Time series analysis of satellite data using ConvLSTM for spatio-temporal feature extraction and prediction.

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Chhatra Pratap Bharti

[Roll No: CS-2002]

under the guidance of

Dr. Sarbani Palit.
Associate Professor
Computer Vision and Pattern Recognition Unit

Indian Statistical Institute
Kolkata-700108, India

July 2022

CERTIFICATE

This is to certify that the dissertation entitled **"Time series analysis of satellite data using ConvLSTM for spatio-temporal feature extraction and prediction."** submitted by **Chhatra Pratap Bharti** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dr. Sarbani Palit
Associate Professor,
Computer Vision and Pattern Recognition Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisor, *Dr. Sarbani Palit*, Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions which added an important dimension to my research work.

Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support. I thank all those, whom I have missed out from the above list.

Chhatra Pratap Bharti
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

The rapid changes in climate of a particular place can effect the lives of local peoples and the area on which they are living. If we are able to detect those changes by mapping the spatial and temporal features of the high resolution satellite image and able to predict the changes before, then we can save ourselves from calamities. In this paper we have used two version of ConvLSTM to capture the spatio-temporal features of high resolution multi-spectral time series satellite images(Landsat-8 image data) and predict the next frame. In the first model(basic ConvLSTM) we simply use the ConvLSTM and predict the next image. The second model we have used is ConvLSTM with additional layer of 3D convolution and 3D Trans-convolution with extract more information about temporal and spatial features. The second model is fast in compare to first basic ConvLSTM model. The predicted result are shown in this paper after conducting experiments demonstrate that second model performs better.

Contents

Chapter 1

Introduction

The satellite used for monitoring earth observation on a regular time period by using some sensors are helpful to capture and analysis the changes occurs on earth surface like climatic changes, land changes and water changes etc. One of the earth observation program is Landsat-8 which uses some sensors to captures these changes in form of very high resolution images. The ongoing changes in climatic and anthropogenic global condition of many places causes danger for that areas and it's people. If we are able to map these changes and can extract the useful information from these mapping then we are able to predict the next changes that can occurs and hence we can save ourselves from dangers.

Landsat-8 data is a high resolution multi-spectral images captures by satellite using Operational Land Imager (OLI) sensor and Thermal Infra-Red Scanner (TIRS) sensor in a regular interval. If we are able to extract the spatial and temporal features of these images then we can predict next images. In this paper we only interested in captures the spatio-temporal features that can forecast images only. But further we can do image classification or segmentation on these predicted images to extract more information.

Previously, many models are proposed for extracting the features of sequence of images. Support vector machine(SVM) [17] are used to extract the spatial features but it fails to captures the temporal feature. Optical flows[15] are also used but their good prediction depends on the parameters we are choosing. Similarly 2D CNN are used to extract the spatial features but lacks the temporal features. Many CNN[20] techniques are build to captures the spatio-tempoarl features and they produce the satisfying result also but problems occur with them that they are not suitable for long-range depencies. Therefore sequential models like RNN and LSTM [1, 2, 11, 12, 3, 4, 9, 6] are introduced to solve the vanishing gradient[7] problem which directly solves long-range dependencies[1]. Sequential models are mainly used to extract the temporal based features but they lack spatial features. So learning of both spatio-temporal

features are crucial for better prediction in case of time series high resolution image.

But, in the recent years advancement in machine learning especially in deep learning has been able to solve this problem. If we have a well-defined end to end structure with sufficient data then we can solve this problem by combing the Convolution with LSTM[8, 13, 10]. So we are using ConvLSTM[8] to captures the spatial as well as temporal features of images and predict the next frame. In this paper we are are using two model of ConvLSTM, the first one is basic ConvLSTM we uses multiple layer of ConvLSTM which takes the sequence of images train with end to end layers and predict the output images. In the second the model of ConvLSTM V2[10] we have additional layer of Convolution layer and Trans-convolution along with ConvLSTM. These additional layer extract the spatio-temporal features with more information and it performs experiments faster than basic ConvLSTM.

Chapter 2

Related Work

Recent studies, deep learning algorithms has achieved great advancement in image classification and feature extraction in the field of pattern recognition and computer vision such as such as object detection[18], object tracking[19]. Support vector machine(SVM)[17] are used to extract the spatial feature for image classification. Stacked AutoEncoder (SAE)[14], a classification method has been used for captures spatial-spectral features. Convolution Neural Network (CNN)[18, 19] are used to extract the features of images. Many version of CNN[20] are introduced to extract the features of images for the purpose of segmentation and classification but most of them lacks to captures temporal features.
Therefore to capture temporal features sequence model like RNN (Recurrent Neural Notwork), LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit)[16]. In [6] a sequence to sequence LSTM encoder-decoder are used to train temporally concatenated LSTMs, one for the input sequence and another for the output sequence. A RNN based model[23] is used to predict next video frame where image patches are quantised for the interpolation of intermediate frames but this model predicts only one frame ahead. This work is followed by [9] that predicts next sequence of images using LSTM encoder-decoder predictor model which reconstructs the input sequence and predicts the future sequence but they lack spatial correlation.
In this paper we have used convolutional LSTM (ConvLSTM)[8] which is a sequence to sequence learning framework proposed in[6]. ConvLSTM used convolutional structures in both the input-to-state and state-to-state transitions. We can build a end to end trainable model for prediction by encoding multiple layer of ConvLSTm layer. The second model used in paper is extended version of ConvLSTM[10] with additional layer Convolution and Trans-Convolution that captures anomaly detection as a spatio-temporal sequence outlier which is used for detecting anomalies in videos.

