



**Using state-of-the-art Machine Learning  
techniques to derive high-resolution  
precipitation projections over the Indian  
region in a warming climate**

**Master's Thesis**

*submitted to*

**Indian Institute of Science Education and Research Tirupati**  
*in partial fulfilment of the requirements for the*  
**BS-MS Dual Degree Programme**

*by*

**Joshin John Bejoy**

Roll Number: 201801163

Supervisor: Dr. Chirag Dhara

Department of Science

Krea University, Andhra Pradesh, 517646 India

April, 2023

©Joshin John Bejoy & Dr. Chirag Dhara(2023)  
All rights reserved

# Certificate

This is to certify that the MS Thesis entitled **Using state-of-the-art Machine Learning techniques to derive high-resolution precipitation projections over the Indian region in a warming climate**, submitted towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Tirupati, represents the study /work carried out by Joshin John Bejoy at Indian Institute of Science Education and Research Tirupati under the supervision of **Dr. Chirag Dhara**, Department of Science, during the academic year 2022-2023 and the same has not been submitted for any other degree/diploma of any other university or institute.

Name of student: **Joshin John Bejoy**

Roll Number: 201801163



Signature of student:

Date: April 5, 2023

Supervisor: Dr. Chirag Dhara



Signature of Supervisor:

Date: April 5, 2023

Co-Supervisor: Dr. Manmeet Singh



Signature of Co-Supervisor:

Date: April 5, 2023



## Declaration

I declare that the matter presented in the MS thesis entitled **Using state-of-the-art Machine Learning techniques to derive high-resolution precipitation projections over the Indian region in a warming climate**, are the results of the study /work carried out by me at the Department of Physics, Indian Institute of Science Education and Research Tirupati (or) External Institute....., under the supervision of Dr. Chirag Dhara .

The contents are expressed in my own words and where others' ideas have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic ethics and integrity and have not fabricated or falsified any idea /data /fact /image /source in my submission.

I understand that violation of the above will lead to disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.

**Name of student: Joshin John Bejoy**

Roll Number: 201801163



Signature of student:

Date : April 5, 2023

**Endorsed by:**

Supervisor: Dr. Chirag Dhara



Signature of Supervisor:

Date: April 5, 2023

Co-Supervisor: Dr. Manmeet Singh



Signature of Co-Supervisor:

Date: April 5, 2023



## Acknowledgements

I express my sincere gratitude to Dr. Chirag Dhara, the supervisor and the one who suggested and motivated me to take up this topic for my MS Thesis work. His expertise, encouragement and invaluable support led to the completion of this work. I express my gratitude towards Dr. Manmeet Singh, the co-supervisor of this project. His previous work has laid the foundation of my project. Dr. Manmeet introduced me to the latest state-of-the-art down-scaling models in computer vision. This project became possible through his support, advice and the computational resources he gave me access to. I used the docker images he had prepared for the project. I thank Mr. Sreehari K., who helped me when I faced issues while handling the climate data. I thank Mr. Vaisakh S. B. for instructing and helping me work with the docker containers. I am grateful to Indian Institute of Tropical Meteorology's HPC facility. I thank my family and friends who strengthened me to complete this journey and finally I thank Indian Institute of Science Education and Research IISER Tirupati, the MS thesis committee, the IT Department, the library services and others who toiled for conducting the process of this research and evaluation smoothly.



# Abstract

Inspired by the observation of previous works that the central Indian region where localised extreme precipitation events are on the rise, several deep-learning techniques like SRCNN, SRGAN, ESRGAN and Real-SRGAN for image super-resolution are reviewed and compared. ESRGAN is found apt for the purpose of down scaling coarse precipitation data. It is adopted for climate application in general and is trained on observational data. This model is evaluated by comparing the original high resolution and down scaled versions of CMIP6 precipitation projections. The code and training techniques are made available through github, which can be directly applied on any coarse NetCDF climate data file to immediately produce a high resolution version of the same.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Rise in extreme climatic events over central India . . . . .	4
1.3	Traditional statistical downscaling . . . . .	6
1.4	Artificial Intelligence . . . . .	7
1.5	Neural Networks . . . . .	8
<b>2</b>	<b>Theoretical background - Downscaling models</b>	<b>11</b>
2.1	Super Resolution Convolutional Neural Networks SRCNN . . . . .	11
2.1.1	Training . . . . .	12
2.1.2	Activation functions . . . . .	12
2.1.3	Image quality evaluation metrics . . . . .	13
2.2	Generative Modelling . . . . .	14
2.2.1	Variational Auto Encoders VAE . . . . .	14
2.2.2	GAN based approaches . . . . .	15
2.3	Visual Geometry Group-VGG . . . . .	16
2.4	Super Resolution GAN type models (SRGAN) . . . . .	16
<b>3</b>	<b>Inferences from various models</b>	<b>21</b>
<b>4</b>	<b>Data set</b>	<b>23</b>
4.1	Data set downloading . . . . .	23
4.2	Kernel width experiments . . . . .	23
4.3	Preparing training set . . . . .	25
<b>5</b>	<b>Training</b>	<b>27</b>
5.1	RRDBNet . . . . .	27
5.2	ESRGAN . . . . .	27

*CONTENTS*

---

<b>6 Results and Discussion</b>	<b>29</b>
6.1 Key training findings . . . . .	29
6.2 Validation of the model . . . . .	29
6.3 Study of trends of extreme events from CMIP6 projections . . . . .	34
<b>7 Conclusion</b>	<b>37</b>
<b>Bibliography</b>	<b>39</b>

# List of Figures

1.1	Trends in global ST relative to 1850–1900 [1] . . . . .	1
1.2	When will risks become a critical threat to the world? (GRPS 2022) [2]	3
1.3	Current state of international risk mitigation efforts in each area (GRPS 2022) [2]	4
2.1	Illustrative chart of GAN Architecture . . . . .	16
2.2	Comparison of LR image and SR image downsampled by x4 using SRGAN	17
2.3	Comparison of LR image and SR image downsampled by x4 using ESRGAN	18
2.4	Comparison of LR and SR image downsampled by x4 using Real-ESRGAN	19
3.1	Bicubic blurred image of a tiger . . . . .	22
3.2	Comparison of LR and ESRGAN . . . . .	22
4.1	a)left: bi-cubic upscaled b)right: bi-cubic downsampled . . . . .	24
4.2	a)left: bi-cubic upscaled b)right: bi-cubic downsampled . . . . .	24
6.1	LR blurred CMIP6 data . . . . .	31
6.2	4x Super-resolved CMIP6 data . . . . .	32
6.3	Comparison of Ground truth CMIP6 projections image and SR images developed by downscaling the LR x4 using ESRGAN . . . . .	33
6.4	Plot of frequency of extreme events . . . . .	35

*LIST OF FIGURES*

---

# List of Tables

*LIST OF TABLES*

---

# Chapter 1

## Introduction

This thesis aims to bridge the gap between cutting edge deep learning technology and the techniques used to study climatic phenomena. We mainly investigate how the latest developments in **Computer vision** can be successfully employed to climate data. For the purpose of the project, we focus our attention on precipitation data.

### 1.1 Background

The global climate temperature reconstructions from paleo-climate archives denote that the global warming that has occurred during the last 170 years is unprecedented considering temperatures over the past 2000 years [1].

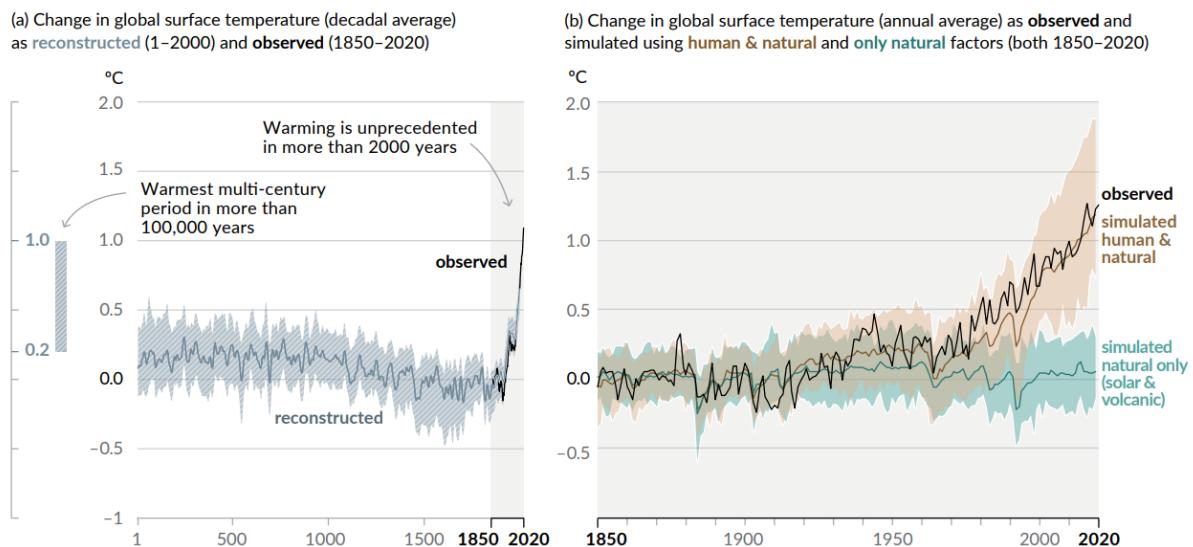


Figure 1.1: Trends in global ST relative to 1850–1900 [1]

## *1.1 Background*

---

According to the IPCC AR6 summary for policymakers, global temperatures have risen by  $1.1^{\circ}\text{C}$  since pre-industrial times. This trend of rising average global temperature was continued to be observed in the last two centuries even after the accuracy of recordings of climate variables from stations across the globe was improved. The roles of anthropogenic and natural factors contribute to the changing climate in various intensities according to IPCC Summary for Policymakers. Among the implications of rising temperatures is an increase in extreme weather events such as extreme rainfall, high-intensity cyclones, heat waves, and forest fires. These extreme events have the most significant impacts on human societies and the natural world. It has implications on food and water security, the economy, biodiversity, health, infrastructural integrity, transportation, and our energy systems. Therefore, studying why and how climate will change in the future and drafting measures to mitigate them are crucial to minimising loss and damage due to climate change.

