Université de Montréal

Photovoltaic power and solar radiation forecasting

IFT-6759 - H2022

Kodjovi Adabra (kodjovi.adabra@umontreal.ca)

Marcos A. A. Souto Jr. (marcos.antonio.almeida.souto.junior@umontreal.ca)

Niranjan Niranjan (niranjan.niranjan@umontreal.ca)

Rui Ze Ma (rui.ze.ma@umontreal.ca)

Agenda

1. Introduction
2. Data exploration
3. Reference Model
4. Recurrent Neural Networks
5. Data Augmentation

1. Introduction

The use of solar energy through photovoltaic panels is becoming more and more widespread because of their zero CO2 emissions and their lower cost. However, the variable energy production under changing weather conditions may hinder the large-scale deployment of this technology. In this work inspired by paper [1], we study the "nowcast" prediction of photovoltaic energy from sky images.

The goal is first try to reach the reference study’s performance and then try to improve it.

2. Data exploration

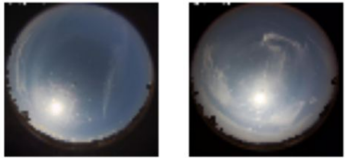
'pv_log_trainval' statistics

	pv_l
count	92975.000000
mean	15.205522
std	7.177062
min	0.000397
25%	9.296960
50%	16.670013
75%	21.374017
max	29.559791

pv_log_test' statistics

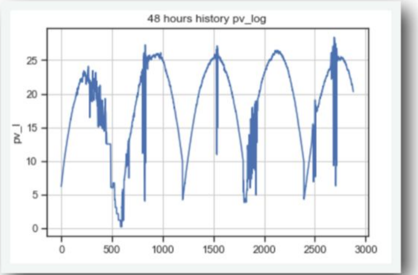
	pv_l
count	9910.000000
mean	14.564921
std	7.474454
min	0.001334
25%	7.893500
50%	16.062818
75%	21.004520
max	29.497691

```
images_trainval_RGB.shape  : (92975, 64, 64, 3)
images_trainval_RGB samples:
```

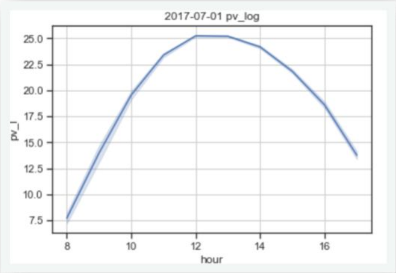


```
images_test_RGB.shape  : (9910, 64, 64, 3)
images_test_RGB samples:
```

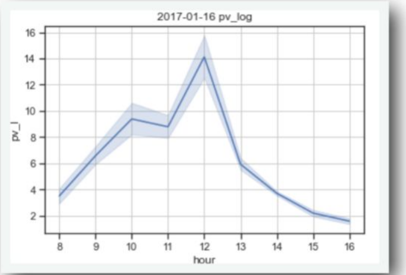
48 hours pv outputs history



pv outputs for a sunny day (2017-07-01)

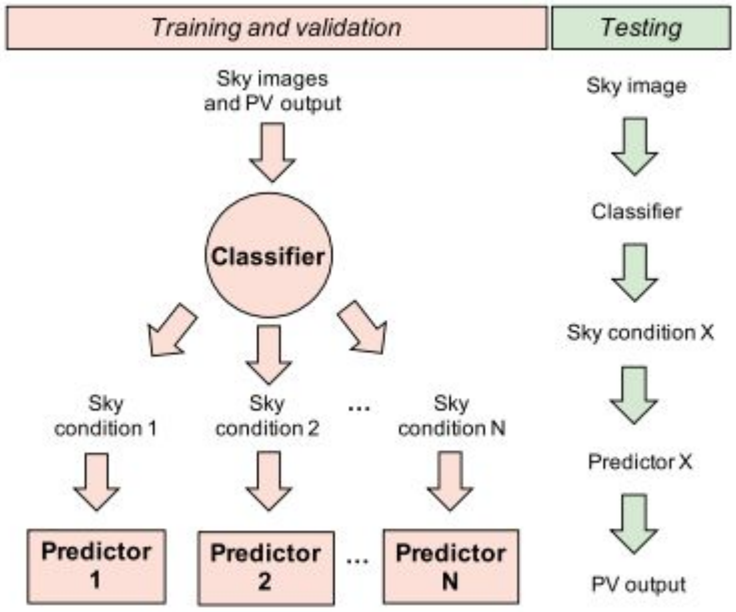


pv outputs for a cloudy day (2018-01-16)