Chapter 3

Preliminaries

In this section we will go through the basic concept that will require to understand idea behind of our model working. We understand that CNN are used to extract spatial features, LSTM are special RNN model used for sequential and time series analysis learning. Therefore LSTM capture temporal features and finally ConvLSTM[8, 10, 13] try to extract spatiotemporal features.

3.1 Convolution as spatial feature extractor

Convolution is a technique which performs affine transformation on volumes to extract the important features and map those features to produce output volumes. Convolution in case of Convolution Neural Network(CNN) is use to captures the important features of input images input propagates toward the deeper layers. This means convolution allow us to encode spatial features of image preserves the relationship between pixels by affine transformation to learn spatial image features.

Convolution operation consists of convolution layer and pooling layer, in both these operation we perform he affine transformation. Convolution layer uses a filter or kernel which performs the dot product between image and filter to extract features result into another image volume. Suppose we have image of size (m x n x n_c) where m is height, n is width and n_c depth in image and a filter F of size (f x f x n_c) then output produce by this convolution operation is (m-f+1 x n-f+1 x n_c). If include the padding and stride of size p and q respectively then output will be of size [($\lfloor\frac{m+2p-f}{s}\rfloor + 1$) x ($\lfloor\frac{n+2p-f}{s}\rfloor + 1$) x n_c]. Pooling is used to reduce the size of representation produced by convolution layer to speed up the computation as well as it detect features more robust while preserving the important spatial features.

Transposed convolution also known as up-convolution which is opposite of normal

convolution i.e., from something that has the shape of the output of some convolution to something that has the shape of its input while maintaining a connectivity pattern that is compatible with said convolution. Therefore it act as a convolution decoder to produce the image.

The problem with CNN is that it only able to captures the spatial features of images, for temporal features we need some time series based model. Therefore we need to use a sequential model also.

3.2 Long Short-Term Memory(LSTM)

Long Short-Term Memory (LSTM) is variant of Recurrent Neural Network (RNN) model which has special variable called memory cell c_t, this memory cell contain different gates to control over the flow of information which traps the gradient, hence it solves a major problem of RNN called the vanishing gradient. Therefore LSTM model can handle the long-range dependencies.

The memory cell c_t collects the information from the previous cell c_{t-1} by activating different gates. If the input gate i_t is activates then it states that a new information is accumulated to the cell. If the forget gate f_t is activates then it means that previous cell state information can forgotten. If output gate o_t is activated then the latest cell output c_t will be propagated to the final state h_t. The key equations of LSTM are shown in (3.1) below, where '$\circ$' denotes the Hadamard product:

$$\begin{aligned} \bar{c_t} &= \tanh(W_{cx}x_t + W_{ca}a_{t-1} + b_c) \\ i_t &= \sigma(W_{ix}x_t + W_{ia}a_{t-1} + W_{ic} \circ c_{t-1} + b_i) \\ f_t &= \sigma(W_{fx}x_t + W_{fa}a_{t-1} + W_{fc} \circ c_{t-1} + b_f) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \bar{c_t} \\ o_t &= \sigma(W_{ox}x_t + W_{oa}a_{t-1} + W_{oc} \circ c_t + b_o) \\ a_t &= o_t \circ \tanh(c_t) \end{aligned} \tag{3.1}$$

where $_1, x_2, ..., x_t$ are inputs, $a_1, a_2, ..., a_t$ are hidden states, $c_1, c_2, ..., c_t$ are output cell, W_{pq} represent weight parameters of q along with it's gate p and b_k represent bias. We can stack multiple LSTM cell to extract temporal features from a complex structure and make future prediction.

3.3 ConvLSTM

ConvLSTM[8, 13, 10] is composite model that extract the spatial and temporal features of the spatio-temporal data simultaneously by replacing the normal matrix

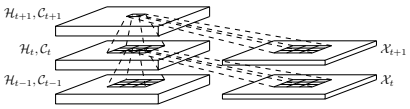

Figure 3.1: Inner Visualization of ConvLSTM structure [8].

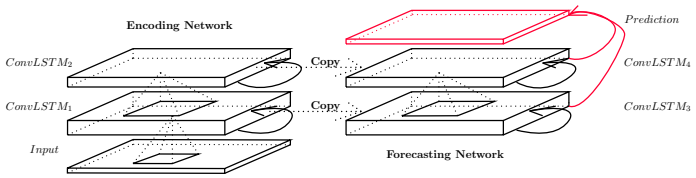

Figure 3.2: Encoding-Images to ConvLSTM network for Frame Prediction [8].

operation to convolution matrix for input to hidden and hidden to hidden transition. This allow ConvLSTM to propagate spatial characteristics temporally through each ConvLSTM state.

