CLOUD COVER FORECASTING USING LSTM AND GANs

Ninad Daithankar[1], Sanket Tangade[2], Tanay Mayee[3], Saniya Deshpande[4], Dr. Jayashree Prasad[5]

[1,2,3,4,5]*Dept .of Computer Engineering, Sinhgad College of Engineering (SPPU), Pune, Maharashtra, India.*

---***---

Abstract - *Climate change is a severe and a top-priority problem faced by humanity and planet earth. Dramatic and rapid measures are required to avoid increasing climate-related risks for natural and human systems. Governmental interventions, as well as initiatives by NGOs and activists, have not yielded expected published support to fight this critical problem. However, the advent of big data, Machine Learning, and Deep Learning technologies have shown immense potential in solving this problem by contributing with remarkable solutions. One of the challenging problems in climate and weather forecasting is accurate cloud forecast. We present a project that aims to forecast the changes in earth's cloud cover from a satellite point of view taking into account current trajectories of rising temperatures and shifting seasonal patterns. As this is a spatiotemporal sequence prediction problem, we propose a solution combining the use of Generative Adversarial Networks for their generation ability, along with LSTM for their forecasting ability. For training, we use satellite data from the Sentinel 2 of 10 different cloud types with multiple layers annotated at pixel level having 3km per pixel resolution for running our experiments and comparing results with other architectures. We aim to be able to accurately forecast cloud cover on a given place at a specific time. The eventual goal is to enable humans to make more informed choices and produce scientifically credible solutions to the future effects of climate change.*

Key Words: Artificial Intelligence, Deep Learning, Machine Learning, Recurrent Neural Networks, Long Short Term Memory, Convolutional Neural Networks, Generative Adversarial Networks.

1. INTRODUCTION

In today's age, Machine Learning (ML) and Artificial Intelligence (AI) increasingly influences life enabled by significant rises in processor availability, speed, connectivity, and cheap data storage. AI is advancing several aspects of human life and it is hard to picture our future without it. Deep Learning, a branch in AI that imitates the workings of the human brain in processing data and creating patterns for use in decision making has especially yielded great results in finding better solutions for pressing problems to humanity.

Interventions led by the government, NGOs, and activists have failed to garner strong public support. This problem stems from a lack of understanding of the gravity of the situation due to inability in visualizing long-term consequences on earth and our environment. In the past few years AI and ML technologies have shown immense potential in solving problems of climate sciences like weather forecasting, analysis of carbon emissions, and prediction of future forest cover.

Along with these areas, incorrect cloud forecasts are another major problem when dealing with weather forecasts and climate change. Accurate forecasting of cloud formation and development will improve modern weather forecasting systems. AI and Deep learning models due to their high functionality benefits in creating high-resolution visuals of future cloud cover on a given space at a specific time and help people become aware of future consequences of climate change.

1.1 Related Work

Extensive research has been done in cloud cover forecasting but is it very limited concerning exclusively using GANs and LSTM. About our study, the research paper by Alec Radford et al. has been fundamental to understanding the implementation of CNNs (Convolutional Neural Networks) for supervised and unsupervised learning. Through their research, they have introduced a class of CNN called Deep Convolutional Generative Adversarial Networks (DCGANs) and demonstrated its utility for unsupervised learning. Their results have validated that in the generator as well as discriminator from object parts to scenes the DCGANs successfully learns the hierarchy of representation.

To understand existing research in cloud forecasting using satellite-based images, we referred to the research paper by Andreas H. Nielsen et al. Here they have proposed a satellite-based cloud cover dataset "CloudCast". We have used the same dataset as input for our proposed model. The evaluation study done in this research to supplement the dataset has been the pioneering research in this domain. The evaluated prediction methods of long-short term memory, generative adversarial networks, and optional flow-based extrapolation have been foundational for us to build the proposed model in our research with required improvements. The model built in this paper was completely based on the LSTM encoder-decoder model which was somewhat not effective and accurate considering the high image quality of the dataset.

