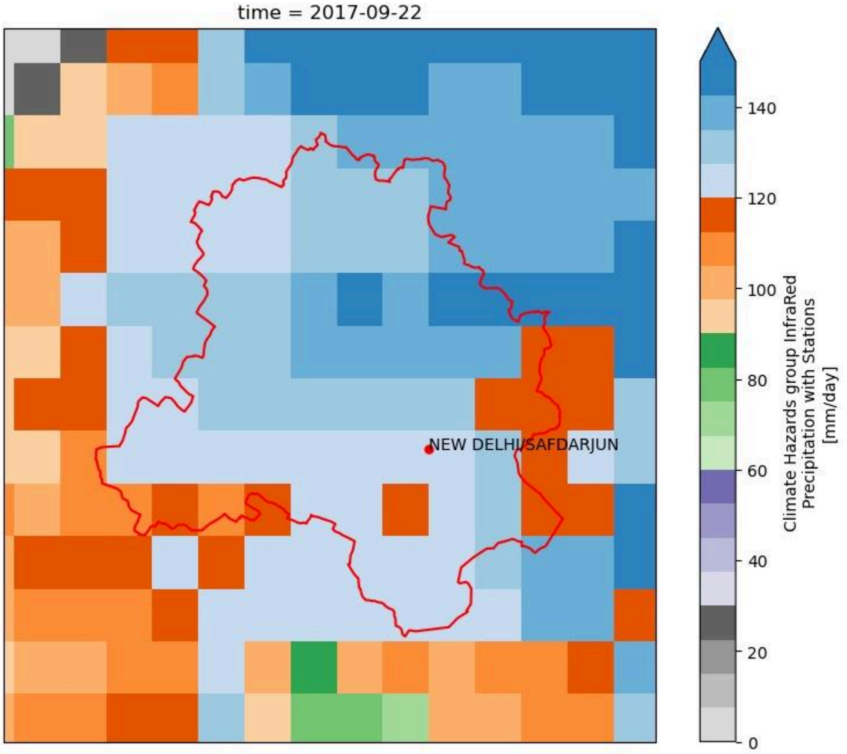

time = 2017-09-22
NEW DELHI/SAFDARJUN
Climate Hazards group InfraRed Precipitation with Stations [mm/day]
0
20
40
60
80
100
120
140

Figure 23: precipiatation over safdarjun on 2017-09-22(CHIRPS)

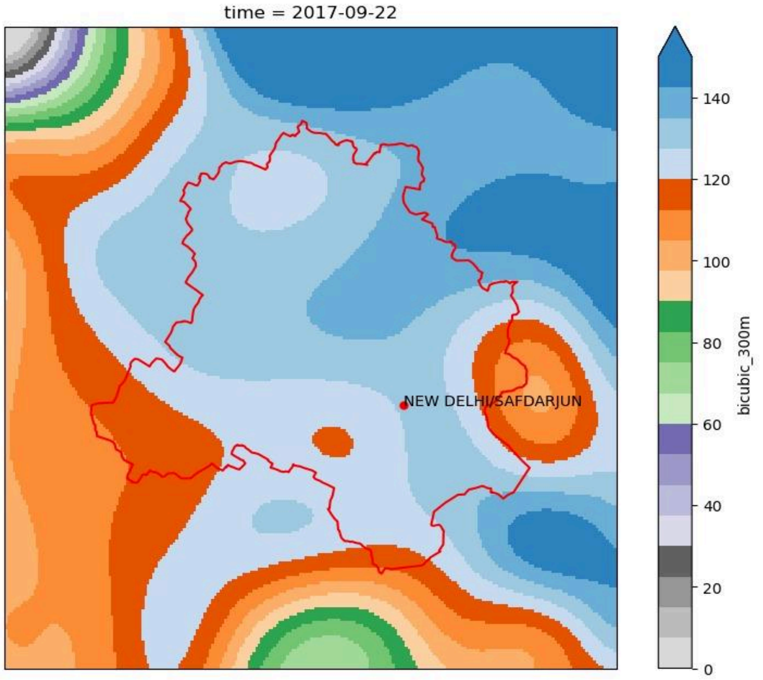


Figure 24:precipitation over safdarjung on 2017-09-22(Bicubic)