Suppose we have input $\mathcal{X}_1, \ldots, \mathcal{X}_t$, hidden states $\mathcal{H}_1, \ldots, \mathcal{H}_t$, cell output $\mathcal{C}_1, \ldots, \mathcal{C}_t$, input gate i_t, forget gate f_t and output gate o_t all are 3-D tensors. The inputs and past states of its local neighbors determines the future state of certain cell using a convolution operator in the state-to-state and input-to-state transitions (see Fig. 3.1). The mathematical representation of ConvLSTM is shown equation in (3.2) below:

$$\begin{aligned}
\bar{\mathcal{C}}_t &= \tanh(W_{cx} * \mathcal{X}_t + W_{ch} * \mathcal{H}_t + b_c) \\
i_t &= \sigma(W_{ix} * \mathcal{X}_t + W_{ih} * \mathcal{H}_t + W_{ic} \circ \mathcal{C}_t + b_i) \\
f_t &= \sigma(W_{fx} * \mathcal{X}_t + W_{fh} * \mathcal{H}_t + W_{fc} \circ \mathcal{C}_t + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \bar{\mathcal{C}}_t \\
o_t &= \sigma(W_{ox} * \mathcal{X}_t + W_{oh} * \mathcal{H}_t + W_{oc} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned} \tag{3.2}$$

where '$*$' represent convolution operator and '$\circ$' represent Hadamard product. All the operation of convolution like padding, stride and pooling are also applicable here.

The ConvLSTM model has two network branch an encoding branch and forecasting branch as shown in Fig 3.2 where the last cell output and it's state of encoding branch is copied to the initial state and cell output of forecasting branch. The encoding LSTM compresses the whole input sequence into a hidden state tensor and the forecasting LSTM unfolds this hidden state to give the final prediction. Multiple ConvLSTM cell are used to extract spatio-temporal features from a complex structure and can be used for prediction.

Chapter 4

Data

4.1 Introduction

We have used here two data-set one is MNIST data-set of handwritten digits and another one is Landsat-8 dataset which is multi-spectral spatiotemporal data. Landsat-8 data contains total 11 bands of images and image bands starts from 1 to 9 are comes from Operational Land Imager (OLI) sensor and 10 to 11 are comes from Thermal Infra-Red Scanner (TIRS) sensor. Each bands represent some some band. For e.g. band-2 corresponds to Blue channel, band-3 corresponds to Green and band-8 corresponds to PAN etc. All these bands are very high resolution images. Landsat-8 data are used to captures the land cover maps information which monitors the ongoing climate changes on those land cover maps. These climate changes includes the surface temperature characteristics, heat, moisture's, rainfall, elevation of grounds and clouds etc. Landsat-8 data can also be used for classification or segmentation of land map region into different sections of bodies like agriculture land, forests, river, ponds, mountains, glacier's and oceans etc.

4.2 Data Preparation

We have downloaded the MNIST data-set from internet which contains total 600 hand-written images each of 28 x 28 resolution. The Landsat-8 data is downloaded from `https://earthexplorer.usgs.gov/` of the region Himachal Pradesh whose coordinates are (Lat: $32^\circ\ 43'\ 54''$ North, Long: $076^\circ\ 33'\ 49''$), (Lat: $33^\circ\ 03'\ 00''$ North, Long: $074^\circ\ 18'\ 21''$), (Lat: $34^\circ\ 26'\ 51''$ North, Long: $074^\circ\ 02'\ 32''$) and (Lat: $34^\circ\ 27'\ 40''$ North, Long: $076^\circ\ 03'\ 30''$).
Each 11 bands of this region has resolution about 7500 x 7700 which is taken over the period of Jan-2021 to Dec-2021. In this duration there are total 17 time-stamp each having 11 bands makes total $17 * 11 = 187$ images are there.

4.3 Data Preprocessing

Images of Landsat-8 data having resolution about 7500 x 7700 are resized to 256 x 256 resolution and 512 x 512 resolution to make two different data-sets for different purposes to train and validate the model. We will see the reason behind to make two different data-sets of Landsat-8 images of different resolution in result section. The MNIST images are resized to 256 x 256 resolution only. Every images in each data-sets are normalize to 0-1 gray-scale image by subtracting each images from global min value and then divided by global max value to make it on same scale.

First we have taken 4 consecutive frames as input to make a single window and the next single frame is for comparison as a output frame then we slide the window by factor of one to select next four consecutive frame and its next single frame again for comparison and continue this process until we cover all images. So here a single input example consists of 4 continuous frame and a single output example consists of one frame only. This output example is act as a ground truth when compared to predicted output.

Similarly we repeated the same process for 6 consecutive frames as a input and next single image a output. So here the single input example consists of 6 frames and the output examples consists of single frame which also act as a ground truth when compared to predicted output.