To improve our proposed model for prediction, we have referred to the research study by Waytehad Rose Moskolai et al. In this research study, they have presented an architecture that produces efficient results for deep learning problems. The success of the ConvLSTM model in this research to predict subsequent occurrences of the sequence of images has been most important to improve the efficiency of our model. ConvLSTM which is based on Convolutional Neural Networks is more effective as the convolutional networks reduce the 3D or 2D images to a single 1D vector which is more feasible and easier to train the LSTM network.

1.2 Dataset Description

For evaluation of our problem statement and proposed model, we have used the "CloudCast" dataset. This dataset consists of 70080 images with 11 different cloud types for multiple layers of the atmosphere annotated on a pixel level. It has a spatial resolution of 928 × 1530 pixels (3 × 3 km per pixel) with 15 min intervals between frames for the period January 1, 2017, to December 31, 2018. All the frames are sourced from Europe.

2. METHODOLOGY

2.1 Problem Formulation

The development and formation of clouds are still one of the main unsolved problems, with cloud inaccuracies having far-reaching consequences for the overall accuracy of weather forecasts. Because clouds play such an important role in the Earth's climate system, incorrect cloud forecasts can cause significant uncertainty in overall weather forecast accuracy. Due to a scarcity of high-resolution data, only a few studies could attempt to approach this difficult subject from a Deep Learning (DL) perspective. Consequently, it has been very difficult to measure clouds quantitatively and to evaluate the success of cloud forecasts because of their vertical and horizontal character.

Along with these areas, climate change and weather forecasting are also some other severe and top-priority problems faced by humanity that need to be addressed. Hence, accuracy in forecasting cloud cover plays an essential role in building improved weather forecast systems. The current research in Artificial Intelligence (AI) for using satellite imagery is very limited and our goal is to bridge this gap through our project. AI and ML techniques have been used to contribute to climate sciences, but the considerably new and advanced approach of Deep Learning has not been much applied to these problems.

To address this issue and stimulate more data-driven atmospheric forecasting research, we aim to build efficient, robust, and advanced solutions using Deep Learning methods which will surely overcome the limitation in atmospheric research and forecasting due to the insufficient use of high-resolution imagery for quality prediction. Our goal is to build a high-caliber and effective model to address these gaps.

This project involves building a system that can predict cloud cover on a given region at a specific time. There will be a web-based platform that will facilitate these things. The user interface will be designed as part of the project that would enable the user to input a sequence of images of past cloud cover data. Once the user inputs this data, it will be passed on to an LSTM-GAN model. This model will be trained using supervised learning techniques. Successfully training a GAN is more of an art than a science as there are immense possibilities of errors and failures. Eventually, the model will generate an output of the image sequence predicted as the future cloud cover on the region that was passed as input.

2.2 Data Preprocessing

The dataset which we used for the project is Cloud Cast which is a novel satellite – based large – scale dataset which served as a baseline for forecasting clouds. When we look closely into the structure of the dataset, it contains 70080 cloud cover satellite images which are labeled with 11 different cloud types corresponding to the multiple layers of the atmosphere out of which 1 type depicts that either there a no clouds or that particular data is completely missing.

To make things more relevant and appropriate first of all we had to filter out the data which was completely missing from the dataset. Further as we mentioned that there are 11 different could types including the type with no clouds, we took a novel approach in which we narrowed down those 11 cloud types to just 4 cloud types solely on the basis of the nature of those clouds. Now the reason for this was to facilitate the ease of implementation which ultimately helped us to produce satisfying results by producing realistic looking cloud cover images.

Also, for the purpose of easing out the training process of the model as the LSTM model seems to be quite heavy to train, we normalized each and every image in the dataset in the range of -1 to 1 in order to successfully predict the cloud cover using our sophisticated well thought model.