The **Global Risks Perception Survey (GRPS 2022)** identifies societal and environmental risks as the primary concerns over the next five years. However, when we look further ahead ten years into the future, the surveyees prioritise the planet's health, perceiving environmental risks as the most significant long-term threats to the world and posing the greatest potential harm to both people and the earth. All the hazardous long-term risks that GRPS had identified faced a direct or indirect impact from "Climate action failure", especially "extreme weather events" and "biodiversity loss".



Figure 1.2: When will risks become a critical threat to the world? (GRPS 2022) [2]

Just like in previous years, GRPS 2023 also continues to identify and give more weightage to the environmental risks in GRR 20231.3.

Cities around the world faced extreme temperatures in 2020, with Madrid experiencing a record high of 42.7°C and Dallas recording a 72-year low of -19°C. The Arctic Circle also witnessed higher summer temperatures. This has increased pressure on governments, businesses, and societies to take action to prevent severe consequences. However, a disorderly climate transition may lead to divergent trajectories worldwide and across sectors, causing countries to drift apart and societies to divide, creating barriers to cooperation.

The GRPS survey asked about global efforts to reduce risk, particularly in relation to global threats. Respondents who were scientists and experts in various research areas identified specific areas where they believed that risk mitigation efforts were inadequate. They felt these efforts were either not yet initiated or were in the early

## 1.2 Rise in extreme climatic events over central India

---

stages of development in several frontiers. Among these, the one that was least established and the one that required immediate attention was the application of Artificial intelligence.

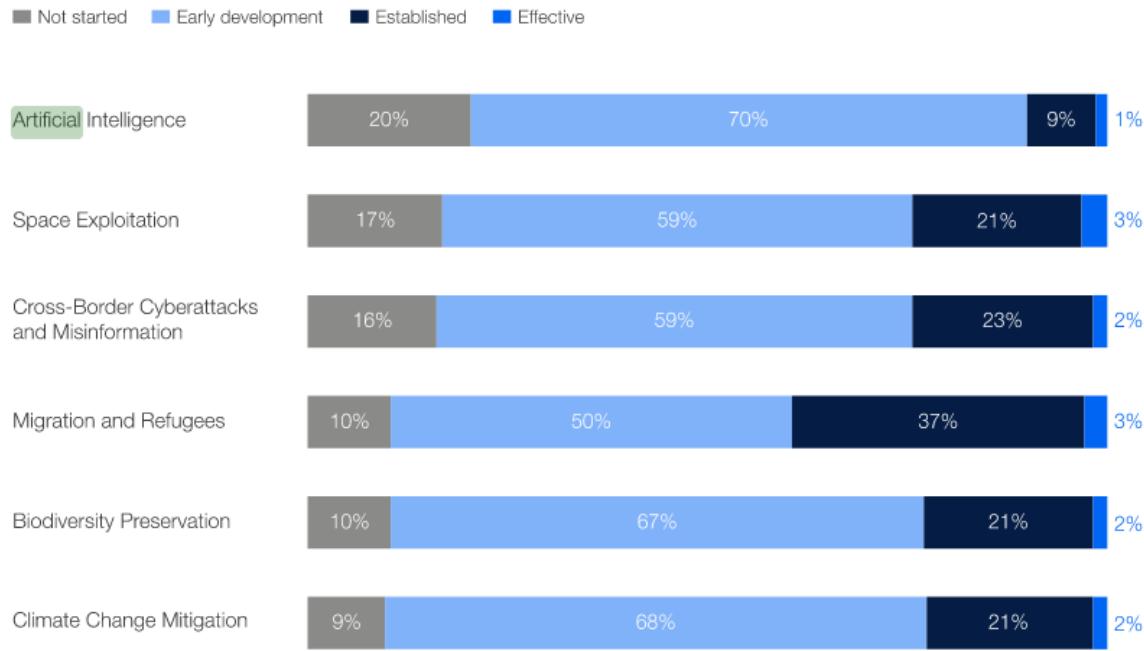


Figure 1.3: Current state of international risk mitigation efforts in each area (GRPS 2022) [2]

The climate is an indeterministic and chaotic phenomenon. Even a simplistic and low-dimensional convection model like the Lorenz system had chaotic states [4]. In the case of climate, its dimensionality is unknown, and the number of underlying variables is too many. The dynamics of observable climate variables like temperature, wind patterns, relative humidity, and precipitation are guided by the coupled dynamics of numerous such unknown and underlying variables. Like any complex system, the collective behaviour of the system is richer than the individual interacting variables and therefore is very challenging to model accurately [5]. We must devise policies and plan accordingly for adverse conditions in the future.

## 1.2 Rise in extreme climatic events over central India

Climate change is one of the most serious issues facing humanity and research groups around the world are extensively studying it. In this process, climate data has accu-

mulated at a much faster rate than has been possible to analyse and understand it. Roxy et al. conducted a study on recorded gridded rainfall data. They found that extreme climatic events, such as floods and heavy-rainfall-induced landslides, have increased by 200% over the central Indian subcontinent. The study also revealed that the graduality of the monsoon rains and annual rainfall has decreased while the frequency of these widespread extreme events has increased. [6]

Global Climate Models (GCMs, also called Earth System Models) are used to perform future projections of changes in climate in response to human emissions. These are simulations of the driving factors of climate through sea, land and atmosphere interactions which provide estimations about climate variables in the future.

While climate models have improved tremendously over the past five decades, one of the major challenges they face is that their spatial and temporal resolutions are still too coarse compared to what is necessary for climate action at regional and local scales. Spatially and temporally localised yet significant events, like extreme precipitation, tropical cyclones, wind pattern, cloudiness etc., often occur on scales that are smaller than the typical resolution of GCMs, and therefore their utility is limited. GCMs simulate the climate dynamics by considering the complex interactions between various components in the environment like the atmosphere, topography and GCMs simulate the climate dynamics by considering the complex interactions between various components in the environment like the atmosphere, topography and hydrosphere. The estimated differential equations behind climate dynamics are arduous to solve, and analytical solutions seldom exist. So scientists rely on brute force numerical integration methods [7], which are computationally quite expensive. The development of better computational facilities can be used to generate higher-resolution data. We propose an alternate method of deriving finer-gridded data from coarse outputs of existing climate models.

Deep learning is a field of intense research at these times. They have been successfully employed in various areas that could not be efficiently dealt with through traditional explicit coding techniques like spam detection, optical character recognition, face detection, analysis of medical data, computer vision etc.

State-of-the-art Deep learning techniques also have high potential in the analysis of the vast deluge of climate data that is being generated worldwide [8]. Deep learning techniques could potentially help in multiple ways. They can help identify complex relationships between climate variables (i.e. parameterisations) which are otherwise

impossible to discern through conventional methods.

The central objective of this thesis is to focus on the problem outlined above. Specifically, to understand and apply ML techniques to the output of GCMs in order to reliably derive high-resolution spatial information from the (relatively) coarse resolution of climate models. This report covers several Deep Learning techniques that are used for deriving high-resolution projections of low-resolution images.

Deep Learning needs to be better utilised in the field of climate research. This thesis aims to bridge the gap between climate research and deep learning, specifically its computer vision applications. Open-access google colab jupyter notebooks are designed so that even those who may be unfamiliar with the deep learning techniques can freely utilise them for various purposes.

### **1.3 Traditional statistical downscaling**

Statistical downscaling attempts to derive a statistical relationship between historical GCM predictions and the observed gridded data. It tries to infer from the given sample, assuming a functional relationship between data and predictions. [9] Some of the successful statistical downscaling are Bias-correction and Statistical Downscaling (BCSD) and Generalised Analog Regression Downscaling (GARD) [10]. BCSD originally used CMIP5 data and observed data sets. They mostly relied on linear interpolation and bias correction.