In one case we have used composite data (MNIST+ Landsat-8) i.e. 600 MNIST images are used to make training examples by above process and 17 images of Band-1 over that time period are used to make validation example.

In another case we have used purely Landsat-8 data for training and validation.Validation are done on 17 images of Band-1 and the training is done on rest images of band 2 to 11 that means training images are 10 x 17 = 170.

4.4 Data Visualisation:

Let us have look on the input examples which can be passed to our model, we have two types of input examples one have window size of 4 and another input example have window size of 6 and the output is single image.

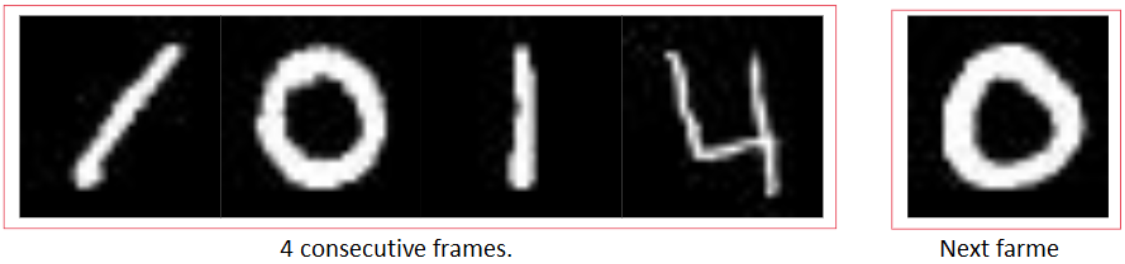

Figure 4.1: MNIST input and output examples when window size for input is 4.

Figure 4.2: MNIST input and output examples when window size for input is 6.

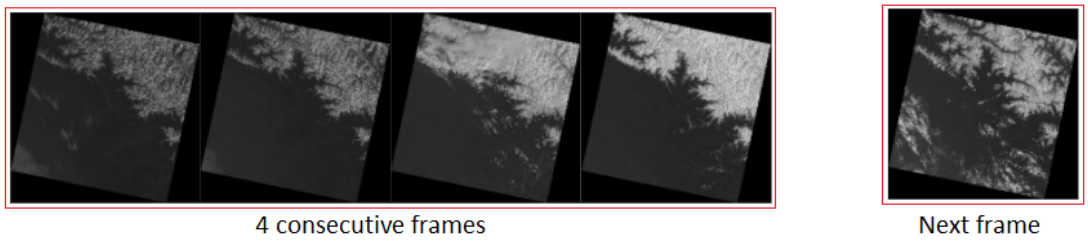

Figure 4.3: Landsat-8 input and output examples when window size for input is 4.

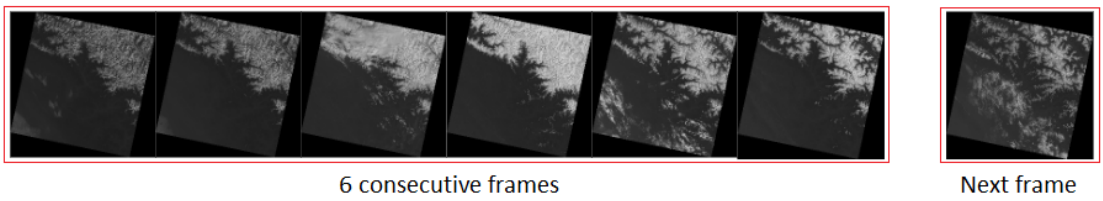

Figure 4.4: Landsat-8 input and output examples when window size for input is 6.

Chapter 5

Prediction using basic ConvLSTM Model [8].

5.1 Motivation

If we are able to capture the ongoing change in the climate conditions and able to map those changes to get some useful information, based on the previous time series data(satellite land maps of different bands) of those climatic changes then we can predict next instance of those changes. Thus we can prepare for calamities that can caused by these rapid changes of climate.

ConvLSTM is used to extracts the spatio-temporal features of time series images and with this property of ConvLSTM we can use it as a prediction model to predict the future frame. Therefore ConvLSTM are better model to get information from time series images and it can be used as a predictor model to forecast next frame.

5.2 Preliminaries

5.2.1 Architecture

The architecture shown in Fig 5.1 takes a sequence of consecutive frames of size N, passes these sequence of images to first layer of ConvLSTM where first layer create the tensor of these sequences and performs the action mentioned in the section 3.3 using some kernel size (5x5) and number of filters (64) to produce the output which passes through next layer and again performs same action with same or different kernel size (3x3) and number of filter (32) and then to next layer with kernel size (3x3) and number of filters (16) then we apply convolution to reconstruct the predicted image using (3x3) kernel size, 3 filters, stride = N. In all the layers we need to check the