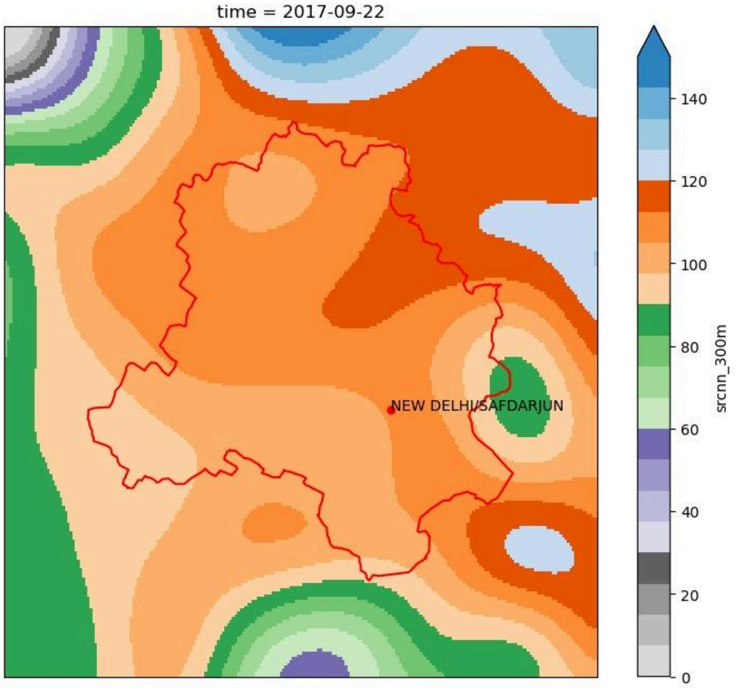


Figure 25:precipitation over safdarjung on 2017-09-2022(MeteGAN)

Chapter 5

Conclusion and Future Work

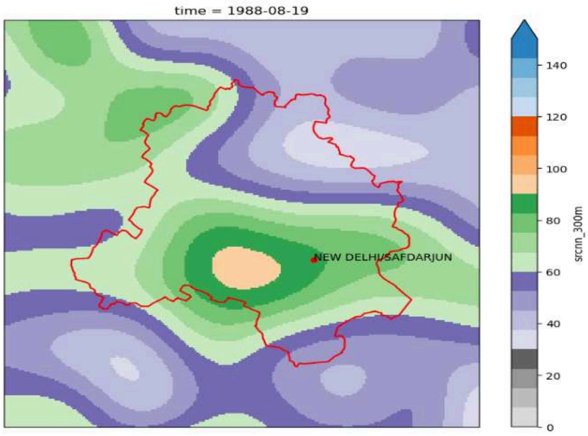
We have generated 300m gridded resolution data of precipitation values by putting our coarse input data to a GAN architecture called MeteoGAN. It has shown a slightly better resolution in the spatial distribution of precipitation plotted for a particular DateTime. If we argue, One can also say that it is not entirely GAN because the Discriminator part does not have a perfect model embedded, the model is presently similar to VGG19 and we plan to do something better with time.

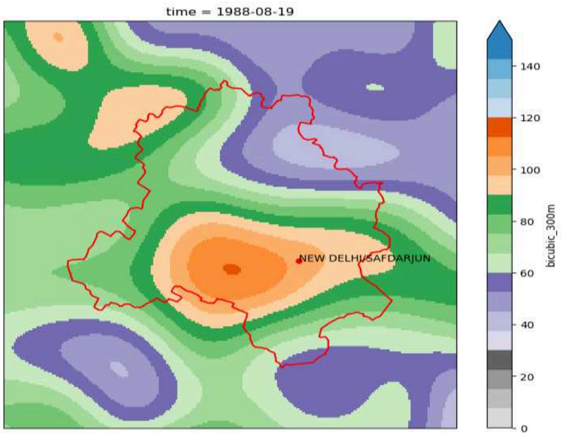
In the Future, we plan to generate more similar plots on spatial variation of precipitation and compare them with our GHCNd station's data for a better outlook. We plan to use more statistical metrics like the Z-score, Kolmogorov-Smirnov, etc, to understand our results in a better way for further improvements. After this, we plan to generate more gridded data over other parts of India and the USA, for which long-term consistent precipitation observations will be available for further comparison and analysis.