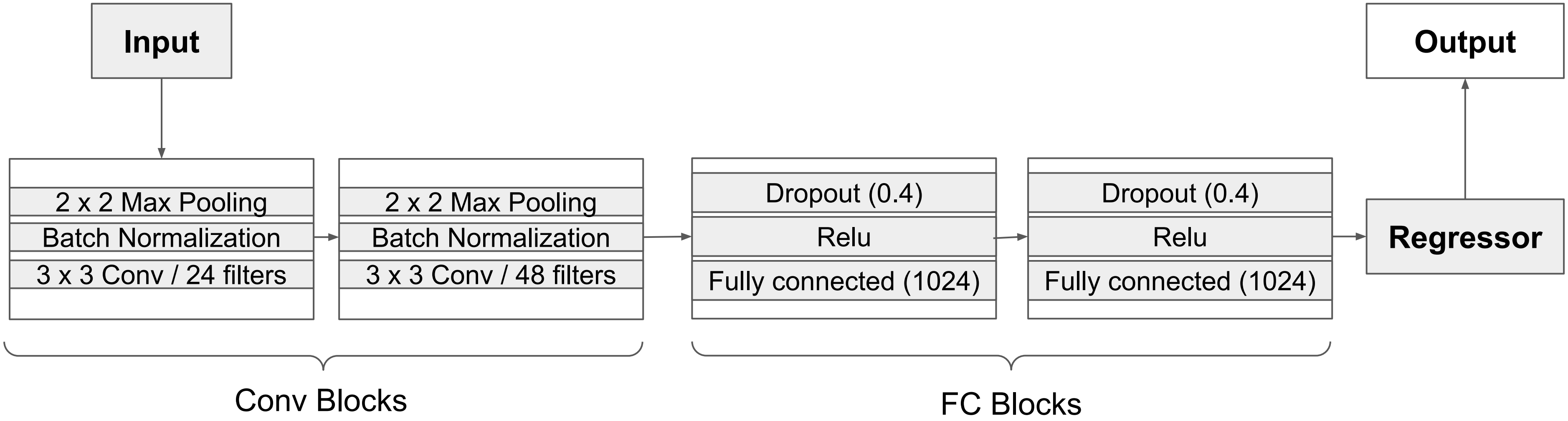
3 - Reference Model

Reference Model - Classification

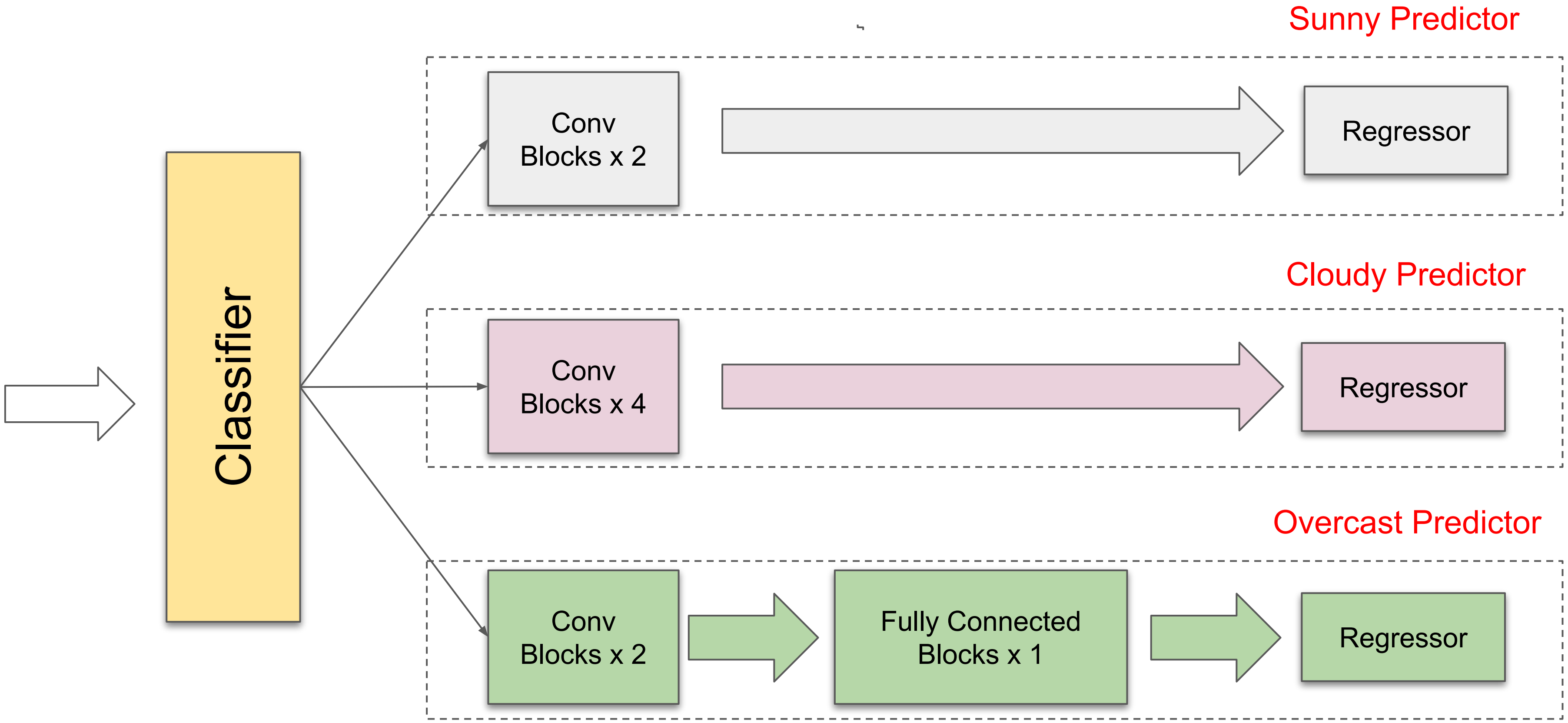


- Classification-Prediction Framework
- Physics-based classifier (non-parametric)
  - Calculating "cloudiness"
  - Taking into account circumsolar pixels
  - Calculate the position of the sun (w/ zenith & azimuth)
- CNN-based classifier (alternative)