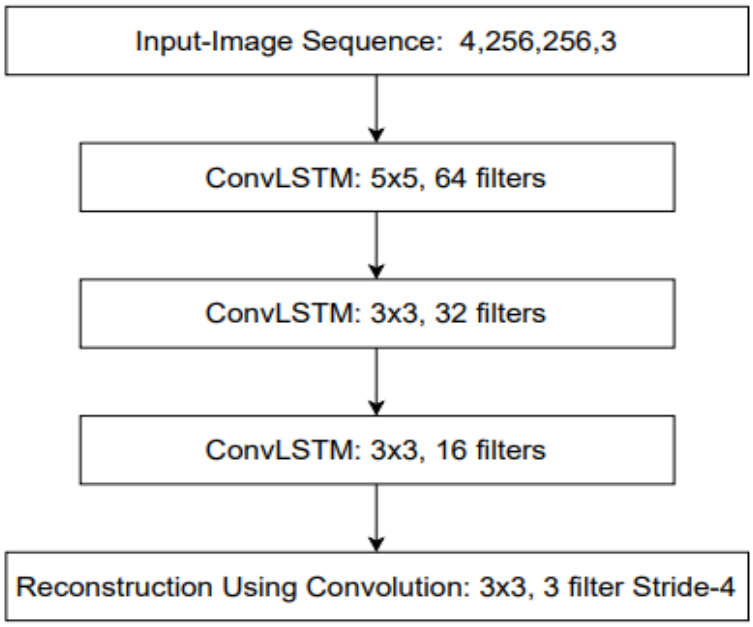

Figure 5.1: Our architecture model. It takes N consecutive frames of images and passed this inputs examples to this model and produce the predicted image of the sequence input.

dimensions of inputs and outputs produce by each layers with proper kernel size, number of filters and strides.