Multivariate Adaptive Constructed Analogs (MACA) [11] produced slightly better results than the earlier linear statistical bias correction methods by associating a combination of observed small-scaled patterns. But it could not account for all time scales and was heavily dependent on the kind of training data used. Such simplistic modelling needs to generalise and have higher accuracy for application to extreme weather events that are localised. In general, the traditional approaches have several shortcomings that could not be resolved even by using a larger sample since this would also increase the variance of the model. A model with high variance is heavily dependent on the training set used, and even small additions to those will diversely affect the performance and output of the model.

## 1.4 Artificial Intelligence

AI or artificial intelligence is a term used to denote smart algorithms that can solve complex problems that are limited by the human capacity to process vast amounts of data. The contemporary research in this field ranges from simple narrow AI models for specific tasks to Artificial general intelligence (AGI).

**Machine learning** is a type of AI problem that is used to do tasks such as classify, regress and predict the outcomes of new data once it has been trained within a training set. They were a revolutionary approach to dealing with problems where we cannot explicitly code every variable and its driving equations. The advantage is that they can learn from the iterative training.

The algorithm is initiated with a simplified model, such as a polynomial regression with arbitrary coefficients, which we call weights. The algorithm is repeatedly trained with thousands of training instances, and the model is penalised based on how far the predicted output is from the actual output. Each penalisation is done through back-propagation using an algorithm that adjusts its weights and parameters to give a reasonably accurate model in the end. This model can now be used to predict unknown data. The disadvantage is that they are incapable of handling large and diverse data sets as well as unlabelled data. This is because the initial algorithm is still chosen by an intelligent guess and trained, hoping that it would fit the data well. When the complexity of data increases, even polynomial regressions are incapable of generating a good model.

Deep learning is a type of machine learning technique specifically designed to tackle the above issue. It is inspired by the structure of the brain where simplified neuron units interact with each other to show diverse collective behaviours.

Just like that, deep learning architecture contains many different special regression units called neurons that are grouped to form layers, and the layers themselves regress the data sequentially.

This approach of using neurons as a collective is a game-changer compared to simple machine learning. Deep learning, unlike machine learning, performs better when the training set is large. Deep learning architectures designed as multiple layers of interacting neurons can predict more accurately but also do take more time to train than simple ML models.

Unlike machine learning, the output between the individual hidden layers is mostly indecipherable and only makes sense within the model. Thus, the models can learn the patterns and best features capable of accurate predictions without requiring human intervention to supply the features like in ML problems. The increased complexity of

deep learning algorithms makes them very slow on CPUs. Therefore parallel processing using specially designed hardware like GPU has been traditionally used to boost the performance of Deep learning algorithms.

There are two kinds of problems in Deep-Learning. Supervised and Unsupervised learning. Supervised Learning is used when we have labelled training data, i.e. the output and input pairs are available. We supervise the algorithm to create a functional mapping between data variables and labels. Unsupervised Learning is a different class of problems where only data is provided without labels. The aim is to find underlying structures and patterns within the data, for example, to cluster them into features.

Our problem with creating models that can map low-resolution images to high-resolution images generally comes under the domain of an unsupervised learning problem. But converting an unsupervised problem into a supervised learning problem by generating labelled training data is also common.

## 1.5 Neural Networks

For the purpose of downscaling low-resolution data, a good algorithm must be enabled to properly identify and process the features at several different spatial scales.

**Simple Artificial Neural Networks (ANN)** algorithms were experimented with for image classification tasks which were one of the successful applications of pattern-recognition algorithms. This was applied in the case of OCR. They worked by directly feeding the pixel space into a dense neural network. Therefore it deals with the flattened representation of multi-dimensional data into 1D arrays for feeding into the neural network while it learns the parameters. Such representation and further training pose several difficulties and disadvantages.

Even for a simple RGB image, if the Simple ANN is a dense neural network that is fully connected, it would have an immense amount of weights that must be trained for it to produce accurate outputs. Thus it is not computationally efficient.

A densely connected ANN also fails to distinguish between data points that are spatially close from those that are far. This also deteriorates its pattern recognition capability. Another issue is that in an ANN or MLP(Multi Level perceptron), every pixel is considered separately, and the information on spatial correlations between the points is also lost. Spatial correlations are significant and inevitable because the pixels that are close to each other in a 2D image are crucial to detect the patterns

and features that make that image meaningful.

ANN that can detect a handwritten digit written in the centre of the image fails to recognise the same digit if it is shifted by a few pixels to the side. ANNs are not robust to spatial translations. This is also an undesirable trait that leads to high variance in the algorithm.

**Convolutional Neural Networks (CNN)** algorithms create feature maps from the given input and preserve the original dimensionality of the image during this process. Convolutional filters detect and map the features in an image. Different features are automatically detected from the images to create a convolution kernel that is operated over the image throughout at an arbitrary stride to create the final array that represents the presence and location of that particular feature in the image. The output of the convolution operation is called a *feature map*. We can stack multiple convolution kernels or filters to simultaneously and independently compute that many feature maps and aggregate them to form more complex features. This is because the convolution operation that happens within the neural network can detect patterns in all spatial scales in a multi-dimensional input data set.

For further simplification, we could use any pooling method where a pooling operator slides across the feature map that summarises the feature maps to be trained. The lower the number of feature maps, the lower the number of feature maps that must be trained. Pooling also helps in increasing the noise tolerance and, equivalently, minimises overfitting.

We must also pass the feature maps that we created into an activation layer. We avoid using sigmoid or softmax activations because they have a constricted output space. This means the derivatives will get smaller and smaller during training and will make the network very hard to train.

The disadvantage of using linear activation functions is that, no matter how many of them there are, the whole group just has the effect of a single linear activation that is a combination of all the components.

We use an alternate activation function called a Rectified Linear Unit (**ReLU**). This is a filter that when trained helps identify and give weightage to the features that are relevant and also skips the unimportant features. This also helps to make the training faster. In the case of ReLU, the non-linearity of the ReLU activation function in between layers ensures that the layers perceive the variances and details in the input data.

After the identification of feature maps that contain all the relevant information about the patterns and their localities within the input, they can be flattened to a 1D array and passed on to fully connected ANNs for training. The advantage is that the number of weights to train is greatly minimised compared to the former approaches. We may use the softmax function for feature classification in the final layer.

Another advantage is that aside from some hyper-parameters to represent the network architecture, that is, the convolutional layers that perform as filters learn every important feature depending on the problem by itself through back-propagation and repeated training to maintain the best-defining features and give less priority to weak features.

Thus, the full deep-learning neural network first extracts the patches and feature maps from the image using convolutional layers, mapping them to high-resolution patches via a non-linear activation like ReLU and finally averaging the higher resolution patches to make the reconstructed version of the input image.

A variety of deep learning algorithms are discussed in the theoretical part with some real-image downscaling examples. All these models were pre-trained, and we selected the most capable model and applied it to climate data. The methodology will mostly include details of preparing training and evaluation of the algorithm by their performance of GCM. We will include colab notebooks with which climate data can be downscaled by anyone as well as the source code to reproduce the work.

# Chapter 2

## Theoretical background - Downscaling models

Here, we first review and discuss a variety of image downscaling models. Then we implement them and explore their capabilities, things that could be further improved, as well as some test cases for general images. Many of the theoretical ideas that will be discussed in the first section on the SRCNN model also apply to the rest of the models too.

### 2.1 Super Resolution Convolutional Neural Networks SRCNN

SRCNN [12] is a Deep-learning based implementation that is capable of fairly accurate mapping of low-resolution input data into high-resolution outputs. This super-resolution model, unlike sparse coding methods, does not explicitly learn the dictionaries. Those functions are performed under the hood using hidden layers in the Neural Network. The general disadvantages of statistical downscaling as well as sparse coding methods are that they have high variance, as we have seen in the introduction section. The larger the data set, the worse their performance. High dependence on the particular instances of training examples is not a desirable trait in such algorithms, which are expected to be capable of doing general super-resolution tasks. SRCNN overcomes this and even works better with a more extensive training set.

### 2.1.1 Training

First, high-resolution images are smoothed by applying a low-pass filter like Gaussian blur. This removes the high-frequency data from the image and produces a low-resolution image. This blurred version of the high-resolution image is again upscaled back to the exact dimensions as the original high-resolution image using classic bicubic interpolation.

The pair of this smoothed image and the desired output being, the HR image, make one instance of the training set.

Then, these are passed on to the SRCNN. The loss is calculated from the MSE of the resultant output and original high dimensional image using another MSE and update the weights of the network using a gradient descent algorithm and again running the sample using backpropagation. This process is done one by one for every training example.

Reducing the image size and advantages of using convolutional filters over standard densely connected networks helps to reduce the computational cost for the millions of back-propagations and thousands of training epochs. The training process involves the filter layers learning to recognise better, more complex features the deeper the neural network goes.

Several architectures were experimented with in the paper, aiming to maximise the PSNR of the final output.