Reference Model - Predictor Baseline



Reference Model - Final Architecture



Reference Model - Replication Results

Classification	RMSE
Sunny	1.344
Cloudy	3.626
Overcast	1.793
Overall	**2.310**

Baseline = 2.35

4 - Recurrent Neural Networks

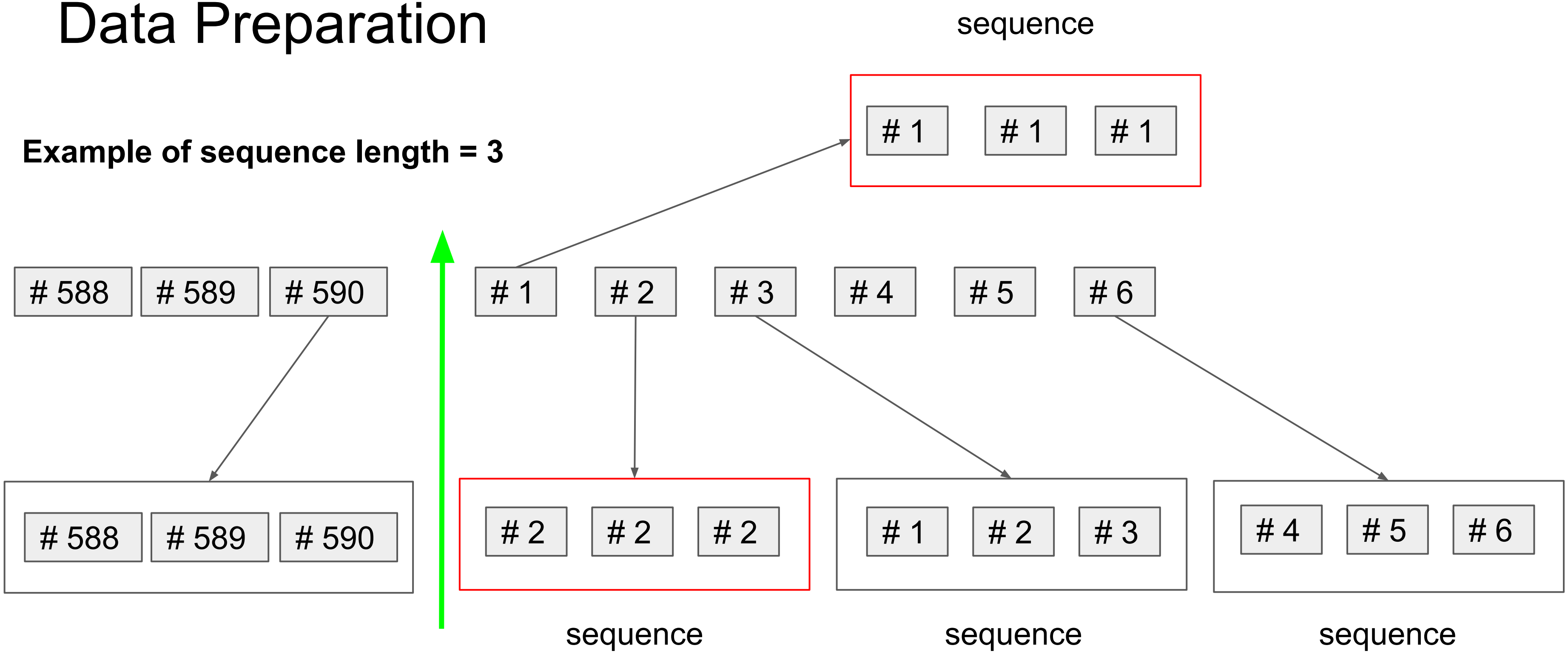
ConvGRU & ConvLSTM

Spatiotemporal Data

- Beyond the CNN model
- Spatiotemporal nature of the data
- Vanilla RNN?
- CNN-RNN
- ConvGRU & ConvLSTM