5.2.2 Implementations

Input layer consists of 4 consecutive images each have dimension 256x256x3 is passed through first layer where spatial features are extracted through convolution with filter size 5x5 and number of filters 64 to produce the output as sequence of 4 images of dimension 256x256x64 and temporal features are also extracted in this process using LSTM also, now we perform batch normalization and passes these output to next ConvLSTM layer which again extract spatio-tempral features using 3x3 kernel size, 32 filters produce output as (4,256,256,32) and performs batch normalization and these output passed to next layer which uses 3x3 kernel size, 16 filters produce output as (4,256,256,16). Finally we will apply convolution to result produced by last layer using 3x3 filter size, 3 filters and stride = 4 to get a single forecasting image of dimension 256x256x3 as shown in Fig 5.2. We will compare this predicted image with the 5^{th} image of the sequence with act as ground truth here.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 4, 256, 256, 3)]  0
_________________________________________________________________
conv_lst_m2d (ConvLSTM2D)    (None, 4, 256, 256, 64)   429056
_________________________________________________________________
batch_normalization (BatchNo (None, 4, 256, 256, 64)   256
_________________________________________________________________
conv_lst_m2d_1 (ConvLSTM2D)  (None, 4, 256, 256, 32)   110720
_________________________________________________________________
batch_normalization_1 (Batch (None, 4, 256, 256, 32)   128
_________________________________________________________________
conv_lst_m2d_2 (ConvLSTM2D)  (None, 4, 256, 256, 16)   27712
_________________________________________________________________
conv3d (Conv3D)              (None, 1, 256, 256, 3)    1299
=================================================================
Total params: 569,171
Trainable params: 568,979
Non-trainable params: 192
_________________________________________________________________
```

Figure 5.2: This is the snapshot of model summary where input layer has 4 consecutive frames each of dimension (256x256x3) and passes through subsequent layers and produces a single predicted future frame.

5.2.3 Result

The first row of Table 5.1 represent that the 6 consecutive images of resolution 256x256x3 makes one input example, we have taken purely Landsat-8 data for training and validation the model, having 163 training examples and 9 validation examples passed through a 3 layer basic ConvLSTM model having batch size 4 with 100 epochs gives training error 0.0107 and validation error 0.0132. The result produced by this combination is shown in Fig 5.3.

Similarly for second row of Table 5.1 has window size of 4 sequential images of purely Landsat-8 data having 164 training examples and 11 validation examples with 2 layers of basic ConvLSTM gives training error 0.0096 and validation error 0.0119. The result produced by this combination is shown in Fig 5.4.

Finally the last row of Table 5.1 has window size of 4 sequential images of mixdata(MNIST+Landsat-8) where training is done on 595 MNIST examples and validation is done on 11 Landsat-8 examples with 3 layers of basic ConvLSTM model gives high error. The output produce here is very bad.

Input	Data Type	Number of Training Examples	Number of Validation Examples	Number of Layers	Batch Size	Number of Epochs	Training Error Loss	Val_loss
6,256,256,3	Landsat-8	163	9	3	4	100	0.0107	0.0132
4,256,256,3	Landsat-8	164	11	2	4	100	0.0096	0.0119
4,256,256,3	Mix-data Train_MNIST Val_Landsat	595	11	3	4	50	0.0571	0.0239

Table 5.1: This table shows the different hyper-parameters combinations and their training loss and validation loss.

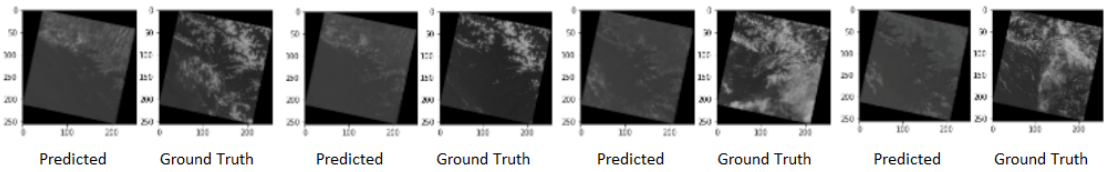

Figure 5.3: This is the result produced by 3-layer basic ConvLSTM model when input is 6 consecutive frames.

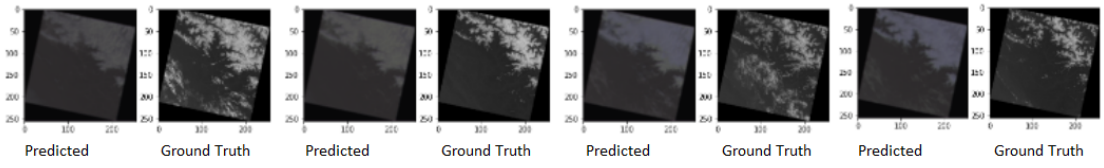

Figure 5.4: This is the result produced by 2-layer basic ConvLSTM model when input is 4 consecutive frames.

Chapter 6

Prediction using ConvLSTM V2 [10].

6.1 Motivation

The basic idea to use this model is that it produce the better result with fast computation. In this model we add more layer of convolution and up-convolution which works as encoder and decoder, hence captures the spatial features with more information and thus predicts the images with better results.

6.2 Preliminaries

6.2.1 Architecture

The architecture shown in Fig 6.1 takes a sequence of consecutive frames of size N, we passed this sequence into two separate convolution layer followed by one another which performs convolution operation explained in section 3.1. These convolution layer act as spatial encoder. Now these N sequence of after convolution has reduced dimension will pass through the 3 layers of ConvLSTM which extract the spatio-temporal features. Since the input given to ConvLSTM has lower dimension which makes this model to train and validate faster than the basic ConvLSTM model(5). The result produced by 3 layers of ConvLSTM is passed into two layers TransConvolution to get back the results which has same dimension as input but with better saptio-temporal features. These TransConvolution layer act as spatial decoder. Now the output of TransConvolution is passed through a convolution layer to predict a single image. These extra layers provides better feature extraction.

