Refining Coarse Projected Precipitation Mapping Using Statistical Downscaling by Deep Learning

Ahmed Al-Ali, *B.S. in Stat. Data Science,*

***Abstract*—We present a deep learning model for refining coarse-grained to fine-grained data using auxiliary spatial and temporal data sets in this paper. The approach we utilized is ConvLSTM-SR, which is a spatio-temporal neural architecture for statistical downscaling. State-of-the-art models are available at coarse-projected (in the order of 25-100's km) for this suggested technique to analyze and adjust on fine-projected forecasts (in the order of km). The current study performs daily downscaling of precipitation variable from a CORDEX output at 0.25 degrees (25 km) to 0.05 degrees (5 km) over one of the most climatically diversified regions, Central America. To create predictions at small projections , downscaling algorithms use data from a coarsely projected mapping. The usage of neural network-based designs is motivated by complex and interdependent local variables (e.g., precipitation across time). We demonstrate a considerable improvement over state-of-the-art baselines in terms of predicting statistics of severe occurrences and temporal seasonal relationships. We make all the algorithms and trained models available on GitHub to enable reproducible research, and data files may be downloaded through the UC Santa Barbara CHIRPS Climate Hazard association.**

I. Introduction

Downscaling techniques are used to generate data projections at HR, these are subdivided into two areas Dynamical downscaling (DD) and Statistical Downscaling (SD), DD takes into account the sub-grid processes into coarse resolution grids' boundary conditions. Atmospheric Physics, land and sea mass properties, and hydrological processes are examples of processes. While dynamical downscaling is beneficial for modelling precipitation events and localized phenomena, the results are very sensitive to boundary conditions [1]. SD attempts to learn the statistical relationship between coarse-projected maps to fine-projected maps, observations such as observed or remotely sensed precipitation. With assumptions of space and time stationarity, this correlation helps in generating fine-projected outputs from coarse projections. Statistical downscaling (SD) methods are used to establish empirical correlations between large-scale atmospheric variables e.g."predictors" and local or regional variables of interest e.g. "predictands" [2] . It is a technique widely employed to enhance the quantity of information included in satellite data. It entails reconstructing a high-coarse-projected observation from a fine-projected one. This subject is comparable to image super-resolution in the computer vision and pattern recognition domain which has made remarkable progress in recent years thanks to deep learning. SD is computationally efficient and may be used to a wide range of geographies [3]. The outcomes, however, are heavily influenced by the predictors chosen. Furthermore, statistical downscaled variables can provide an inaccurate image of the observed projections due to variance biasness and extremes [3]. accounted as non-linear perplexities.

II. Key Contributions

We present a Convolutional LSTM based Super-resolution approach toward statistical downscaling of climatic data from coarse-resolution to fine-resolution projections to get a high resolution observation data, which accounts for spatial and temporal dependence in space and time to target variable e.g. "precipitation in the present case".We note that the term Super-resolution is used to refer to the specific neural network based architectures. Term Super-resolution or SR is widely used in the computer vision architectures that has also motivated the choice of my architecture. Furthermore, we note that the resolution of final output is guided by the predicant datasets that are used for model training.

III. Data and Study Area

To elaborate more on the data itself, we gained access to the public available data at CHIRPS we have individual NetCDF files (extension .nc) for each year in the range 1950-2020 containing the low-resolution data at coarse projection and High-resolution data at fine projection for the years from 1980-2020 are stored in another single NetCDF file. Each of these contain spatiotemporal data: Each image represents a single day, with grids divided by location and colored according to precipitation amount. The low-resolution data consists of grids that are 0.25 degrees in resolution and each instance in the file corresponds to a precipitation value for a grid location on a given day. The high-resolution data is about 1/5 the size of the low-resolution, so each grid is only 0.5 degrees in resolution.

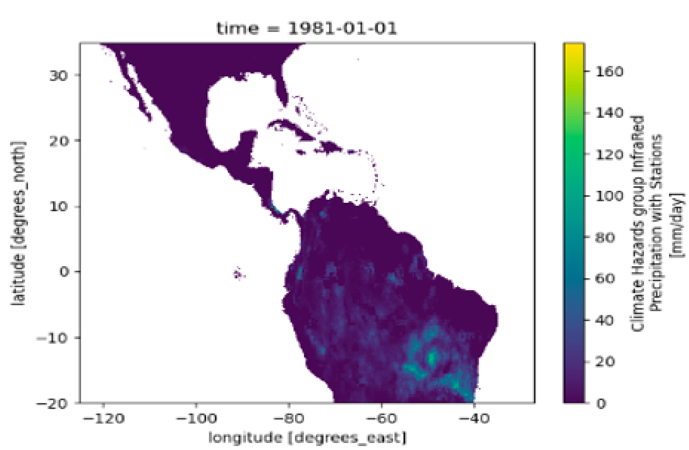

Fig. 1: Coarse-Projected Precipitation Map

As shown in Fig 1, each grid point contains four informatics, spatial which are latitude and longitude coordinate points, temporal is time span and numeric percentage of precipitation rate. The study area covers a fixed region central America consisting of Panama, Costa Rica, Nicaragua, Honduras, El Salvador, Guatemala, and Beliz.