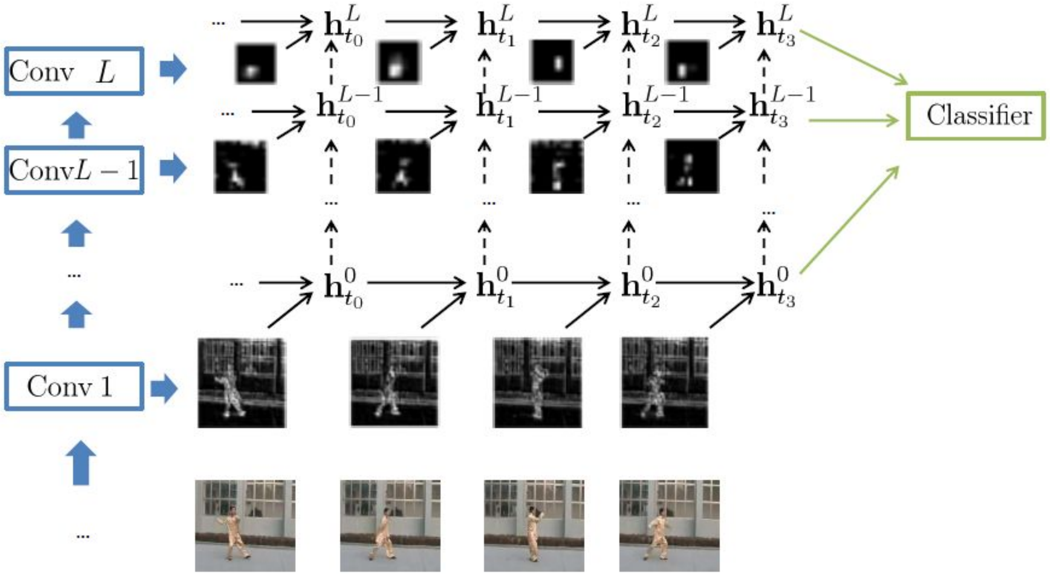
Data Preparation

Example of sequence length = 3



New day start

Multilayer ConvGRU

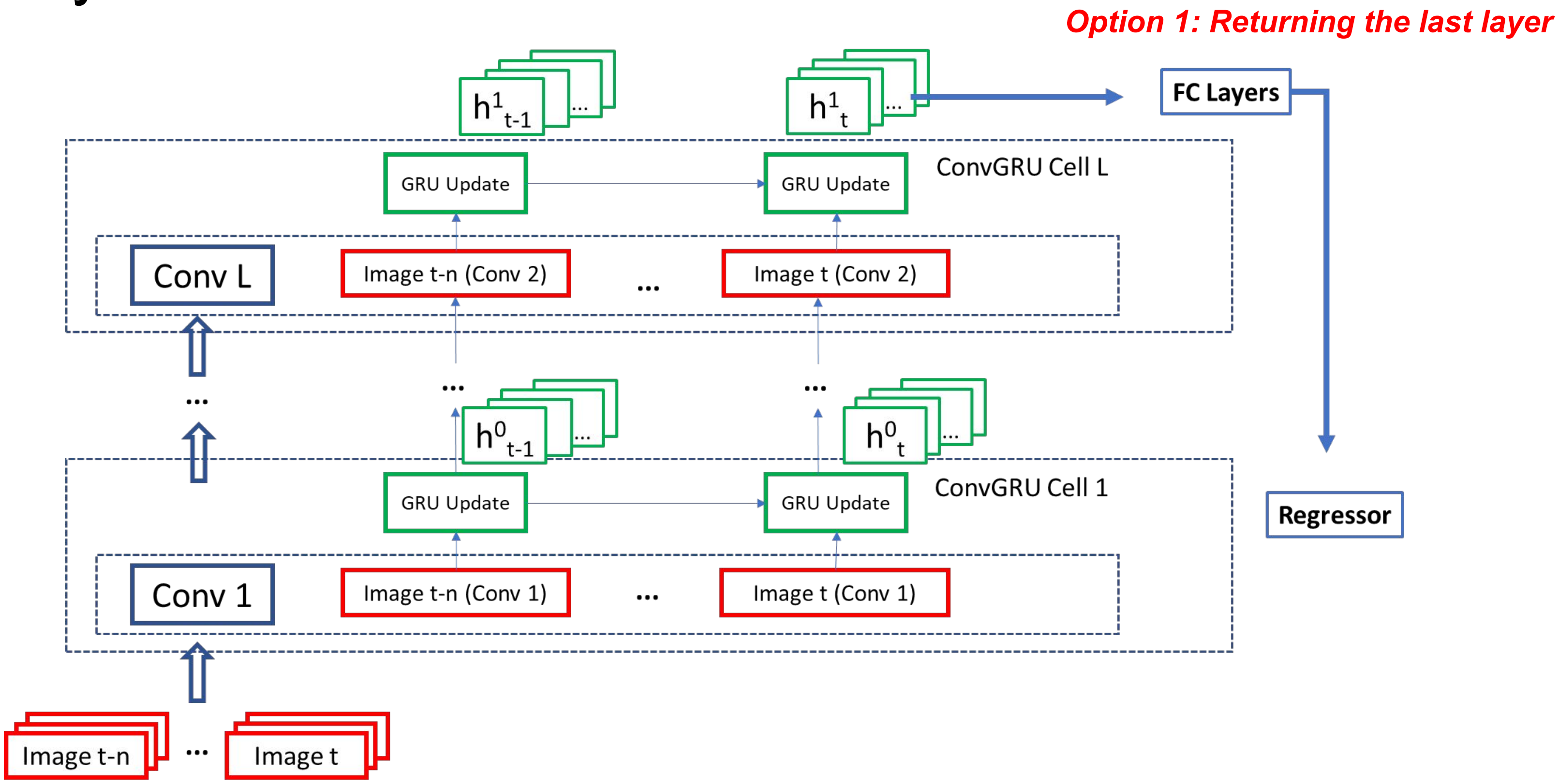