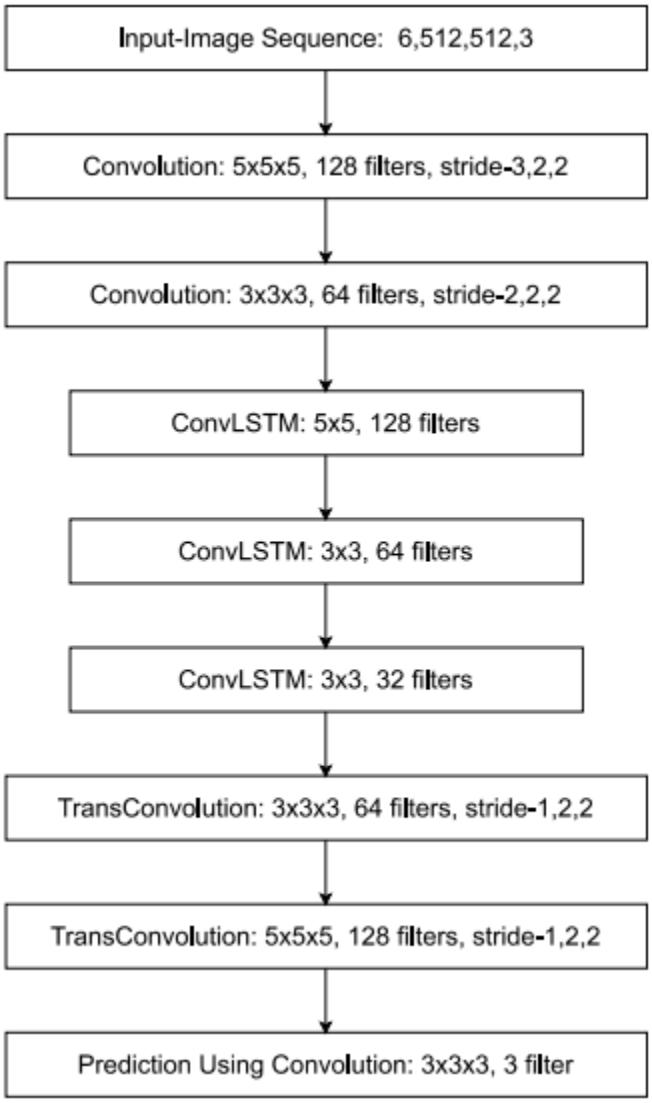

Figure 6.1: This architecture of ConvLSTM V2 have added two layers of convolution and two layers of up-convolution which takes input as a sequence of images and passed through subsequent layer to produce single predicted image.

6.2.2 Implementations

The Fig shown in 6.2 takes input examples of 6 consecutive images each having dimension 512x512x3 are passed through first layer of convolution using filters=128, kernel size=(5, 5, 5), strides = (3, 2, 2) to produce output shape(2, 256, 256, 128) along with max pooling, this output passed down to second layer of convolution using filters=64, kernel size=(3, 3, 3), strides = (2, 2, 2) with pooling layer to produce output result (1, 128, 128, 64). This output has reduced dimension which passed through first layer of ConvLSTM with parameters filters=128, kernel size=(5, 5) to produce output shape (1, 128, 128, 128) followed by batch normalization, these first layer ouput are passed through second layer of ConvLSTM with same process as ConvLSTM layer 1 has, but having different prameters [filters=64, kernel size=(3, 3)], the output produced by ConvLSTM layer 2 are passed down to next layer of ConvLSTM using filters=32, kernel size=(3, 3) produce output shape(1, 128, 128, 32). Now this result are passed to next layer of Transpose Convolution layer using filters=64, kernel size=(3, 3, 3), strides = (1, 2, 2) to produce output shape (1, 256, 256, 64), this output passed to next Transpose Convolution layer using filters=128, kernel size=(3, 3, 3), strides = (1, 2, 2) to produce output shape (1, 512, 512, 128) . Finally this result are passed through convolution layer with 3 filters and kernel = (3, 3, 3) to produce as single predicted image.

These input shape can be changed with different window size and dimensions and we can change the hyperparameter like number of layers, filter size, number of kernel, strides with suitable values. We will se the different results in section 6.2.3.

6.2.3 Result

The first row of Table 6.1 represent that the 6 consecutive images of resolution 512x512x3 makes one input example, we have taken purely Landsat-8 data for training and validation the model, having 163 training examples and 9 validation examples passed through the architecture mentioned in 6.1 with 6 batch size, 200 epochs produces result shown in Fig6.3 having 0.0112 training loss and 0.0102 validation loss.
For second row of Table 6.1 we have changed only batch size = 4 and epochs = 100 and rest things remain same as first row of Table 6.1 gives result shown in Fig6.4 with 0.0107 training loss and 0.0100 validation loss.
In the third row of Table 6.1, we have changed number of ConvLSTM layers (2) with 6 batch size and 100 epochs gives result as shown in Fig6.5 having 0.0134 training loss and 0.0109 validation loss.
Finally for the fourth row of Table 6.1 we have used mix-data which produces the worst result with 0.0518 training loss and 0.0191 validation loss.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 6, 512, 512, 3)]  0
_________________________________________________________________
conv3d (Conv3D)              (None, 2, 256, 256, 128)  48128
_________________________________________________________________
max_pooling3d (MaxPooling3D) (None, 2, 256, 256, 128)  0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 1, 128, 128, 64)   221248
_________________________________________________________________
max_pooling3d_1 (MaxPooling3 (None, 1, 128, 128, 64)   0
_________________________________________________________________
conv_lst_m2d (ConvLSTM2D)    (None, 1, 128, 128, 128)  2458112
_________________________________________________________________
batch_normalization (BatchNo (None, 1, 128, 128, 128)  512
_________________________________________________________________
conv_lst_m2d_1 (ConvLSTM2D)  (None, 1, 128, 128, 64)   442624
_________________________________________________________________
batch_normalization_1 (Batch (None, 1, 128, 128, 64)   256
_________________________________________________________________
conv_lst_m2d_2 (ConvLSTM2D)  (None, 1, 128, 128, 32)   110720
_________________________________________________________________
conv3d_transpose (Conv3DTran (None, 1, 256, 256, 64)   55360
_________________________________________________________________
conv3d_transpose_1 (Conv3DTr (None, 1, 512, 512, 128)  1024128
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 1, 512, 512, 3)    10371
=================================================================
Total params: 4,371,459
Trainable params: 4,371,075
Non-trainable params: 384
_________________________________________________________________
```