2.3 LSTM Auto-encoder Model

Initially, we implemented the ConvLSTM model in which we proposed an auto-encoder architecture that consists of an encoder that generates the desired encoded embedding vector further passing it on to the decoder which eventually generates realistic-looking images.

i. Basic understanding of the model and its architecture:
ii. Encoder (encodes the input list)
iii. Encoder embedding vector (the final embedding of the entire input sequence)
iv. Decoder (decodes the embedding vector into the output sequence)

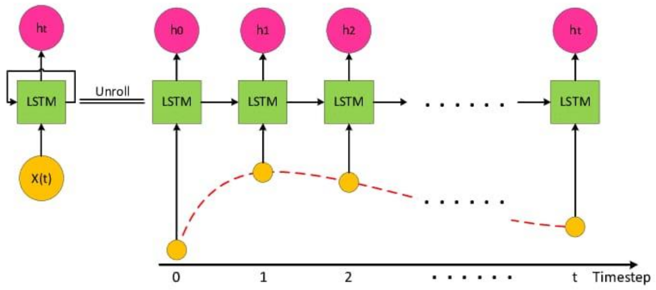

Fig-1: Unrolled LSTM Architecture [*Image downloaded from src: https://www.researchgate.net/figure/Unrolled-LSTM-uses-time-series-data-as-input_fig1_333791434]*

Equations for the regular LSTM cell:

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i\right)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o\right)$$

$$h_t = o_t \circ \tanh(c_t)$$

How does it work?

- Each input is fed into a new encoder LSTM cell and the outputs for all the above equations are calculated.
- The above process is carried out several times and the hidden state which is generated from the final LSTM cell after passing the values through multiple frames consists of the encoded embedding vector which is our final encoder state.
- Here is when the LSTM decoder comes into the picture. It uses the final encoder state (encoded embedding vector) as an input which is then processed iteratively through the decoder LSTM cells to generate a realistic-looking image output.

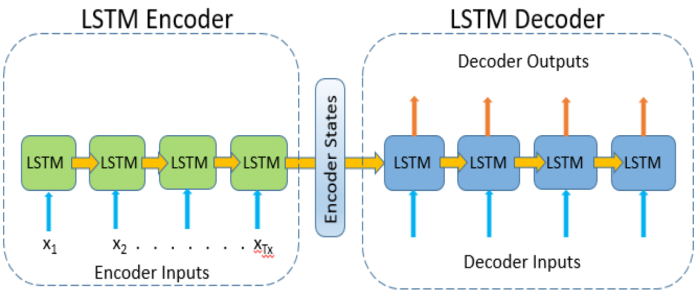

Fig -2: LSTM encoder-decoder architecture
[Image downloaded from src :https://towardsdatascience.com/text-summarization-from-scratch-using-encoder-decoder-network-with-attention-in-keras-5fa80d12710e]

2.4 Problems Encountered

As a result of the previous implementation, after a lot of experimentation, we encountered that the ConvLSTM auto-encoder model's decoder did not produce satisfying results and the model could not produce realistic-looking images with an overall lower training time. As a result, instead of using a typical LSTM decoder, we started experimenting using the Deep Convolutional Generative Adversarial Networks (DC-GANs). After experimenting a bit we came to know that the GAN network is more efficient in generating good quality images as output compared to the LSTM decoder. This motivated us to train the GAN generator for further modelling of the project. To understand it thoroughly let us first look into how the generative adversarial networks work.

2.5 Deep Convolutional Generative Adversarial Networks (DC-GANs)

Deep Convolutional Generative Adversarial Network (DC-GAN) also called Convolutional Neural Network-Generative Adversarial Networks (CNN-GANs) is one of the generative methods based on CNN methods used in deep learning technology. By framing two models, generator and discriminator, GANs are a clever way to train any kind of generative model. Considering the architecture of the GANs model, it is differentiated into two sub-models the generator and the discriminator.

Generator: The generator model takes in random noise inputs in the form of fixed-length vectors as input, processes it, and then ultimately generates a sample realistic-looking image that is expected from the model.

Discriminator*:* The discriminator model is a deciding factor in the model to identify a real or fake image provided as an input. The discriminator takes in real images from the dataset as well as the generated realistic-looking images from the generator and finally classifies them as real or fake. By doing this repeatedly, we can train the generator to produce accurate realistic-looking predicted images.