Imlementation ConvGRU:
Copyright 2017 Jacob C. Kimmel

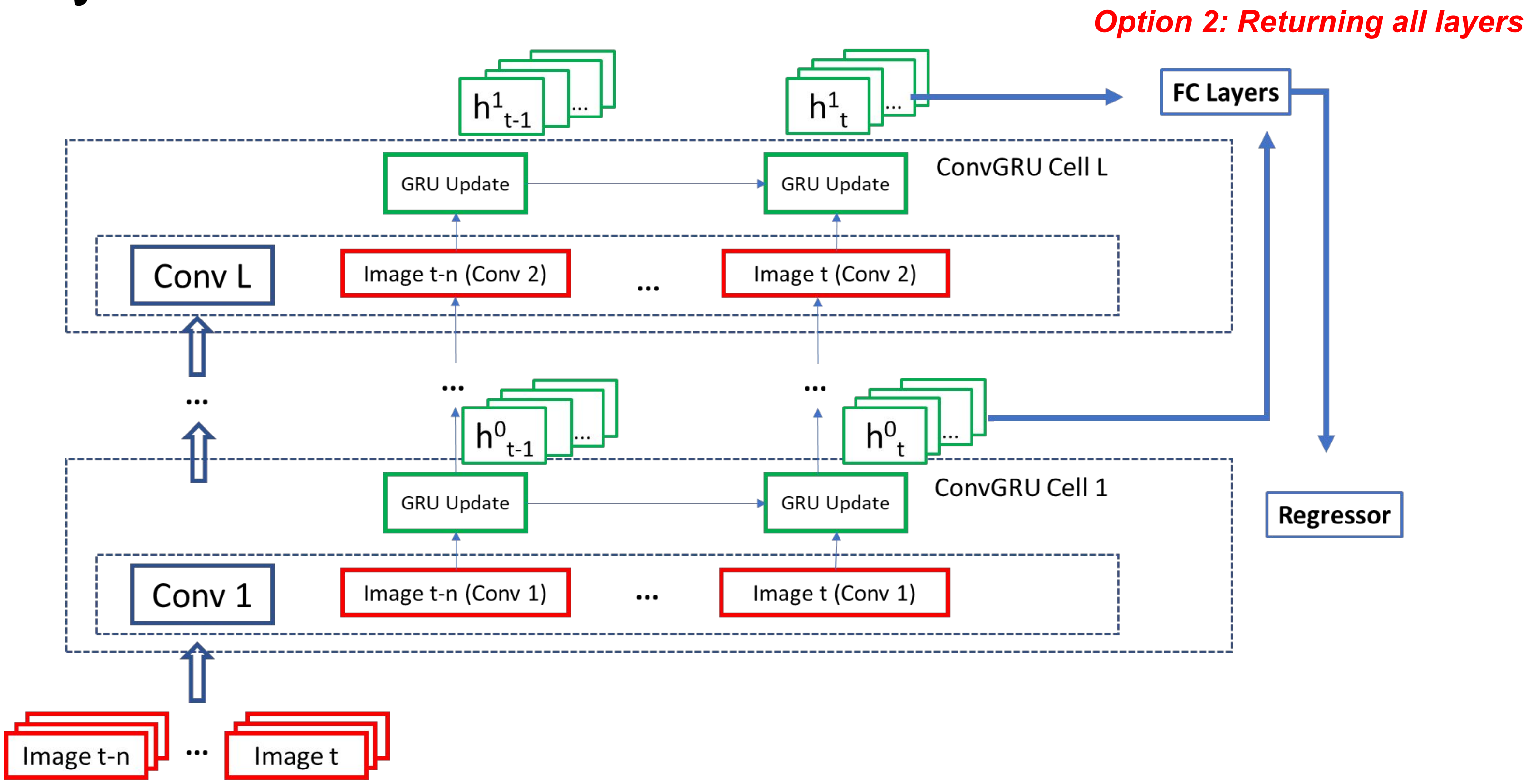
Laboratory of Cell Geometry at the University of California, San Francisco.