### 2.1.2 Activation functions

The ReLU activation function has two parts. It outputs zero, i.e. it remains in the off state for every value below the threshold, and it is linear above the activation threshold. Since its derivative is zero under the threshold and a positive constant beyond it, it plays a vital role in updating the weights of the network in the right amount. Therefore by default, generally, ReLU is preferred over any linear activation function for the non-linearity that it introduces in the network.

But there are cases of using ReLU in some network problems that have high learning rates, and they cause an issue known as the dying ReLU problem [13], where entire neurons become unresponsive after a number of back-propagations.

To combat this, we may use a leaky ReLU one, which does not completely disregard the simulations by modifying the left-hand side with a straight line of low negative slope. Another option available is to parametrise the range below the threshold

as a linear function of a parameter arbitrarily close to zero. The slope representing the leakiness of the ReLU is a parameter that is also learnt by the model in the training process. Such a modified activation function is termed parametric-ReLU or **PReLU**.

### 2.1.3 Image quality evaluation metrics

Along the process of training and testing, we must perform different evaluation techniques to calculate errors and apply them to algorithms such as backpropagation so as to be able to set the weights properly. There are several strategies that could be used to compare the image quality and thereby estimate how accurate a downscaled version is to the original image. This helps in comparing different models.

#### Mean Squared Error MSE

For SRCNN, we use the standard mean squared error (MSE). In order to compare two images  $f, g$  of equal sizes of resolution  $M \times N$  is defined to have a Mean Squared Error of:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (2.1)$$

where  $f_{ij}$  and  $g_{ij}$  represents the  $ij^{th}$  element of the image array.  $f_{ij}$  and  $g_{ij}$  are the respective intensity values of the same pixels in the two images that we compare. In the case of an RGB image, it becomes the mean of the errors of all the three channels:

$$MSE = \frac{1}{3} \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij}^R - g_{ij}^R)^2 + (f_{ij}^G - g_{ij}^G)^2 + (f_{ij}^B - g_{ij}^B)^2 \quad (2.2)$$

#### PSNR

Suppose there are  $n$  bits to code the image channel values per pixel, each of the binary bits contributes a factor of 2, and the peak value of the intensity of each pixel would be  $I_{max} = 2^n - 1$ . This value is 255 for an 8-bit pixel. Now we may define **Peak Signal to Noise ratio (PSNR)** as follows:

$$PSNR = 10 \log_{10} \left( \frac{I_{max}^2}{MSE} \right) \quad (2.3)$$

## SSIM

Structural Similarity Index measure (**SSIM**) is not based on simply quantifying pixel-wise differences like MSE or PSNR. SSIM also considers the spatial correlation between nearby pixels and therefore is a perceptive metric that can measure structural variations. It is also built on the idea of using subsections of the full image to locally estimate values. It could also use overlapping sections for better structure perception but that quickly becomes highly complex. The literature suggests to use semi-overlapping windows to calculate SSIM by trading off between computational cost and accuracy.

Let  $x$  and  $y$  be two windows of equal sizes,  $N \times N$

$$SSIM(x, y) = \underbrace{\left( \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \right)}_{\text{luminous similarity}} \underbrace{\left( \frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \right)}_{\text{contrast similarity}} \quad (2.4)$$

The terms  $c_1$  and  $c_2$  are used to prevent ambiguity when the sub-sections are entirely either dark or uniform.

SSIM has values in between -1 and 1, the closer it is to 1, the more similar the images are.

The windows  $x$  and  $y$  are allowed to stride throughout the images  $x$  and  $y$ , and the final value of SSIM is taken as the average of these local SSIM values.

## 2.2 Generative Modelling

In generative modelling, training examples taken from a set are used to train a model that can generate fresh samples that resemble the one in the given set. There are situations where the amount of data and the data diversity itself are limited. In such cases, it becomes important to not only augment the existing data but also to generate fresh samples that are truthful to the training set but with sufficient variability. For example, the CelebHQ data set is widely used for any kind of facial super-resolution as well as face recognition deep learning problems. This data set and many others like this were developed with Generative modelling.

### 2.2.1 Variational Auto Encoders VAE

This is not a generative model but another architecture that is used within generative models. VAE is used to find and tune the latent variables. They are the variables in an input-output training set that are hidden/ not apparent.

**Auto encoders** is an unsupervised learning approach that creates low-dimensional latent vectors or latent spaces by processing raw data. Essentially the raw data is mapped into a compressed representation of the data.

These compression techniques can help increase computational efficiency, especially when dealing with image analysis. But the low-dimensional output also needs to be a good representation of the original data without losing the necessary underlying features of the raw data. For this, a convolutional decoder architecture can be used to map the compressed representation back to the original data set. A loss can be defined as a difference between the raw data and the decoded output to create a good encoder. Thus an auto-encoder encodes the necessary information within data into a smaller latent space.

Variational Auto-encoders do the encoding to create a stochastic sample instead of a deterministic latent space. Thus the encoding process is made to learn the probability distribution because of the stochastic sampling, back-propagation through the sampling layer is not possible.

Following the concept of Variational Auto-encoders, **VDVAE Very Deep VAE** was proposed. But it requires a large amount of training time which can only be partially resolved through transfer learning if pre-trained data is available for this purpose. Still, the proposed architecture of VDVAE is immense and has 180 million weights that must be learnt.

### **2.2.2 GAN based approaches**

There are two main parts to this Generative Adversarial Network. One is a generative network that creates random latent vectors representing the images and a Discriminative network. The Discriminative network is fed with real images and outputs from the generator. It is a binary classifier that is tuned to learn to identify whether the input is a fake image or a real image.

The adversarial part deals with both generative and discriminative components competing against each other. The generator is trained so that the image produced is more and more realistic to fool the discriminator, while the discriminator is trained to get better at recognising the fake images with each iteration.

Therefore the Discriminator and Generator are forced into a zero-sum game where each of them tries to fool the other network. The generator is penalised so that it generates a realistic image that could be passed as a real one when the discriminator is wired to distinguish between the generated fakes and the real ones accurately.

### 2.3 Visual Geometry Group-VGG

---

Therefore if a real image is identified as fake, it counts as the discriminator error, and if a fake image is detected as fake, the generator is penalised.

As the training cycles pass on, many epochs later, both the generator and discriminator progressively get better at their tasks, and we stop the algorithm when an arbitrarily good discrimination capacity is achieved.

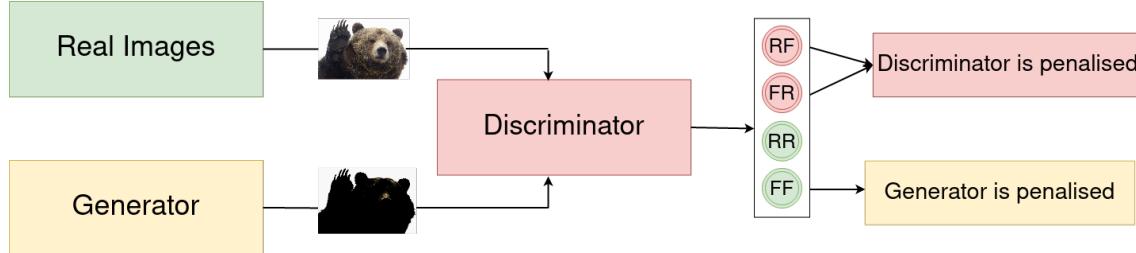


Figure 2.1: Illustrative chart of GAN Architecture

## 2.3 Visual Geometry Group-VGG

VGG networks are another class of deep CNN models which are especially good with feature extraction and recognition of the distinctive features in images. Therefore a pre-trained VGG is first used on the raw data. This ensures that the discriminator works within the rich feature space as opposed to the raw pixel space. Basically, the VGG architecture contains stacked-up convolutional layers. There are max-pooling operations in between to reduce the dimensionality further and extract only the necessary features, thereby reducing the computational costs. The dense layers in between are the ones that take most of the run-time during training.

VGG models pre-trained on large data sets can recognise the patterns in the image, and this is what is fed into the discriminator, not the entire image. Thus we can reduce the computational expenditure in the case of general super-resolution, but for climate down-sampling, data sets must be trained separately first to apply VGG. This is called transfer learning. For all GAN models, we need pre-trained VGG networks. There are different kinds of VGG architectures available, like VGG16, and VGG19 having 16 and 19 convolutional layers each.

## 2.4 Super Resolution GAN type models (SRGAN)

In Super Resolution GANs [14] (**SRGAN**), pre-trained weights of VGG19 having 19 convolutional layers are used. This is also used in computing a part of the loss function. The generator and discriminator networks are trained by penalising each

other in an adversarial framework and in the end both the generator and discriminator will be sufficiently good.

SRGAN Model utilises two special kinds of ReLU activation functions for its two components to overcome the limitations of normal ReLU, including the dying ReLU problem. The original algorithm is designed with a Discriminator requiring leaky ReLU and the Generator with a PReLU (parameterised ReLU with an arbitrary negative slope for values below the threshold).