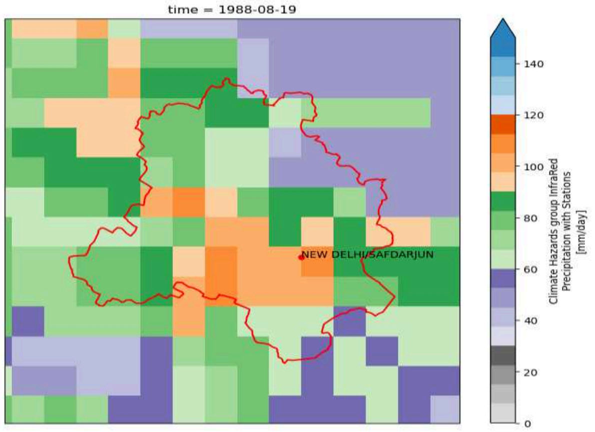
Appendix

Some more examples of spatial plots genetared by MeteoGAN and its respective CHIRPS and bicubic-interpolation applied plots.

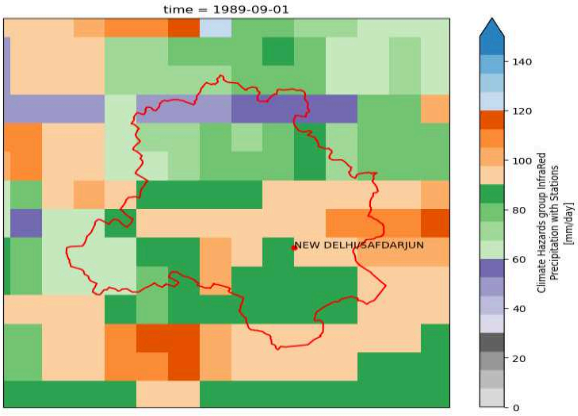
Date:1988-08-19

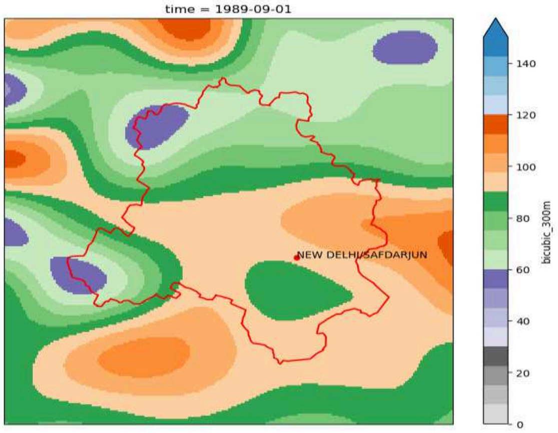


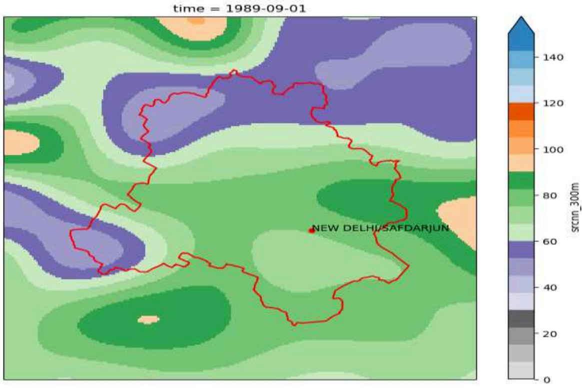



time = 1988-08-19
NEW DELHI/SAFDARJUN
140
120
100
80
60
40
20
0
Climate Hazards group InfraRed Precipitation with Stations [mm/day]