IV. Related Work

State of the art model discuss probabilistic model fitting which involves different types of regression discussed in [4] quantile regression (with respect to a package implementation in R). This method uses quantile regression, which builds on linear regression in a possibly nonlinear setting, estimating the conditional median of the response variable as opposed to the mean as in mean squared error. This allows for more dynamic predictions of values, for example, where a prediction on the 25th quantile means there is a 25% chance the target value is below the prediction. This method is preferable in the downscaling problem field due to the uncertain nature of weather data. The natural extension of quantile regression is applying it within an artificial neural network (ANN). Specific to the 'qrnn' package in R, the method is a multilayer perceptron ANN. This version of the QRNN model is flexible in representing predictor-predictand relationships without explicit specification of the relationships by the modeler. One notable drawback of this model is the fact that optimization methods may converge prematurely, which results in sub-optimal parameters and possible problems with the model. Another important drawback to mention, for our purposes, would be this model does not deal with spatial data at all. It takes in relational data formats, which is not useful in our proposition.

The next method discussed in [5] approached a statistical downscaling problem from the neural network point of view. As the authors state, deep neural networks have shown significant "success in solving some of the most computationally difficult problems" [6]. These machine learning models can learn hierarchical representations and highly complex functions, so they can be effective in the statistical downscaling problem. The specific architecture of this paper's model uses a two-layer deep model with Long Short-Term Memory (LSTM) applied in the second hidden layer. The addition of LSTM allows the model to remember more previous states for each new training example, and thus it is able to learn a more accurate representation of past data. Precipitation downscaling has long-term predictor variables, and the implementation of LSTM in this study showed improvements over previous short-term focused models. In our application, this increased memory component of the model will help with the seasonality aspect of precipitation data. For example, the model will be able to learn that a certain range of days sees more precipitation than others (the rainy season) and apply this to our interpolation.

With large spatial datasets, convolution helps models transform data into a more useful format. Many models can benefit from convolutional layers, as presented in [7] where they apply 3 convolutional layers in a neural network with an additional fully connected layer. The convolution process essentially filters data in sequential blocks and scales down features while maintaining the value to be gained from the data. Put more simply, the model "learns higher-level abstractions from predictors" [7] which can improve performance when compared to other simpler neural networks. We can apply convolution in our interpolation problem to achieve these same benefits, since our data covers a wide spatial area that could have inherent features to be learned. The filtering process used in CNNs applied to an LSTM model may provide the perfect combination of feature manipulation and long-term consideration needed for our precipitation interpolation project.

A final model detailed expands upon the CNN concept with a Super Resolution Deep Residual Network (SRDRN). The main differences are the "unusually deep convolutional layers with batch normalization and residual networks..." [8]. Each residual block consists of convolution followed by batch normalization, and a parametric ReLU activation function. The convolution and normalization are repeated and an elementwise sum is computed, and this process is repeated in successive residual blocks. The model then uses unsampling blocks to downscale coarse resolution feature maps to finer resolution. This model provides a promising alternative to a ConvLSTM architecture, as it performs similar operations. The only drawback upon initial research is the lack of long-term memory, which is an important factor in our time-series data.

V. Proposed Method

A. Technical Background

Super Resolution models are for grid patterned data such as image processing and recognition, they are designed to adaptively learn spatial hierarchies of features from minimal to high-level patterns. SR is made up of three types of layers: convolution, pooling, and fully linked layers. The first two layers, convolution and pooling, extract features, and the third, a fully connected layer, transfers the extracted features into final output, such as forecasting a grid. A convolution layer is essential in CNN, which is made up of a stack of mathematical operations such as a sort of linear operation. Pixel values in grids of nc file are analogous to digital images which are stored in a two-dimensional (2D) grid, an array of numbers e.g. "Precipitation value" and a small grid of parameters e.g. "Latitude and longitude " called kernel, an optimizable feature extractor is then applied at each grid position. Training is the process of improving parameters such as kernels in order to reduce the difference between outputs and ground truth labels using optimization algorithms such as backpropagation, gradient descent or built in Adam optimizer, among others.

It is impractical to connect neurons in the preceding volume to all neurons when dealing with high-dimensional inputs such as images. Instead, each neuron should only be