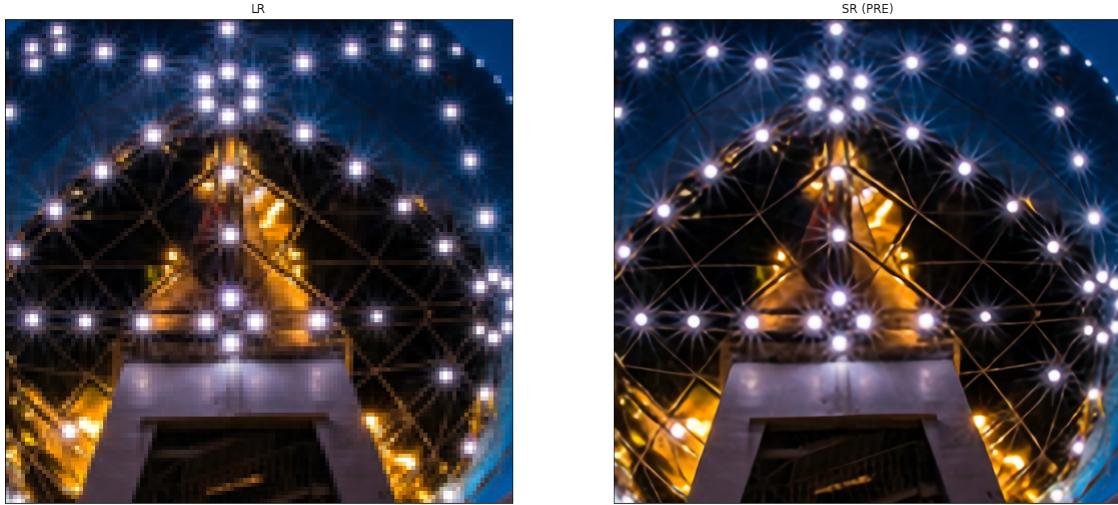


Figure 2.2: Comparison of LR image and SR image downscaled by x4 using SRGAN

The **generator network** has convolution layers, batch normalisation and PReLU. The output is passed into a number of residual blocks in sequence, and at the end, upsampling blocks containing pixel shuffle layer up-samples the image by increasing its width and height.

The discriminator helps the generator to create better images whenever the generator's output is detected as a fake image.

Previously we compared images pixel-by-pixel using MSE. Since images can have low pixel-wise errors and yet lack the visually perceived similarity, instead of focusing on individual pixels, SRGAN utilises a new perceptual loss function to generate photo-realistic images.

For the content loss to work better, we utilise the pre-trained VGG models and compare the VGG feature maps with the generated image's feature maps rather than directly comparing pixels. The rest of the loss is contributed by the discriminator that compares it to real images. The adversarial loss forces the generator images to become more and more realistic. A weighted sum of both of these gives the perceptual

## 2.4 Super Resolution GAN type models (SRGAN)

---

error that is to be minimised.

Later **Enhanced-SRGAN ESRGAN** [15] model was published. ESRGAN maintained the basic architecture of SRGAN and the idea of using GANs for downscaling while introducing some modifications. In the generator, it removed the batch normalisation layers because unwanted artefacts were created in the output image. The ResNet blocks were replaced with new residual-in-residual dense blocks by stacking them in a dense network. Also, in place of adversarial loss, instead of seeing the probabilities, the ESRGAN uses a relativistic discriminator.



Figure 2.3: Comparison of LR image and SR image downscaled by x4 using ESRGAN

In the figure, we can note that ESRGAN produced realistic high-frequency details like fur when they were not present in the low-resolution input.

In the context of real-world image upscaling, ESRGAN was further developed into **Real-ESRGAN** [16]. The algorithm has a pixel unshuffler that creates a latent space of the original image before ESRGAN processes it. As explained in the case of VAE, this reduces the dimensionality of the image and therefore is faster than the regular ESRGAN algorithm.

Almost all models before this have used bicubic downsampled images, which is ideal and unlike real-world LR images. Real-ESRGAN performs much better at downscaling real-world images, unlike its predecessors, because it does not assume deterministic downsampling for generating training sets. Real-ESRGAN utilises a more complex second-order non-linear degradation model and added noise artefacts to resemble an

actual low-resolution image better.

Further, it was noted that Real-ESRGAN could find a better trade-off between sharpness and artefact suppression if trained on sharpened real-world photos.



(a) LR



(b) Real-ESRGAN

Figure 2.4: Comparison of LR and SR image downsampled by x4 using Real-ESRGAN



# Chapter 3

## Inferences from various models

Several experiments were conducted with different downscaling models on real world images.

We compared an SRCNN model trained on a small T91 data set and compare its outputs with the same network trained on the BSD100 model. We use 20:80 ratio for testing and training sets. Then while giving the same low-resolution image to upscale, we see that the one trained on BSD100 is qualitatively and objectively the best. It has sharper corners and more contrast. BSD100 data set contains a variety of different images and is larger in size than T91. From this, it can be concluded that for general super-resolution tasks, it is always better to train on diverse and larger data sets, especially in the case of SRCNN.

SRCNN is a model that is fitting for training on a small data set. Its results are quite dependent on the training set itself. Therefore the training set needs to have more inherent diversity to get better results while testing with images outside of the training set.

But GAN models all require large training sets and in cases where the data available is limited, data augmentation and other methods of increasing diversity in the data set are employed to construct the training set. They have millions of weights and biases to be trained and therefore also computationally intensive and extremely slow to train. This is also because, after every training epoch is completed, both the generator and discriminator have to be updated based on the losses of the model. But once trained, GAN models have the upper hand as can be observed here.



Figure 3.1: Bicubic blurred image of a tiger



(a) SRCNN

(b) ESRGAN

Figure 3.2: Comparison of LR and ESRGAN

The most significant update Real-ESRGAN has compared to ESRGAN is that the procedure to prepare the training set is slightly different. When ESRGAN uses a simple bicubic algorithm to reduce the dimensions of the input image, Real-ESRGAN introduces randomness and other JPEG compression artefacts that are common to see in real-world low-quality images.

But this particular change is not expected to have the intended effect in case of climate downscaling. First, the resolution of the image we are going to use is negligible compared to the images Super-resolution GAN models are trained for. Since there is no chance of high-frequency JPEG compression artefacts occurring in our training set, going with Real-ESRGAN could introduce undesirable artefacts in the final model. The RRDBNet models already have the disadvantage of adding artefacts to the super-resolved image, and further increasing the low-resolution artefacts in training data would only worsen its performance. Previously, different works have been done on similar models like SRGAN [17]. So we decide to go with the best option that is, ESRGAN which also has not been applied specifically to central Indian precipitation data yet.

# Chapter 4

## Data set

Inspired by the paper from Roxy et al., we decided to use the rainfall data from the central Indian zone. The gridded data set available from Indian Meteorological Department was prepared by interpolating station data [18]. The data is at a spatial resolution of  $0.25^\circ \times 0.25^\circ$  and at a temporal resolution of 1 day. The whole gridded data set from the year 1901 to the year 2022 were used for training.

### 4.1 Data set downloading

Using a Python module imdlib from IMD, the data set was downloaded. The obtained data is in the format of *.grd*. It was converted and saved separately as NetCDF data files for each year. For the purpose of training ESRGAN, we also require a training set. We chose the whole data from 2005 for this purpose.

### 4.2 Kernel width experiments

The entire data set is preloaded before each training. Each time-slice of the data set is blurred with a bi-cubic image resize function. We require  $1^\circ \times 1^\circ$  data, so we resize the data by a scale of 0.25. Different **kernel width** parameters were experimented with. The larger the kernel width of the resize function, the lesser will be the extremes. But if kernel width is too low, the extreme values of individual grids are unrealistically high. We iterate through different kernel widths and compare the blurred images with the CMIP6 precipitation projections to choose the kernel width which is just right. This is because the later validation of the model will be performed by one such CMIP6 projection.

We chose kernel width = 1 for preparing the low resolution inputs to the algorithm.