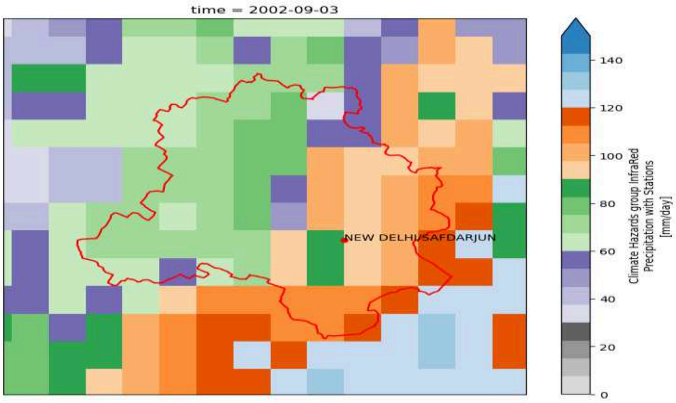
Date:1989-09-01

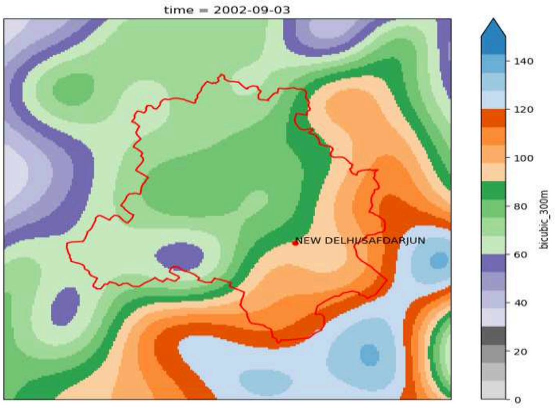

time = 1989-09-01
NEW DELHI/SAFDARJUN
140
120
100
80
60
40
20
0
Climate Hazards group InfraRed Precipitation with Stations [mm/day]