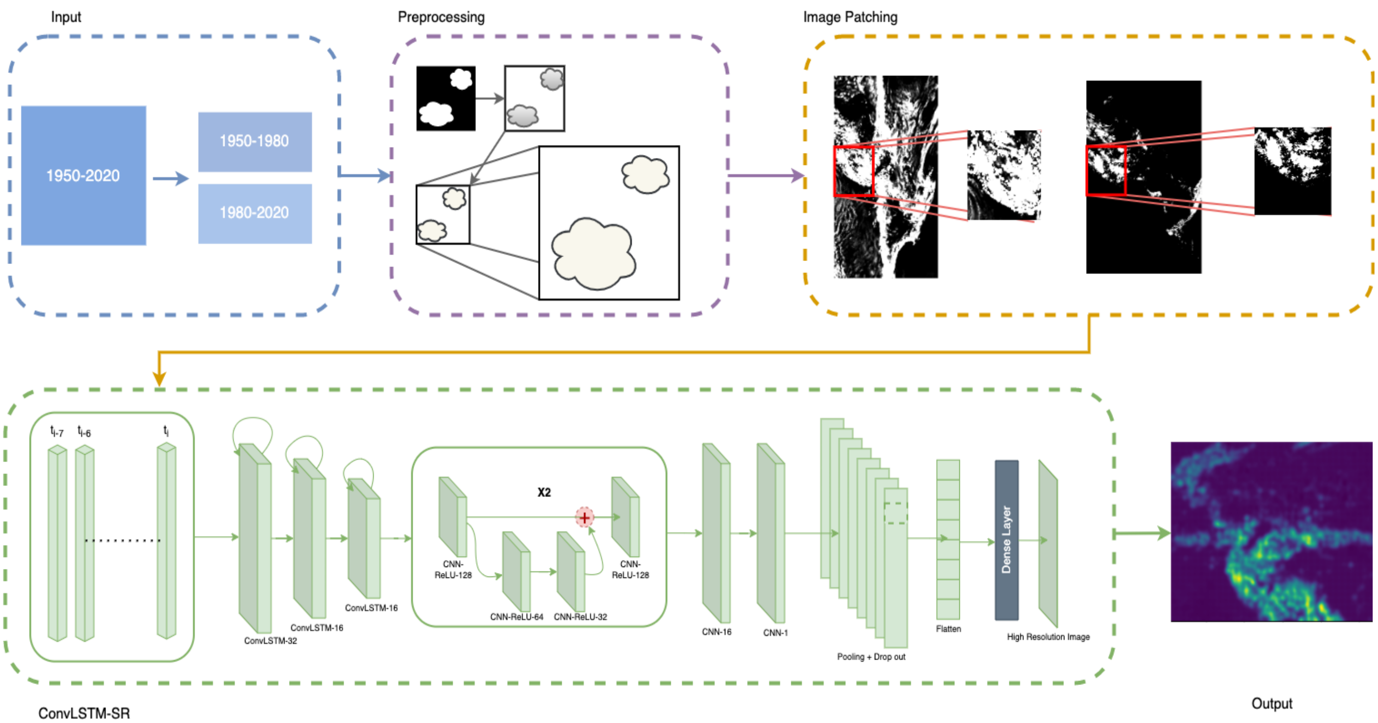

Fig. 2: Overview of our Methodology. The input is divided into two datasets: (1950-1980) data that has both high and low resolution and (1980-2020) that has only low resolution information. The preprocessing involves masking, noise reduction, and upsampling. After extracting patches, our model ConvLSTM-SR is trained and generates the output.

connected to a small portion of the input volume. The spatial extent of this connectivity is a hyperparameter known as the neuron's receptive field (equivalently this is grid size). Grid correlates to the three channels which are latitude, longitude and time. This must be emphasized once more in how We treat the spatial dimensions e.g."Latitude and longitude" and the depth dimension e.g."Time". Difficulties with existing neural networks in capturing spatial and temporal dependencies, and the inability to estimate both averages and extreme precipitation events with a single method motivated us to use a Super-resolution based Convolutional LSTM neural network to solve this problem network to solve this problem e.g.(ConvLSTM-Sr).

We present a model with a single image super-resolution (SR) architecture that has considerable potential in statistical downscaling because both objectives depend on learning a mapping between fine and coarse projected images. In addition, We use a convolutional long short-term memory module to capture long-range relationships and nonlinear dynamics. In this study, enhanced daily predictability and the capacity to estimate extremes with the deep learning-based precipitation downscaling model, the groundwork for future iterations to generate fine-projection climate change estimates from coarse-projection ones accurately and reliably.

B. Model Network Description

Suppose that x_t represents coarse projected map at t day consisting of four input sequence representing one climate variable e.g."precipitation", spatial and temporal dependecies. Each xt represents an "map/image" of size 4 × N × M, where last two dimensions representing spatial dimensions are (1100,1960).

$$x_t = [x_{T-t}, x_{(T-1)-t}, x_t]$$

where the initial parameters of an image taken in model has a predefined values as follows. No.Channel which is the number of predicant variables set to one, dimension projections set to [40,40] which means a compiled total of 40 latitude and longitude points in an "image" corresponding to 1600 individual grid points, times step indicates number of compiled "images' for the telecommunication of temporal dependencies set to $T = 7$.