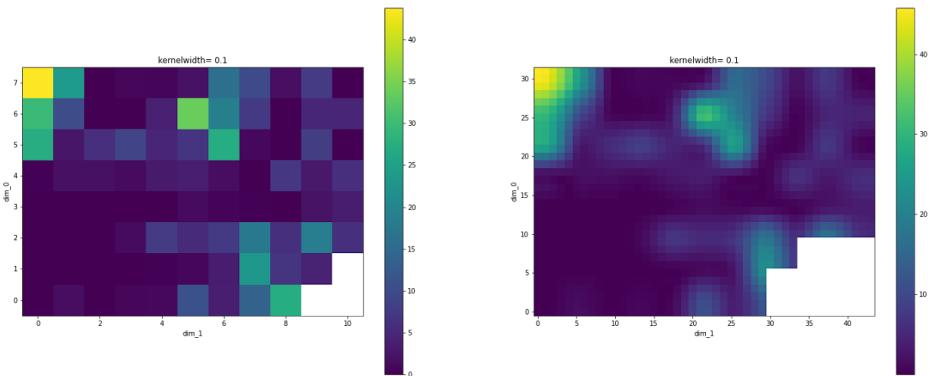


Figure 4.1: a)left: bi-cubic upscaled b)right: bi-cubic downscaled

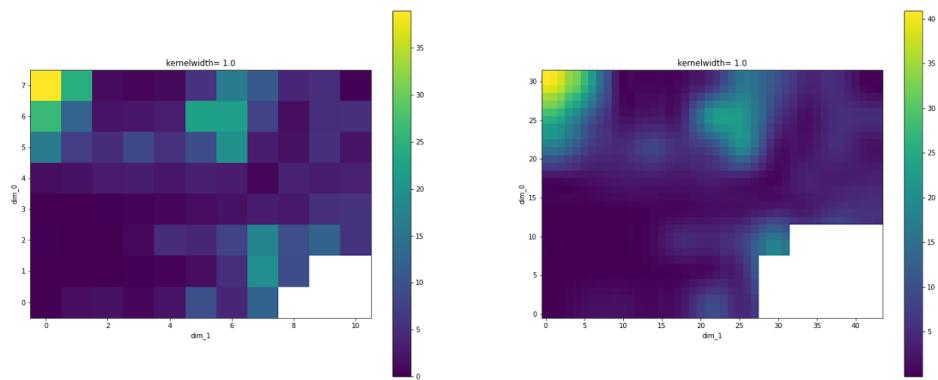


Figure 4.2: a)left: bi-cubic upscaled b)right: bi-cubic downscaled

## 4.3 Preparing training set

Using the CUDA data loader, we pre-load the entire training and testing data set into memory. This increases the speed of training. Loading data sets involves, first cropping the gridded data set to the central Indian region in the range  $76^{\circ} - 86^{\circ}$  longitude and  $18^{\circ} - 26^{\circ}$  latitude. After this, the NaN values of this region are filtered away. Then we convert it to a numpy array and load it iteratively as a training set using CUDA pre-fetcher. This range of 32x44 values is important because a pre-trained VGG network is designed for image sizes of a minimum of 32x32. During this loading process, even more variability is introduced by random vertical or horizontal mirror flips, random image rotations etc. This is called **data augmentation**. This is a surefire way to make the model exposed to more variability in the data set so that it performs well in the validation stage. We further employ a random shuffler that randomly shuffles the training data set during loading. This is an important step to have a guaranteed consistent performance throughout the training. It prevents over-fitting. This shuffler is not used in the case of testing because this stage does not adjust the weights of the models but only helps to choose one of the good epochs from many.



# Chapter 5

## Training

Training is divided into two parts. First, we train the RRDBNet model which makes the Generator of ESRGAN until we get sufficiently good PSNR values. After this, generator will be used to train the discriminator network in an adversarial fashion.

### 5.1 RRDBNet

This is a deep convolutional network which has additional batch normalisation layers and leaky ReLU within. We use 23 RRDB blocks to form the generator. RRDB was trained for 1500 epochs and the performance has saturated.

### 5.2 ESRGAN

We used Adam Optimiser just like the original ESRGAN paper. We initialised the learning rate with  $2 \times 10^{-4}$  and the beta parameters of the Adam optimiser were chosen to be 0.9 and 0.99. The model EMA decay parameter was 0.99998.

We additionally had a learning rate scheduler set up so that it would reduce the learning rate after some key stages are passed in the total number of epochs. The gamma for the learning rate scheduler was chosen to be 0.5.

These settings were identical to those used for training RRDBNet also.

ESRGAN training involves training a discriminator from scratch and resuming the generator training. We use Adam Optimiser to optimise the learning rates. Further, learning rate schedulers were employed to reduce the learning rate by a fixed amount every fixed number of epochs for 4 times throughout the training. Even for a less number of epochs, we could get great results, and the best PSNR of the images was **59.2157**, and SSIM was **0.9031** during testing at the end of the run.

## *5.2 ESRGAN*

---

Using this pre-trained model, we can give any CMIP6 NetCDF file to return a high-resolution version of the same.

Once the training is done, we test the model's performance on a smaller data set. Our training set had 365 images. The best epoch is chosen by the one which returns the best PSNR and SSIM value. This model is backed up and saved for direct application to climate data.

# Chapter 6

## Results and Discussion

### 6.1 Key training findings

We trained it on an NVIDIA A100-SXM4-40GB GPU using PyTorch. In the original ESRGAN paper, they used a batch size of 16 images per training iteration. For general images, this utilises almost all memory available in the GPU. We took another route. It was observed that we could afford to increase the batch size by a hundred-fold. This not only improves the speed of the training but also permanently improves the quality of the model after each epoch. This is because each batch captures more of the gradients and variability in our data set rather than small batches. This was a new finding in this project. The same technique could be coupled with multi-GPU training and could be employed even more efficiently.

The model began to saturate after a couple of epochs, giving the best PSNR value of **59.21579** and the best SSIM **0.9031** after testing.

### 6.2 Validation of the model

To get an accurate estimate of the performance of this ESRGAN model, we choose a CMIP6 data set. Unlike IMD data, this is stored in precipitation flux indices, so we first convert them to the required units of *mm/day*.

The data set, which is at 25km resolution, is interpolated to make a low-resolution version. Now we downscale this and see how close it comes to the original data set. For comparison, we use MSE and PSNR. By averaging through the data set for 4 years, we could obtain the average **PSNR of 120.29dB** between the original and the down-scaled versions. Typically, PSNR values for general super-resolved images

## *6.2 Validation of the model*

---

range from 30-50dB, which means that the downscaled data is almost exactly close to the original data.