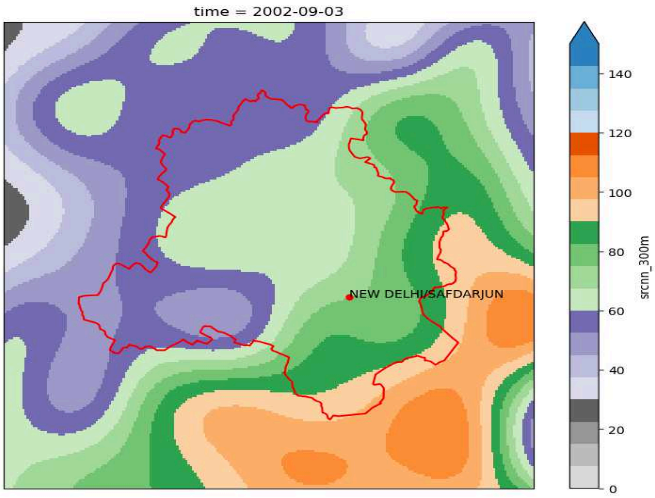




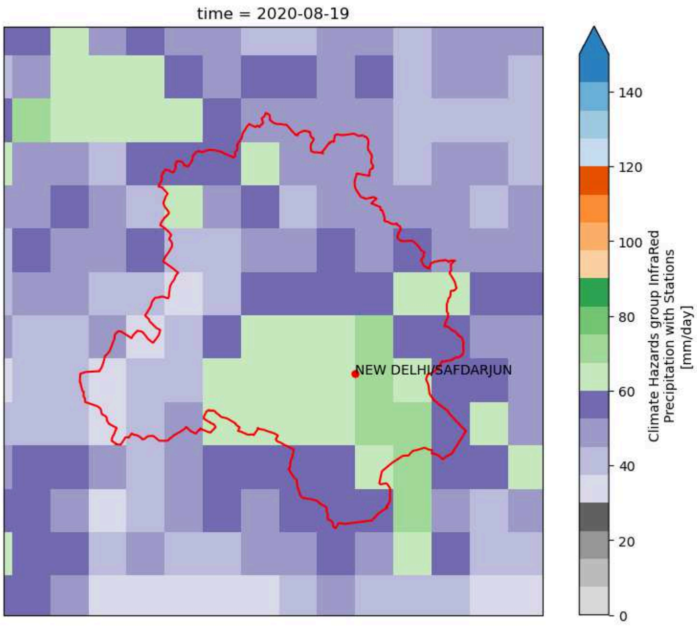
Date:2002-09-03

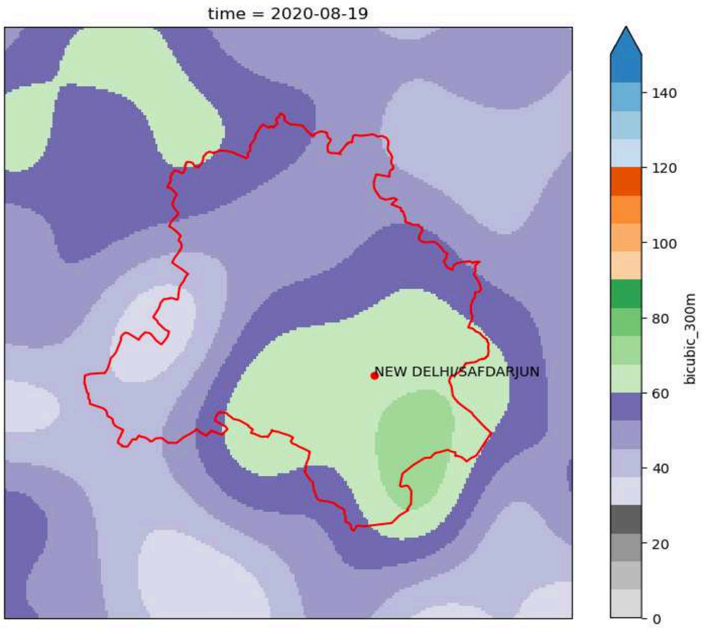

time = 2002-09-03
NEW DELHI/SAFDARJUN
140
120
100
80
60
40
20
0
Climate Hazards group InfraRed Precipitation with Stations [mm/day]

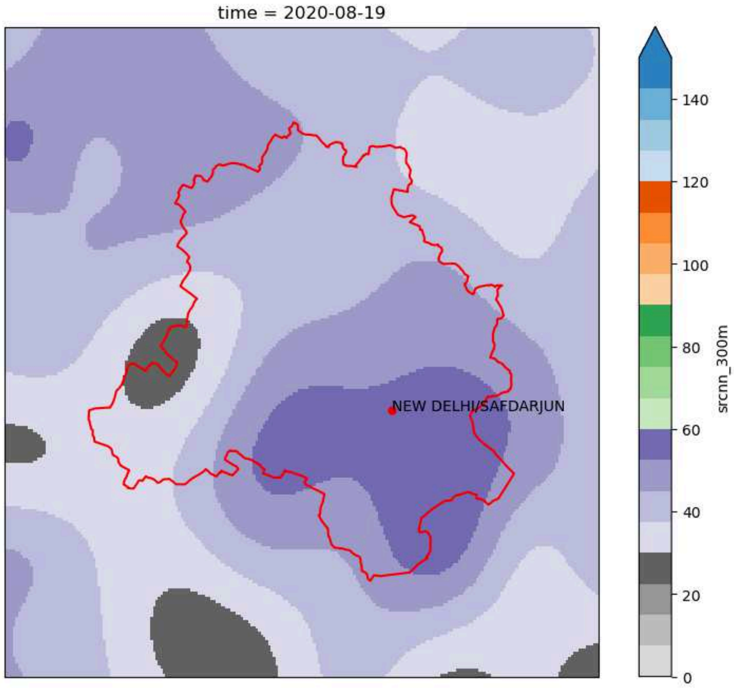




Date: 2020-08-19


time = 2020-08-19
NEW DELHI/SAFDARJUN
Climate Hazards group InfraRed Precipitation with Stations [mm/day]
0
20
40
60
80
100
120
140





Bibliography

al, D. e. (2018). *medium*. Retrieved from medium: https://medium.com/@ramyahrgowda/srgan-paper-explained-3d2d575d09ff

analyticsarora. (2023). *17 Unique Machine Learning Interview Questions about Gradient Descent.* analyticsarora.

bipin. (2023, march 03). *On the modern deep learning approaches for precipitation downscaling*. Retrieved from springer link: https://link.springer.com/article/10.1007/s12145-023-00970-4

Bre, F. (2017). *Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks*. Retrieved from resaerchgate: https://www.researchgate.net/publication/321259051_Prediction_of_wind_pressure_coefficients_on_building_surfaces_using_Artificial_Neural_Networks?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoiX2RpcmVjdCJ9fQ

centre, c. h. (n.d.). *CHIRPS*. Retrieved from https://www.chc.ucsb.edu/data/chirps

centree, c. h. (n.d.). *CHIRPS*. Retrieved from rainfall estimation: https://www.chc.ucsb.edu/data/chirps