The Conv Long Short-Term Memory (LSTM) module introduces self-loops for the output state to allow the training process to run for long periods of time without reaching extremely high or low weight gradient values. Furthermore, the weight of the loop allows the state or memory of the unit to be gated, where a separate hidden unit controls whether a new state is updated or not and is conditioned as part of the training process [9]. The Conv Long Short-Term Memory (LSTM) module introduced self-loops for the output state to allow the training process to run for long periods of time without reaching extremely high or low weight gradient values. Furthermore, the weight of the loop allows the state (or memory of the unit) to be gated, where a separate hidden unit controls whether a new state is updated or not and is conditioned as part of the training process [10]. Additional

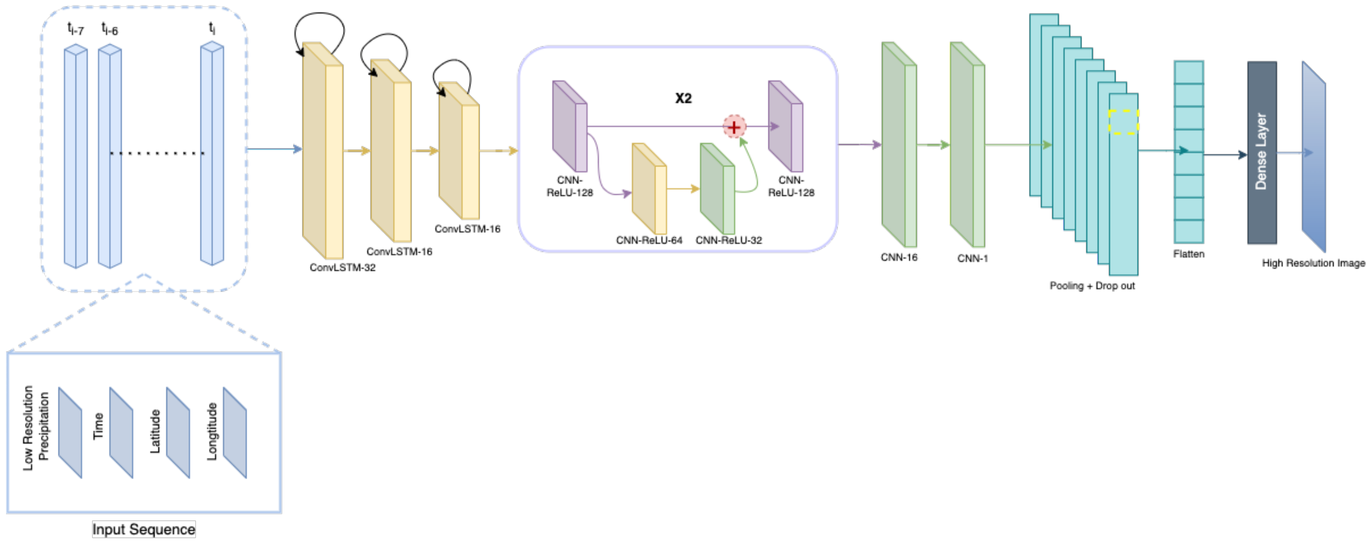

Fig. 3: Proposed Conv-LSTM SR Starting by Channel Inputs Towards Down-scaling a Precipitation Projection Map.

gates are dynamically learned, including an input gate that regulates whether or not to take input and an output gate that allows the module to learn whether or not to emit output [10]. In order to define an on-off signal, each of the gates uses the sigmoid non-linear activation, and the input activation can utilize any of the activation functions [10]. The LSTM can learn very long-term dependencies between inputs thanks to the combination of the self-loop and gate functions, but the state is confined to a single cell [10]. ConvLSTM-SR this model uses up-sampling blocks to downscale coarse resolution feature maps to finer resolution. This model provides a promising alternative to a ConvLSTM architecture, as it performs similar operations. With an advantage over other deep learning techniques long-term dependency by SR blocks, which is an important factor in our time-series data.

As presented in [11] where they apply 3 convolutional layers in a neural network with an additional fully connected layer. The two final CNN convolution process essentially filters data in sequential blocks and scales down features while maintaining the value to be gained from the data.

C. Model Network Heuristics

Our model is as follows a Convolutional LSTM,

$$\text{ConvLSTM}(\delta, k)$$

, where delta and k represent number of filters and kernel dimensions, respectively as per mathematical indication of [12] :

$$F_1(x) = \text{ConvLSTM}(\delta_1, k_1)(x)$$

$$F_2(x) = \text{ConvLSTM}(\delta_2, k_2)F_1(x)$$

$$F_3(x) = \text{ConvLSTM}(\delta_3, k_3)F_2(x)$$

δ_1, δ_2, δ_3 are 32, 16, and 16, respectively. Similarly, we set k_1, k_2 and k_3 to 9, 3, 3. After three stacked ConvLSTM processes, the resultant output tensor is 14×N×M in size. The ConvLSTM's recurrent nature aids in the utilization of data's temporal dependencies. Convolutional actions inside the cell, on the other hand, aid in the preservation of spatial correlation. As a result, the output tensor encodes the data's temporal as well as geographical dependencies. Fig 3's super-resolution (SR) block improves the resolution of high-dimensional feature data obtained from the ConvLSTM layers preceding it. The SR block is made up of two deep Convolutional layers stacked on top of each other with skip connections between them.Relu activation is present in each convolution layer [12]. Filters of sizes 128×5×5, 64×3×3, and 32×3×3 are represented by the parameters.