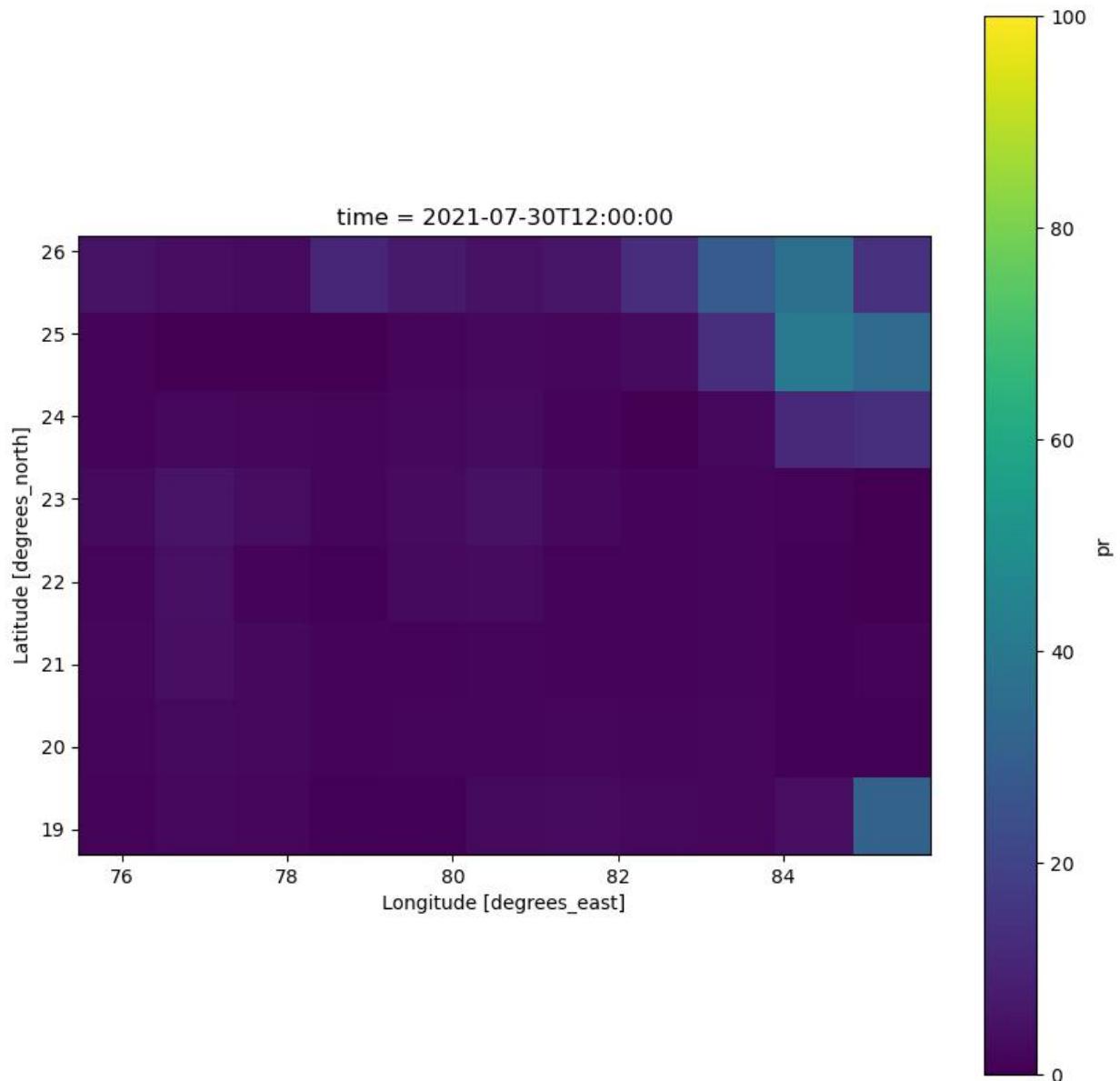


Figure 6.1: LR blurred CMIP6 data

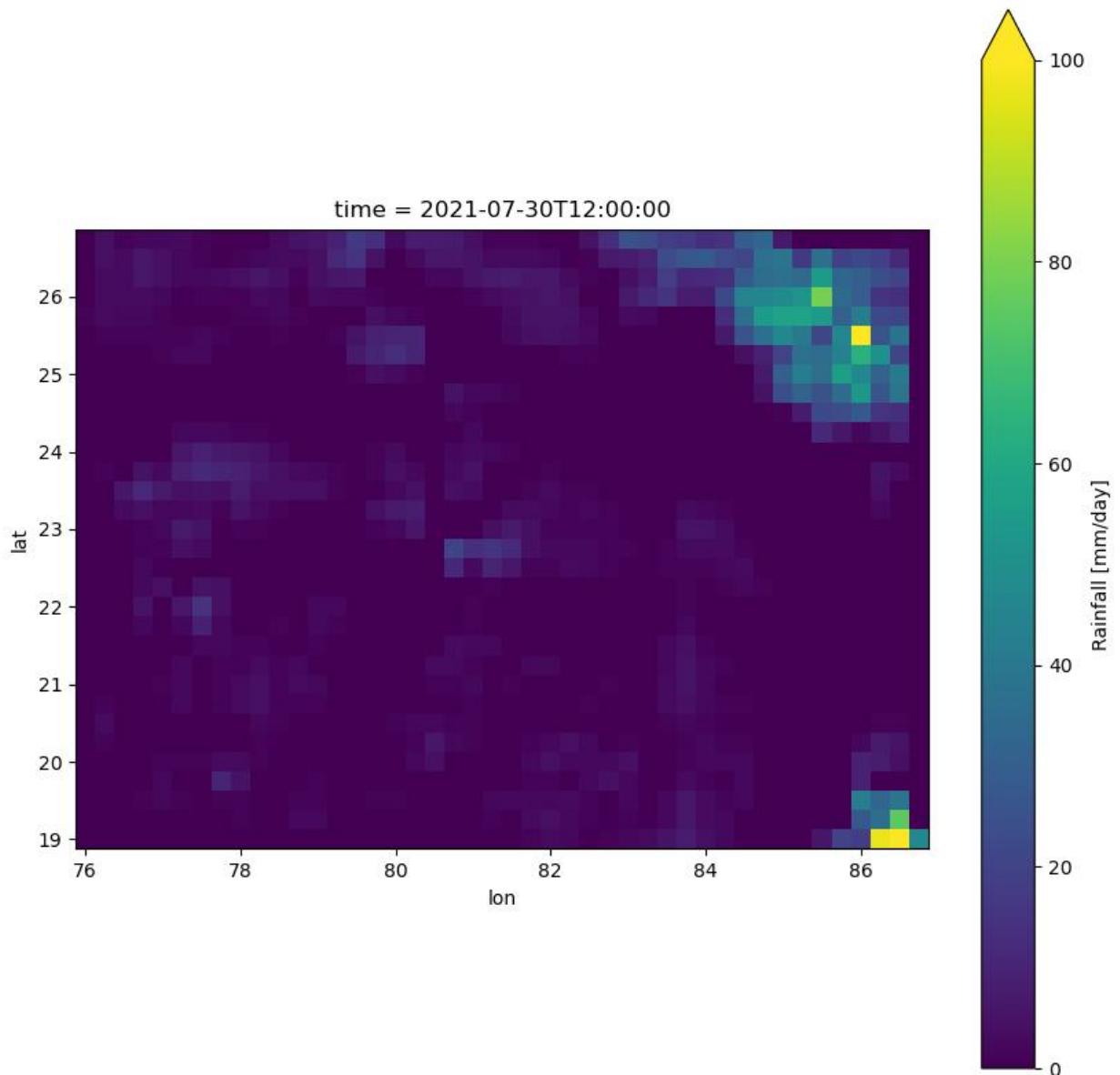


Figure 6.2: 4x Super-resolved CMIP6 data

In these images, we note that even though the low-resolution map does not contain any extreme events, the down-scaled version does contain localised and extreme

events. This is just what we require for studying extreme precipitation events.

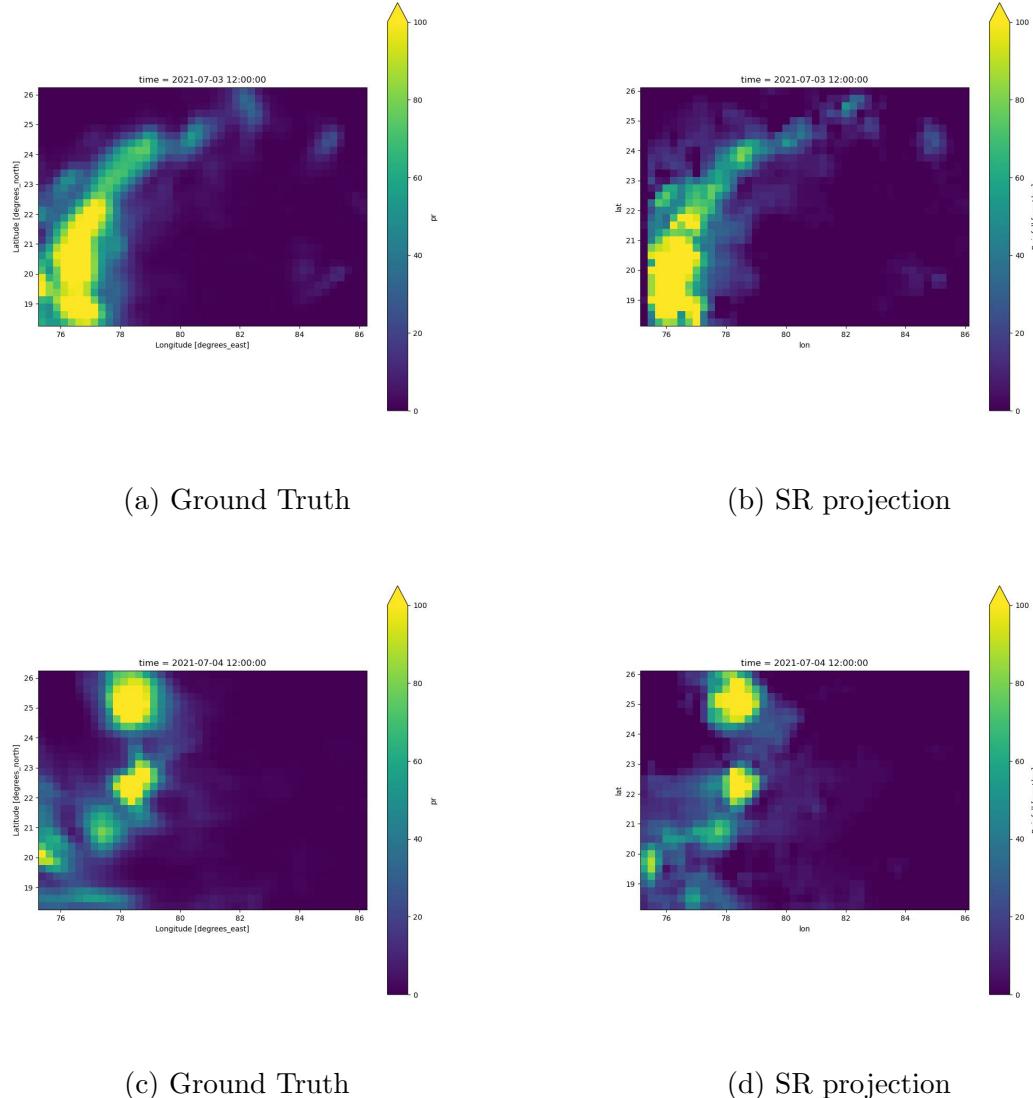


Figure 6.3: Comparison of Ground truth CMIP6 projections image and SR images developed by downscaling the LR x4 using ESRGAN

From the comparison of the ground truth images and also the generated down-scaled plots, we can conclude that giving more weight to PSNR observation rather than giving equal importance to SSIM would produce better results, especially when studying extreme events. PSNR is a pixel-by-pixel comparison technique and when we aim for the best PSNR, we can capture the extremes better. The SSIM values focus more on keeping the structural similarity and higher SSIM values increase the bloom effect around the extremes in the down-scaled images.

We now can choose any CMIP6 model to quantify the frequencies of localised extreme precipitation events in the future using this.

The source code of the project necessary to replicate the work is updated in the GitHub repository. It is designed for directly downscaling climate data, even for those unfamiliar with machine learning techniques. Currently, this is only for precipitation data, but this can be similarly expanded and trained for any climate variable.