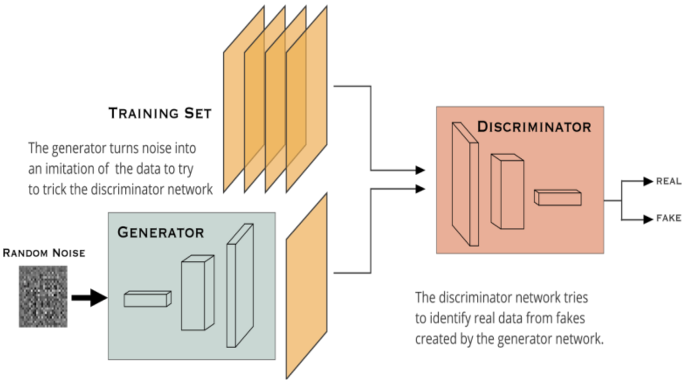

Fig-3: GANs architecture *[Image downloaded from src: https://towardsdatascience.com/image-generation-in-10-minutes-with-generative-adversarial-networks-c2afc56bfa3b]*

After successfully implementing the ConvLSTM auto-encoder, the GAN generator is trained by providing random noise images to generate realistic-looking cloud cover images using the discriminator. As the LSTM decoder was not generating satisfactory good quality output images, GANs being a very effective method to generate high-quality output images, using GANs generator in place of the decoder was a good decision.

The predicted image vector from the ConvLSTM encoder is then passed on to the GAN generator which ultimately generates the desired predicted output image from the single 1D vector. Using the GANs model, not only improved the accuracy of the desired images but also improvised the quality of the output image sequence. This proves that using the LSTM-GAN model proved to be more beneficial and effective in place of the previous LSTM auto-encoder decoder model.

2.6 Model Reconstruction (ConvLSTM Encoder + GANs Generator)

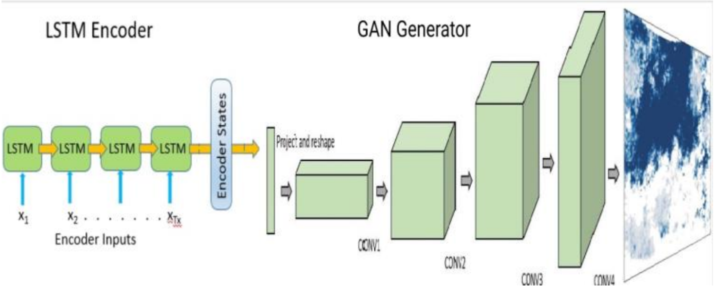

Fig-4: LSTM Encoder - GAN Generator model

After carrying out lots of experiments using GAN and performing multiple iterations, it became pretty clear that the results generated by a GAN are considerably far better than what a typical ConvLSTM decoder would have generated. Considering all the aspects of the final results such as the overall quality, more realistic appearance of the generated images, and a better generation ability of the GAN generator, we followed our instinct and designed a more sophisticated model which in a particular sense produces far better results when compared to a traditional ConvLSTM model.

Initially, the input frames are passed into the ConvLSTM encoder and an embedded tensor is obtained. The tensor is then passed onto the generator, which samples the next 4 frames iteratively as per the input and thus generates predictions. The predictions are then rescaled back with the mean and std deviation used during the normalization phase of preprocessing.

As a result after a lot of analysis and testing, we implement a very divergent model in which we replace the traditional ConvLSTM decoder with a more coherent GAN generator that uses its generative principles to generate more pragmatic results and is far more consistent than a typical ConvLSTM decoder.

3. RESULTS AND DISCUSSION

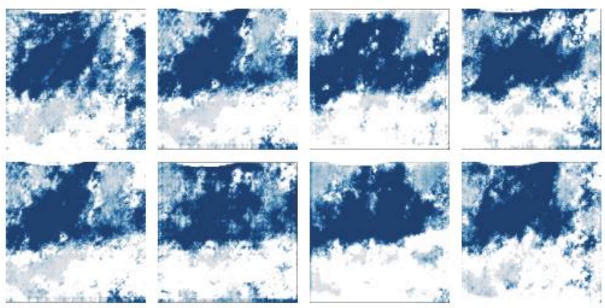

Fig-5: Final GANs Generator Output

As we can see above, the image sequence output generated by the GANs generator using the single 1D vector predicted by the LSTM auto-encoder is the future cloud cover of the earth's surface based on the provided dataset input image sequence. After successfully implementing both the models including the LSTM encoder decoder as well as GANs generative model we can finally observe a good quality cloud cover image has been predicted and obtained by our present LSTM-GAN model overpowering the past LSTM auto-encoder decoder model.