Convolution is followed by batch normalization and a parametric ReLU activation function in the SR block. Furthermore, after the SR block, the image vector representations are preceded to the CNN-16 and CNN-1 layers, with represented filters of sizes 16×5×5 and 1×3×3. This operation is repeated in subsequent blocks, with the convolution and normalizing repeated and an elementwise sum obtained. Then, using those vector feature representations, we minimize the network's superfluous feature dependencies by a factor of 0.5, making it simpler and improving its generalization abilities. In addition to pooling, which down samples the representation of an input vector.As a result, the model becomes less sensitive to certain translations, resulting in better translation invariance. Flattening the vector representation into a one-dimensional array of precipitation values and densely incorporating it into the spatial dependencies of the input images.

VI. Data Pre-Processing

To be able to train the model and extrapolate missing Fine resolution data, some pre-processing is required as illustrated

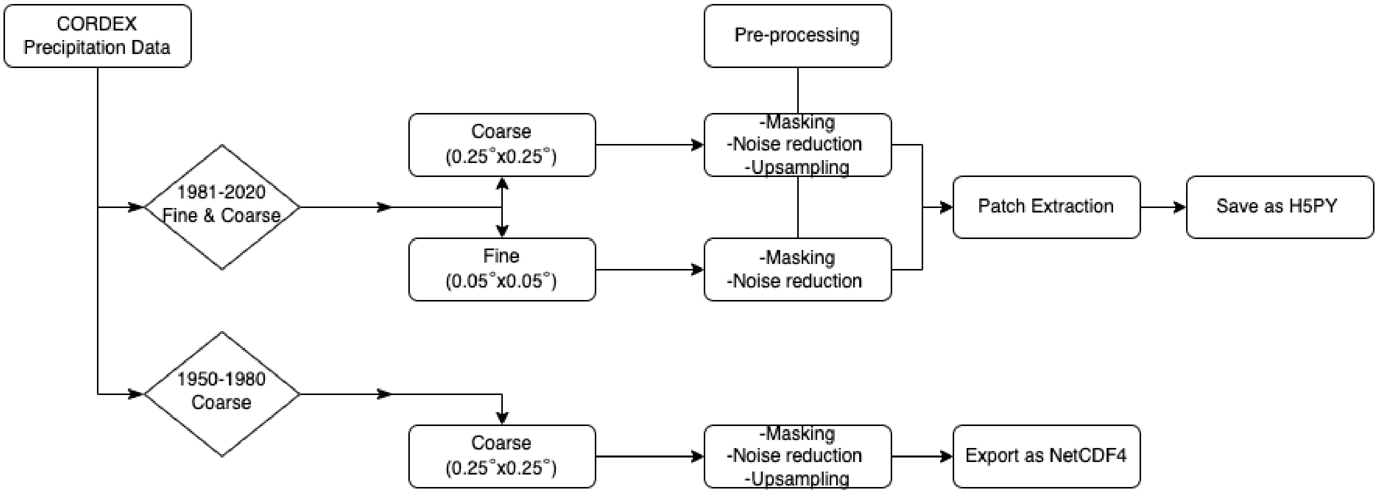

Fig. 4: Web-map of Pre-processing

in Fig 4. Depending on the year index of data framework is divided into two parts to handle fine and coarse projected data . For 1981-2020 data of both projections We mask data in which We replace by zero this is the most arbitrary solution to not increase the variance and add biasness to the data, now since our end goal of preprocessing is to have overlapping patches. We up sample the coarse data to match spatial scale of fine data in which We use cubic interpolation technique from spatial resolution of (241,229) to (1100,1960), since this introduces noise to the values We then perform a scaling by maximum value to reduce noise of precipitation value and apply Gaussian Filter to minimize distortion of images apply same is carried for fine projected data without up sampling.

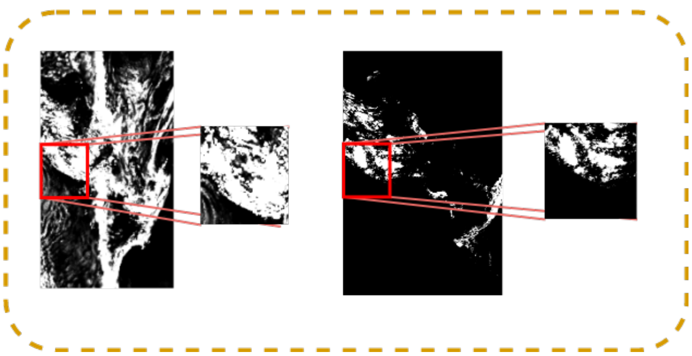

Fig. 5: An example of patch extraction in our data

For patch extraction each image of Central America Map which is represented as an image is divided to sub-image e.g."patches" each patch has 40 longitude and latitude points and total number of sub-images is defined by latitude $= \frac{1100}{40} = 27$ and longitude $= \frac{1960}{40} = 49$ which we set to be a value between (0-27,0-49) first value argument means in a patch how many rows of a single image index to contain and second value is how many index of longitude points, e.g."which means if value takes in 0 one row of latitude points or one column of longitude points and We constrain the total number of patches in a year to be 225*225.