## 6.3 Study of trends of extreme events from CMIP6 projections

Since we are focusing on just the central Indian region where the spatial variability is limited, we can afford to use constant thresholds for estimating whether an event is indeed an extreme event or not. This is following the methodology in the work of Roxy et al. [6] which had quoted the observation of World Meteorological Organisation [19] that the fixed thresholds for classifying extreme events are more reliable than percentile values. Here, we first generate the CMIP6 models and count how the frequency of extreme events changes in the period of 2015-2035. We could also compare how much different the mean number of extreme events is in a year in the original high-resolution vs the downscaled data we produced.

We are also choosing  $150\text{mm/day}$  as the threshold for considering anything an extreme event.

The CMIP6 models are usually available in the units of precipitation flux which is the volume of the water precipitated in a region per second. We first convert it to  $\text{mm/day}$  by multiplying it by the number of seconds in a day.

Now, we first create a  $1^\circ \times 1^\circ$  low-resolution map of this HR data with  $0.25^\circ \times 0.25^\circ$ . This LR data set is now downscaled 4x to give back the prediction of the model. We create such data files for all the days of the years 2015-2035.

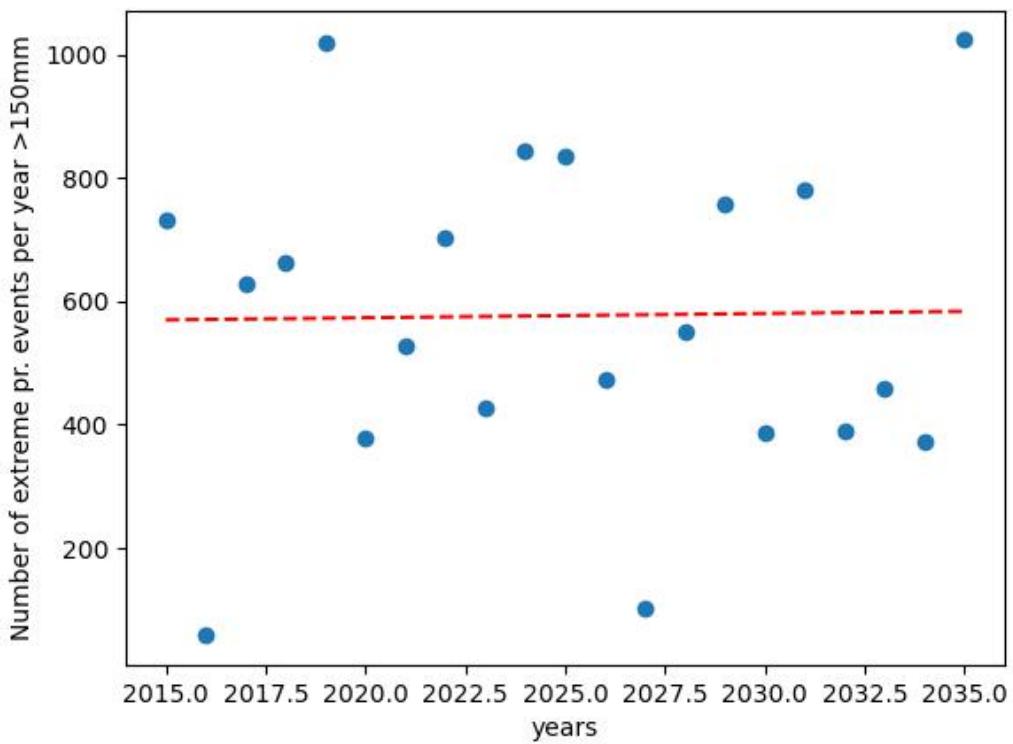


Figure 6.4: Plot of frequency of extreme events



# Chapter 7

## Conclusion

In this thesis work, we could experiment with various downscaling models. ESRGAN was chosen and trained on IMD precipitation data. After this, we evaluated the model, and it was used to downscale climate data from CMIP6 a GCM model. Finally, we also tried to see the trend on increasing extreme events. But based on the particular CMIP6 simulation we had chosen, we could see that the frequency of extreme events was slowly rising but not at a rate that Roxy et al. had observed. This is nothing conclusive, more accurate models could also be downscaled and could have results that match the results in [6]. We open-sourced the code for the work and make it available on GitHub so that anyone working in climate can make use of this model. We also provide interactive Colab notebook links where we could simply load a coarse NetCDF file for it to return a high-resolution NetCDF file.

Rest of the work to study the implications on sustainable energy, we could train it on cloud patterns and on temperature distributions. But this could not be completed due to time and computational resource constraints. GAN models themselves are very hard to train. We could further modify ESRGAN algorithm itself to make it lighter to train smaller zones like the central Indian region we had studied.

VGG models pre-trained on diverse data sets are absolutely necessary for minimising the training times and to produce better results. Having to train a VGG network from scratch takes up lot of computational resources and time. All these GAN algorithms uses some sort of VGG algorithm for feature extraction and calculating loss, but pre-trained VGG models for climate data is not available in public domain right now. Therefore we pre-train and these VGG-19 models will be updated in the Github repository so that it can be used by anyone conducting a similar study.

- Climate downscaling ESRGAN Google Colab notebook
- Github repository of climate downscaling



# Bibliography

- [1] Masson-Delmotte, V., P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou (eds.), *IPCC, 2021: Summary for Policymakers. In: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, pp. 3-32, doi:10.1017/9781009157896.001. ,5-6.
- [2] *Global Risks Report GRR 2022*, World Economic Forum (2022)
- [3] *Global Risks Report GRR 2023*, World Economic Forum (2023)
- [4] E. N. Lorenz, *Deterministic Nonperiodic Flow*, J. Atmos. Sci. 20, 130 (1963)
- [5] Gupta, Shraddha & Mastrantonas, Nikolaos & Masoller, Cristina & Kurths, Jürgen, *Perspectives on the importance of complex systems in understanding our climate and climate change—The Nobel Prize in Physics 2021*, Chaos: An Interdisciplinary Journal of Nonlinear Science. 32. 052102(2022). 10.1063/5.0090222.
- [6] Roxy, M.K., Ghosh, S., Pathak, A. et al. *A threefold rise in widespread extreme rain events over central India.*, Nat Commun 8, 708 (2017). <https://doi.org/10.1038/s41467-017-00744-9>
- [7] A.J. Broccoli, *Paleoclimate Modeling of Last Glacial Maximum GCMs, Reference Module in Earth Systems and Environmental Sciences*, Elsevier, (2014), ISBN 9780124095489,<https://doi.org/10.1016/B978-0-12-409548-9.09409-4>.
- [8] Reichstein, Markus, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. *Deep Learning and Process Understanding for Data-Driven Earth System Science.*, Nature 566, no. 7743 (2019): 195–204. <https://doi.org/10.1038/s41586-019-0912-1>.

## BIBLIOGRAPHY

---

- [9] Lanzante, John R. ; Dixon, Keith W. ; Nath, Mary Jo ; *Some Pitfalls in Statistical Downscaling of Future Climate* ; Bulletin of the American Meteorological Society, 99(4), (2018) ,doi:10.1175/BAMS-D-17-0046.1
- [10] Wood, A. W., Maurer, E. P., Kumar, A., & Lettenmaier, D. P. *Long-range experimental hydrologic forecasting for the eastern United States.*, Journal of Geophysical Research: Atmospheres, 107(D20), (2002) <https://doi.org/10.1029/2001JD000659>
- [11] Abatzoglou, J.T. and Brown, T.J., *A comparison of statistical downscaling methods suited for wildfire applications.*, Int. J. Climatol., 32: 772-780. (2012), <https://doi.org/10.1002/joc.2312>
- [12] Dong, Chao & Loy, Chen Change & He, Kaiming & Tang, Xiaonan., *Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence.* 38. (2014) 10.1109/TPAMI.2015.2439281.
- [13] S. C. Douglas and J. Yu, "Why RELU Units Sometimes Die: Analysis of Single-Unit Error Backpropagation in Neural Networks," 52nd Asilomar Conference on Signals, Systems, and Computers, (2018), pp. 864-868, doi: 10.1109/ACSSC.2018.8645556.
- [14] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi; *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*
- [15] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, Xiaonan Tang, *Enhanced Super-Resolution Generative Adversarial Networks*; doi:10.48550/arXiv.1809.00219
- [16] Xintao Wang, Liangbin Xie, Chao Dong, Ying Shan; *Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data*; doi:10.48550/arXiv.2107.10833
- [17] Kumar, B., Atey, K., Singh, B.B. et al. *On the modern deep learning approaches for precipitation downscaling.* Earth Sci Inform (2023). <https://doi.org/10.1007/s12145-023-00970-4>

## BIBLIOGRAPHY

---

- [18] Pravat Jena, Sourabh Garg and Sarita Azad, *Performance Analysis of IMD High-Resolution Gridded Rainfall ( $0.25^\circ \times 0.25^\circ$ ) and Satellite Estimates for Detecting Cloudburst Events over the Northwest Himalayas* (2020)
- [19] Tank, A. M. G. K., Zwiers, F. W. & Zhang X. *Guidelines on Analysis of Extremes in a Changing Climate in Support of Informed Decisions for Adaptation*, World Meteorological Organization, (2009)