David. (2019). *Tackling Climate Change with Machine Learning*. Retrieved from arxiv: https://arxiv.org/abs/1906.05433

devolopers, G. (n.d.). *CHIRPS Pentad: Climate Hazards Group InfraRed Precipitation With Station Data (Version 2.0 Final)* . Retrieved from Earth Engine Data Catalog: https://developers.google.com/earth-engine/datasets/catalog/UCSB-CHG_CHIRPS_PENTAD#bands

Dong. (2014). Image Super-Resolution Using Deep Convolutional Networks. *Europeon conference in computer vision*, 16.

enthalppy. (n.d.). *SRGAN-Super-Resolution-GAN*. Retrieved from GitHub: https://github.com/entbappy/SRGAN-Super-Resolution-GAN#readme

et.al, G. (2020). *Degradation state recognition of piston pump based on ICEEMDAN and XGBoost*. Retrieved from reserchgate: https://www.researchgate.net/publication/345327934_Degradation_state_recognition_of_piston_pump_based_on_ICEEMDAN_and_XGBoost?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoiX2RpcmVjdCJ9fQ

et.al, Y. (2019). *Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail*. Retrieved from https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451

Gabormelli. (2024). *artificial nueron*. Retrieved from gabormelli: https://www.gabormelli.com/RKB/Artificial_Neuron gfg. (GFG). *what is linear regression* (first ed.). GFG. Retrieved from https://www.geeksforgeeks.org/what-is-regression-line/

Huang, C.-J. (2022, june 15). *Vanilla Generative Adversarial Networks*. Retrieved from medium: https://bchuan110.medium.com/vanilla-generative-adversarial-networks-4cd90d624197

Kumar, B. (2021, January 02). *Deep learning–based downscaling of summer monsoon rainfall data over Indian region*. Retrieved from Springer: https://link.springer.com/article/10.1007/s00704-020-03489-6

manmeet. (2022, august 15). *Urban precipitation downscaling using deep learning: a smart city application over Austin, Texas, USA*. Retrieved from arxiv: https://arxiv.org/abs/2209.06848

Manmeet, S. (2023). *metrgan architecture*. Retrieved from docs: https://docs.google.com/document/d/12IepLWTSrNoe_eS-gwmTzGnElXHKfQCsVGm9Ujtn60U/edit

Markus. (2019). *https://www.nature.com/articles/s41586-019-0912-1*. Retrieved from Deep learning and process understanding for data-driven Earth system science: https://www.nature.com/articles/s41586-019-0912-1

neelesh. (2022, december). *High-resolution downscaling with interpretable deep learning: Rainfall extremes over New Zealand*. Retrieved from ScienceDirect: https://www.sciencedirect.com/science/article/pii/S2212094722001049

Park, S. (2021). *SRCNN EXPLAINED*. Retrieved from medium: https://medium.com/analytics-vidhya/srcnn-paper-summary-implementation-ad5cea22a90e

scottothosking. (n.d.). *get-sations-data*. Retrieved from GitHub: https://github.com/scotthosking/get-station-data

superannote. (2024). *Convolutional Neural Networks: 1998-2023 Overview*. Retrieved from superannote: https://www.superannotate.com/blog/guide-to-convolutional-neural-networks

vandal. (2017, march 09). *DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution*. Retrieved from arxiv: https://arxiv.org/abs/1703.03126

Simonyan, K., & Zisserman, A. (2014, September 4). *Very deep convolutional networks for Large-Scale image recognition*. arXiv.org. https://arxiv.org/abs/1409.1556

Audit Trail

Document Details

Title	vishal_ms_thesis.pdf
File Name	vishal_ms_thesis.pdf
Document ID	1384638ab45c46a39102ef6780a8dedc
Fingerprint	6f2af957687c5fd02fb6337c5fb27758
Status	Completed

Document History

Document Created	Document Created by Manmeet Singh (manmeet.cat@tropmet.res.in) Fingerprint: dd875a321738e921a7350bc7acfa0f72	May 22 2024 03:14AM UTC
Document Signed	Document Signed by Manmeet Singh (manmeet.cat@tropmet.res.in) IP: 128.62.164.190 Manmeet Singh	May 22 2024 03:14AM UTC
Document Completed	This document has been completed. Fingerprint: 6f2af957687c5fd02fb6337c5fb27758	May 22 2024 03:14AM UTC