Finally we save as H5PY format for training as this allows me to store pandas data frames in HDF. It also provides a simple way to write, save and extract data with lowest time complexity in which our model can process faster in this format file as shown in Fig 6.

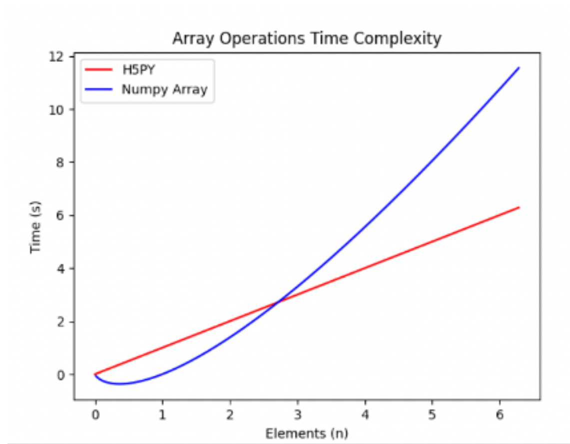

Fig. 6: Array Operations Time Complexity

For the coarse data sets of 1950-1980 similar is done which is masking, Image enhancement e.g."Noise reduction" and up sampling. But at the end We save as NetCDF4 format this is because We will use the trained model to extrapolate 1950-1980 data. Illustrated in Fig 7 the pseudocode of pre-processing this is only for illustrational purposes to get a fundamental understanding of process spoked about in previous paragraphs in more details.

VII. Experimentation

A. Dataset-Splits

A summary of dataset split is shown in Fig 8. For training purpose of our model. We define three sets as train, test and validate. Training set (1992-2020) is a set of used for learning, that is to fit the parameters of the model. Validation set (1988-1991) is a set of used to tune the parameters of a model, for example to choose the number of hidden units in a neural

Algorithm 1: Pre-processing of Data

//Input:
fileLowRes 1981-2020 Low resolution Data as NetCDF4 format files Path
fileHighRes 1981-2020 High resolution Data as NetCDF4 format files Path
filePredict 1950-1980 High resolution Data as NetCDF4 format files Path
ncOutput Extracted patches as arrays output path
//Output:
Extracted patches and saves preprocessed filePredict files

//Description:
This function is the main pre-processing script for ConvLSTM-SR model it consists of getting data masked, Image enhancement e.g.,"Noise reduction" by max value of climate attribute e.g.,"precipitation", Up sampling using Cubic interpolation to match the spatial scale of coordinates in high resolution images, Patch Extraction for Overlapping coarse and fine data and saving coarse data in NetCDF4 format

Initialization
Arrays with precipitation values for 3 file types
Extracting longitude and latitude values and endpoints for different 3 file types
Print Informatics
For loop Cubic interpolation
Get array
Flip
resize by cubic interpolation
Result File

For loop Noise Reduction
Normalize=File/ max precipitation value
Result Normalize

For loop Image enhancement
Gaussian Filter
Result Filtered

For loop Write files
Copy output path from ncOutput
Append File
Resize to float16 type

For loop Extract Patches
Result [0*40:(0*40) + 40, 41*40:(41*40) + 40]
Fine[0*40:(0*40)+40 , 41*40:(41*40)+40]
Result (Normalize, Fine)

Fig. 7: Pseudocode of Pre-Processing Data

network. Test set (1981-1987) is a set used only to assess the performance of a fully-specified model, then we extrapolate model on ”Predict” data set.

B. Settings

The proposed model generates one high resolution image by encompassing seven days low resolution. We incorporate recurrent dropouts of 0.5 to the ConvLSTM layers and between ConvLSTM layers. Additionally, we keep regularization with weight decay of value 0.01 in the ConvLSTM layers. The RMSE loss is optimized using Adam optimizer. Moreover, we keep a learning rate with an initial value of 0.0001 and then it updates by an additive affect of 0.02 . Models were built using Keras-Tensorflow and trained on a CPU of 16 GB RAM.

Overall	Year- range Climate Variables Instances Input shape Output Shape	1950-2020 1 25,550 365 × 229 × 241 365 × 1100 × 1960
Train	Year- range Instances	1992-2020 10220
Validate	Year- range Instances	1988-1991 1095
Test	Year-range Instances	1981-1987 2190
Predict	Year-range Instances	1950-1980 10950

Fig. 8: Experimental Dataset-Splits

C. Comparisons

We compare our proposed model with three recent models and one state-of-the art model, ResLap [13] and [14] DeepSD [15],Quantile Mapping Approach [16], Standard SRCNN [17].

ResLap generates high resolution climate change estimates by layering a Residual Dense Block (RDB) on top of a Laplacian pyramid-based super-resolution network. From the initial precipitation projection, ResLap extracts hierarchical features at various levels. The low-resolution projection is then upscaled to the target scale using a sequence of transposed convolution networks.