Figure 6.2: This is the snapshot of model summary where input layer has 6 consecutive frames each of dimension (512x512x3) and passes through subsequent layers and produces a single predicted future frame.

Input	Data Type	Number of Training Examples	Number of Validation Examples	Number of Layers	Batch Size	Number of Epochs	Training Error Loss	Val_loss
6,512,512,3	Landsat-8	163	9	2, **3**, 2	6	200	0.0112	0.0102
6,512,512,3	Landsat-8	163	9	2, **3**, 2	4	100	0.0107	0.0100
6,512,512,3	Landsat-8	163	9	2, **2**, 2	6	100	0.0134	0.0109
6,256,256,3	Mix-data Train_MNIST Val_Landsat	595	9	2, **3**, 2	4	50	0.0518	0.0191

Table 6.1: This table shows the different hyper-parameters combinations and their training loss and validation loss for ConvLSTM V2.

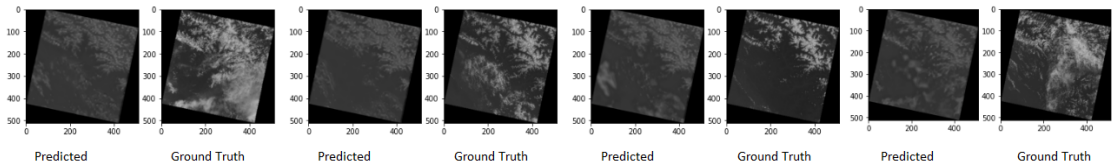

Figure 6.3: This is the result produced by 3-layer ConvLSTM V2 model when input is 6 consecutive frames with the parameters mentioned in the first row of the table 6.1.

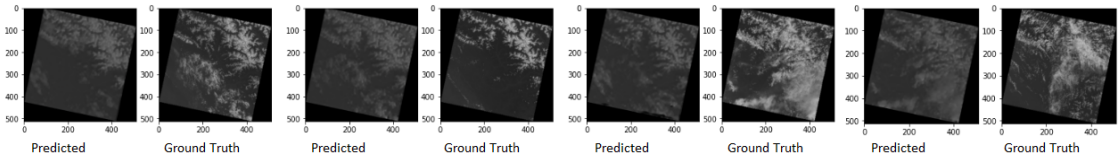

Figure 6.4: This is the result produced by 3-layer ConvLSTM V2 model when input is 6 consecutive frames with the parameters mentioned in the second row of the table 6.1.

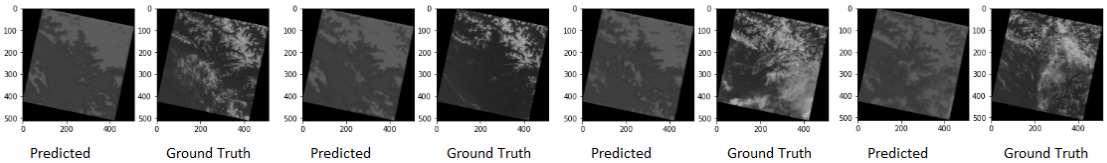

Figure 6.5: This is the result produced by 3-layer ConvLSTM V2 model when input is 6 consecutive frames with the parameters mentioned in the third row of the table 6.1.

Chapter 7

Future Work and Conclusion

7.1 Conclusion

In our paper we are using two deep learning model to extract the spectral and temporal features of high resolution multi-spectral time series images of Landsat-8 data and predicting the next image. ConvLSTM model are well know that captures the spatio-temporal features, because ConvLSTM is composite model which uses both convolution and LSTM simultaneously. The first model is simple basic ConvLSTM and second model ConvLSTM V2 have additional layer of convolution and TransConvolution. On performing experiments we found that ConvLSTM V2 perfroms faster and produces better predicetd image than basic ConvLSTM

7.2 Future Work

For future work, we can extend this ConvLSTM model with some deep learning model to classify or segment those predicted image to see the changes in the ground truth and predicted images for comparison. If have enough data for each bands of Landsat-8 data then we predict name frame of each band and combine them using QGIS software and preform some actions.

Bibliography

[1] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[2] A. Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.

[3] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In CVPR, 2015.

[4] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604, 2014.

[5] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In ICML, 2015.

[6] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In NIPS, pages 3104–3112, 2014.

[7] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063, 2012

[?] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural network for sequence learning," Computer Science, 2015.

[8] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in Proc. Adv. Neural Inf. Process. Syst., 2015.

[9] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In ICML, 2015.

[10] Yong Shean Chong, Yong Haur Tay "Abnormal Event Detection in Videos using Spatiotemporal Autoencoder", DBLP:journals/corr/ChongT17 - 2017

[11] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In CVPR, 2015.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, pages 1724– 1734, 2014.

[13] Jefferson Ryan Medel, Andreas Savakis "Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks", DBLP:journals/corr/MedelS16 - 2016

[14] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learningbased classification of hyperspectral data," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 6, pp. 2094-2107, Jun. 2014.

[15] W.C. Woo and W.K. Wong. Application of optical flow techniques to rainfall nowcasting. In the 27th Conference on Severe Local Storms, 2014.

[16] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. DBLP:journals/corr/ChungGCB14,2014

[17] L. Pan, H. Li, W. Li, X. Chen, G. Wu, and Q. Du, "Discriminant analysis of hyperspectral imagery using fast kernel sparse and low-rank graph," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 11, pp. 6085-6098, Nov. 2017.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, Jun. 2017.

[19] G. Zhu, F. Porikli, and H. Li, "Robust visual tracking with deep convolutional neural network based object proposals on pets," in IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Las Vegas, NV, 2016, pp. 1265-1272.

[20] Zhou, S., Shen, W., Zeng, D., Fang, M., Wei, Y., Zhang, Z.: Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes. Signal Processing: Image Communication 47,358–368 (sep 2016)