The model's outcome and results can be successfully used by the weather forecast committee for predicting future changes in the cloud cover based on the past or present cloud cover satellite image sequence.

4. CONCLUSIONS

After doing considerable research work in the field of satellite image variations and their prediction, we came to know that there is not much research work done regarding the satellite images and their prediction of the earth surface, climate, cloud cover, etc. This motivated us to work in this field and contribute something in this domain that will indeed help people in this field in the future. Doing a short study in deep learning and machine learning, we came to know that Long Short Term Memory (LSTM), which is a Recurrent Neural Network (RNN) architecture is the most helpful and effective in solving time series data. Also, Generative Adversarial Networks (GANs) is very effective in decreasing the dimension of the output data and generating improved quality desired image outputs.

Using these algorithms we have successfully designed a ConvLSTM – GAN model to predict the future changes in the cloud cover of the earth's surface using satellite image sequences. The ConvLSTM model efficiently scales down the 2D image sequence inputs into a single 1D embedding vector using Convolutional Neural Network concepts. It processes it and ultimately produces a single vector predicted output. This output is then further sent to the GAN generator which is trained using random noise inputs and GAN discriminator. The generator effectively generates the desired image output of the cloud cover.

To sum up, we have successfully achieved the goal of predicting the earth's surface cloud cover changes and to some extent have helped to overcome gaps and insufficient research present in the weather forecast domain. This will help people to get an idea beforehand of the changes in the cloud cover based on the current or previous cloud cover and weather conditions which will indeed help the users to work and take actions accordingly beforehand.

REFERENCES

[1] Andreas H. Nilsen, Alexandros Iosifidis, Henrik Karstoft," CLOUDCAST: A SATELLITE-BASED DATASET AND BASELINE FOR FORECASTING CLOUDS" arXiv preprint arXiv:2007.2018, 2018, July 17, 2020

[2] Alec Radford & Luke Metz, Soumith Chintala "UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS", arXiv preprint arXiv:1511.0634, 7 Jan 2016

[3] Waytehad Rose Moskolaï, Wahabou Abdou, Albert Dipanda, Kolyang "APPLICATION OF lSTM ARCHITECTURE FOR NEXT FRAME FORECASTING IN SENTINEL-1 IAMGES TIME SERIES" Proceedings of CARI 2020 Bruce Watson, Eric Badouel, Oumar Niang, Eds Ecole Polytechnique de Thiès, Sénégal October 2020

[4] Sandra Aigner and Marco Körner. Futuregan: "Anticipating the future frames of video sequences using spatiotemporal 3d convolutions in progressively growing autoencoder gans" arXiv preprint arXiv:1810.01325, 2018

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016. Available: http: //www.deeplearningbook.org. Accessed: 2019-04-11. [Online]

[6] Isabel Urbich, Jörg Bendix, and Richard Müller. A Novel Approach for the Short-Term Forecast of the Effective Cloud Albedo. Remote Sensing, 10(6):955, 2018.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances In Neural Information Processing Systems (NIPS), pages 1097–1105, 2012.

[8] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair,Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets.

[9] Gregor, Karol, Danihelka, Ivo, Graves, Alex, and Wierstra, Daan. Draw: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623, 2015

[10] Mordvintsev, Alexander, Olah, Christopher, and Tyka, Mike. Inceptionism : Goingdeeper into neural networks.http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html. Accessed: 2015-06-17

[11] Zhao, Junbo, Mathieu, Michael, Goroshin, Ross, and Lecun, Yann. Stacked what-where autoencoders. arXiv preprint arXiv:1506.02351, 2015.

[12] S. Hong, S. Kim, M. Joh, and S.-k. Song, "Psique: Next sequence prediction of satellite images using a convolutional sequence-to sequence network," arXiv preprint arXiv:1711.10644, 2017