DeepSD It employs a super-resolution architecture SRCNN based on stacked convolutional neural networks. DeepSD stacked SRCNN modules are each trained separately to create climate change projections at various resolutions. Low-resolution climate projections are then systematically fed through this stacking pipeline to produce high-resolution climate projections.

Quantile Mapping Approach (QMA). A point-based statistical estimator is followed by a Bias Correction Spatial Disaggregation or Quantile mapping technique. It’s a basic but effective statistical downscaling strategy. These models correct systematic biases in climate variable distribution.

Super Resolution (SRCNN) uses spatial hierarchies of features from minimal to high-level patterns. It consists of three types of layers: convolution, pooling, and fully linked layers. The first two layers, convolution and pooling, extract features, and the third, a fully connected layer, transfers the extracted features into final output.

VIII. Results and Evaluation

A. Objective Loss Metrics

For evaluating our model we'll be using the metrics below suggested by many research papers that was reviewed.

$$\text{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_1 - \hat{y}_1)^2}$$

Above is a measure of how accurate our predictions of precipitation values are, in this equation y and $\hat{y}$ are the truth image and the model output , where n is the fixed number of patches out model is trained on.

$$\text{PSNR} = 10n\log_{10}\left(\frac{1}{\sum_{i=1}^{n}}|I(i) - \hat{I}(i)|^2\right)$$

Above is a measure of how quality of image in enhanced it compares the precipitation values pixel-pixel where each pixel encapsulates total of 27 and 49 longitude and latitude points.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\frac{(\hat{y}_1 - y_1)^2}{n}}{\sum_{i=1}^{n}\frac{(\bar{y}_1 - \hat{y}_1)^2}{n}}$$

Above is a measure of how variability in precipitation values by spatial and temporal attributes.

B. Projection Enhancement

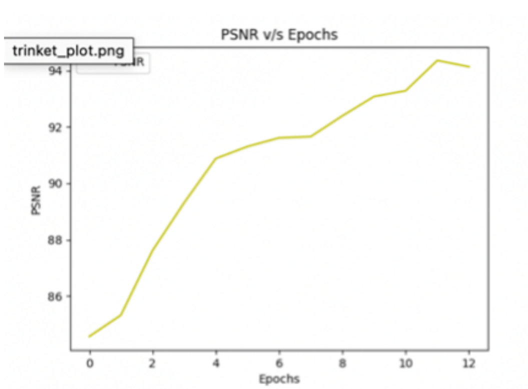

Fig. 9: PSNR Metric While Training for 120 Epochs

We can track the resolution of projection enhancement metric for the trained epochs at about 116 Epoch we eventually achieved a loss of 94.72.

C. Daily Predictability

We can track the predictability of precipitation on daily basis metric for the trained epochs at about 120 Epoch we eventually achieved RMSE of 21.69 and R^2 of 81.06.

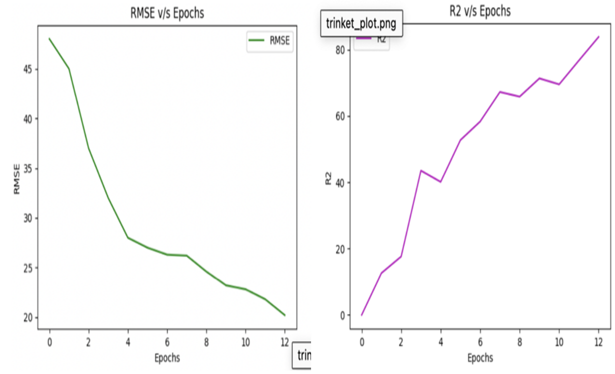

Fig. 10: RMSE and R^2 While Training for 120 Epochs

D. Base-line Comparison

	ConvLSTM-SR	ResLap	DeepSD	QMA	SRCNN
RMSE	21.69	16.02	18.43	26.52	29.83
R^2	81.06	-	-	-	37.66
PSNR	91.121	-	-	-	-

Fig. 11: Comparison of Performance Among Models

Comparing different model approaches as shown in Fig 11. While comparing values between different models performance performed by other researchers note that we confine ourselves to RMSE and R-squared since other researchers didn't provide R^2 or PSNR metric, Between all recent approaches we can see that our model ranks in the 3rd position compared to performing better than QMA and Standard SRCNN as we can see our Deep learning model contains higher hierarchal information both spatially and temporally than SRCNN.

E. Downscaling Result

Shown in Fig 12 is the downscaled result of 1950-01-01, as can be seen going from $0.25\,\text{deg}$ to $0.05\,\text{deg}$ projection of precipitation value diminishes the assigned values of coarse projection precipitation map.

IX. Conclusion and Future Prospects

After numerous trials and errors implementing our model, and having gathered results for evalutation, we see that our

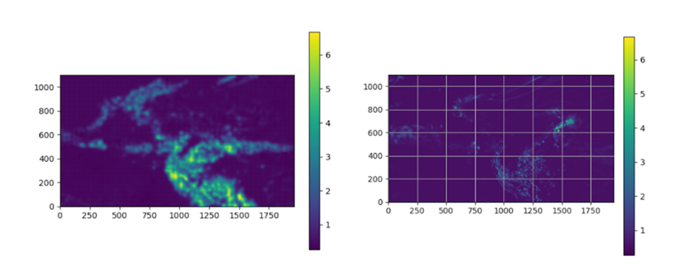

Fig. 12: Coarse and Fine Projected Central America Map

model does outperform other models in some domains. However, we are certain that we have some areas as our potential candidates of improvements. First, we can use a graph neural network to accommodate for non-linear precipitation area coverage. Also, we plan to expand our experiment in different areas by training our model per geographical region partition Central America into Mexico, Panama Guatemala, and etc. We also see that we can have a metriculously designed models with having different models per season (e.g., ”Compile winter for all years and train”). Finally, we see the potential of incorporating Physical constraints based on physical equations of earth system models (e.g., Humidity, Pressure etc.)

ACKNOWLEDGMENT

The authors would like to thank The University of California, Santa Barbara for providing datasets at https://data.chc.ucsb.edu/products/CHIRPS-2.0/.

REFERENCES

[1] Yongkang Xue, Zavisa Janjic, Jimy Dudhia, Ratko Vasic, and Fernando De Sales. 2014. A review on regional dynamical downscaling in intraseasonal to sea- sonal simulation/prediction and major factors that affect downscaling ability. Atmospheric research 147 (2014), 68–85.

[2] Bedia, J., Baño-Medina, J., Legasa, M. N., Iturbide, M., Manzanas, R., Herrera, S., Casanueva, A., San-Martín, D., Cofiño, A. S., and Gutiérrez, J. M.: Statistical downscaling with the downscaleR package (v3.1.0): contribution to the VALUE intercomparison experiment, Geosci. Model Dev., 13, 1711–1735, https://doi.org/10.5194/gmd-13-1711-2020, 2020.

[3] Ahmed, K.F., G. Wang, J. Silander, A.M. Wilson, J.M. Allen, R. Horton, and R. Anyah, 2013: Statistical downscaling and bias correction of climate model outputs for climate change impact assessment in the U.S. northeast. Glob. Planet. Change, 100, 320-332, doi:10.1016/j.gloplacha.2012.11.003.

[4] Cannon, A. J. 2011. Quantile regression neural networks: Implementation in R and application to precipitation downscaling. Computers Geosciences 37(9):1277–1284.https://coek.info/pdf-quantile-regression-neural-networks-implementation-in-r-and-application-to-preci.html

[5] Kingma, D. and Ba, L. 2015. Adam: A Method for Stochastic Optimization. Conference Paper, ICLR 2015. https://arxiv.org/pdf/1412.6980.pdf

[6] Misra, S., Sarkar, S., and Mitra, P. 2017. Statistical downscaling of precipitation using long short-term memory recurrent neural networks. Theor. Appl. Climatol. https://link.springer.com/article/10.1007/s00704-017-2307-2

[7] Sun, L. and Lan, Y. 2020. Statistical downscaling of daily temperature and precipitation over China using deep learning neural network models: Localization and comparison with other methods. International Journal of Climatology 41:1128-1147.

[8] Wu, J., and X. J. Gao, 2013: A gridded daily observation dataset over China region and comparison with the other datasets (in Chinese). Chin. J. Geophys., in press.

[9] Y. Wang, L. Wang, H. Wang, and P. Li, "End-to-End Image Super-Resolution via Deep and Shallow Convolutional Networks," pp. 1–10.

[10] Bashir, S., Wang, Y., Khan, M., Niu, Y. (2016). A comprehensive review of deep learning-based single image super-resolution.

[11] Ghimire, S, Deo, R C, Raj, N and Mi, J (2019). Deep solar radiation forecasting with convolutional neural network and long term short-term memory

[12] S. Kim, S. Hong, M. Joh and S. Song, "DeepRain: ConvLSTM Netwo rk for Precipitation Prediction using Multichannel Radar Data," arXiv arXiv:1711.02316, 2017.

[13] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipi- tation nowcasting," Jun. 13, 2015, arXiv:1506.04214. [Online]. Available: https://arxiv.org/abs/1506.04214

[14] R. L. Wilby and C. W. Dawson, "The Statistical DownScaling Model: insights from one decade of application," in INT J CLIMATOL, vol. 33, no. 7, pp. 1707–1719, 2012.

[15] J. Cheng, Q. Kuang, C. Shen, J. Liu, X. Tan, and W. Liu. 2020. ResLap: Generating High- Resolution Climate Prediction Through Image Super-Resolution. IEEE Access 8 (2020), 39623–39634.

[16] Alex J. Cannon, Stephen R. Sobie, and Trevor Q. Murdock. 2015.Bias correction of GCM precipitation by quantile mapping: how well do methods preserve changes in quantiles and extremes? Journal of Climate 28, 17 (2015), 6938–6959.

[17] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2472–2481.

[18] Thomas , Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. 2017. Deepsd: Generating high resolution climate change projections through single image super-resolution. In Proceedings of the 23rd international conference on knowledge discovery and data mining. 1663–1672.