Delving Deeper into Convolutional Networks for Learning Video Representations (2015).
Nicolas Ballas, Li Yao, Chris Pal, Aaron Courville

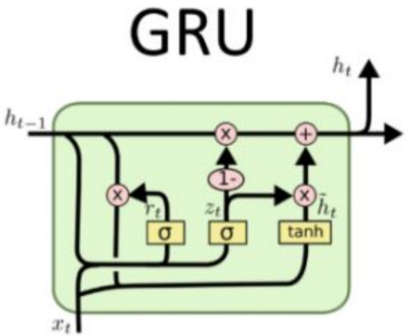
Multilayer ConvGRU



Multilayer ConvGRU



ConvGRU Cell



Update, reset gates:

$$\mathbf{z}_t = \mathrm{sigm}(\mathbf{b}_{[z]} + \mathbf{U}_{[z]}\mathbf{h}_{t-1} + \mathbf{W}_{[z]}C(w_t))$$

$$\mathbf{r}_t = \mathrm{sigm}(\mathbf{b}_{[r]} + \mathbf{U}_{[r]}\mathbf{h}_{t-1} + \mathbf{W}_{[r]}C(w_t))$$

Cell state:

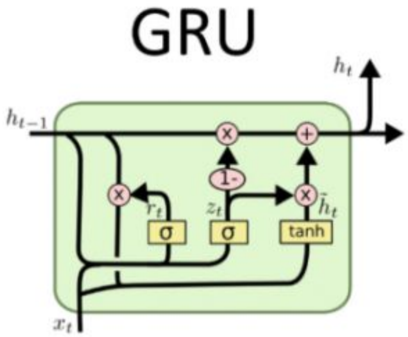
$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{b}_{[c]} + \mathbf{U}_{[c]}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{W}_{[c]}C(w_t))$$

$$\mathbf{c}_t = (1 - \mathbf{z}_t) \odot \mathbf{c}_{t-1} + \mathbf{z}_t \odot \widetilde{\mathbf{c}}_t$$

Hidden layer:

$$\mathbf{h}_t = \mathbf{c}_t$$

ConvGRU Cell



Update, reset gates:

$$\mathbf{z}_t = \mathrm{sigm}(\mathbf{b}_{[z]} + \mathbf{U}_{[z]}\mathbf{h}_{t-1} + \mathbf{W}_{[z]}C(w_t))$$

$$\mathbf{r}_t = \mathrm{sigm}(\mathbf{b}_{[r]} + \mathbf{U}_{[r]}\mathbf{h}_{t-1} + \mathbf{W}_{[r]}C(w_t))$$

Cell state:

$$\widetilde{\mathbf{c}}_t = \tanh(\mathbf{b}_{[c]} + \mathbf{U}_{[c]}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{W}_{[c]}C(w_t))$$

$$\mathbf{c}_t = (1 - \mathbf{z}_t) \odot \mathbf{c}_{t-1} + \mathbf{z}_t \odot \widetilde{\mathbf{c}}_t$$

Convolutional operations

Hidden layer:

$$\mathbf{h}_t = \mathbf{c}_t$$

Results - RMSE

Option 1:
Returning the last layer

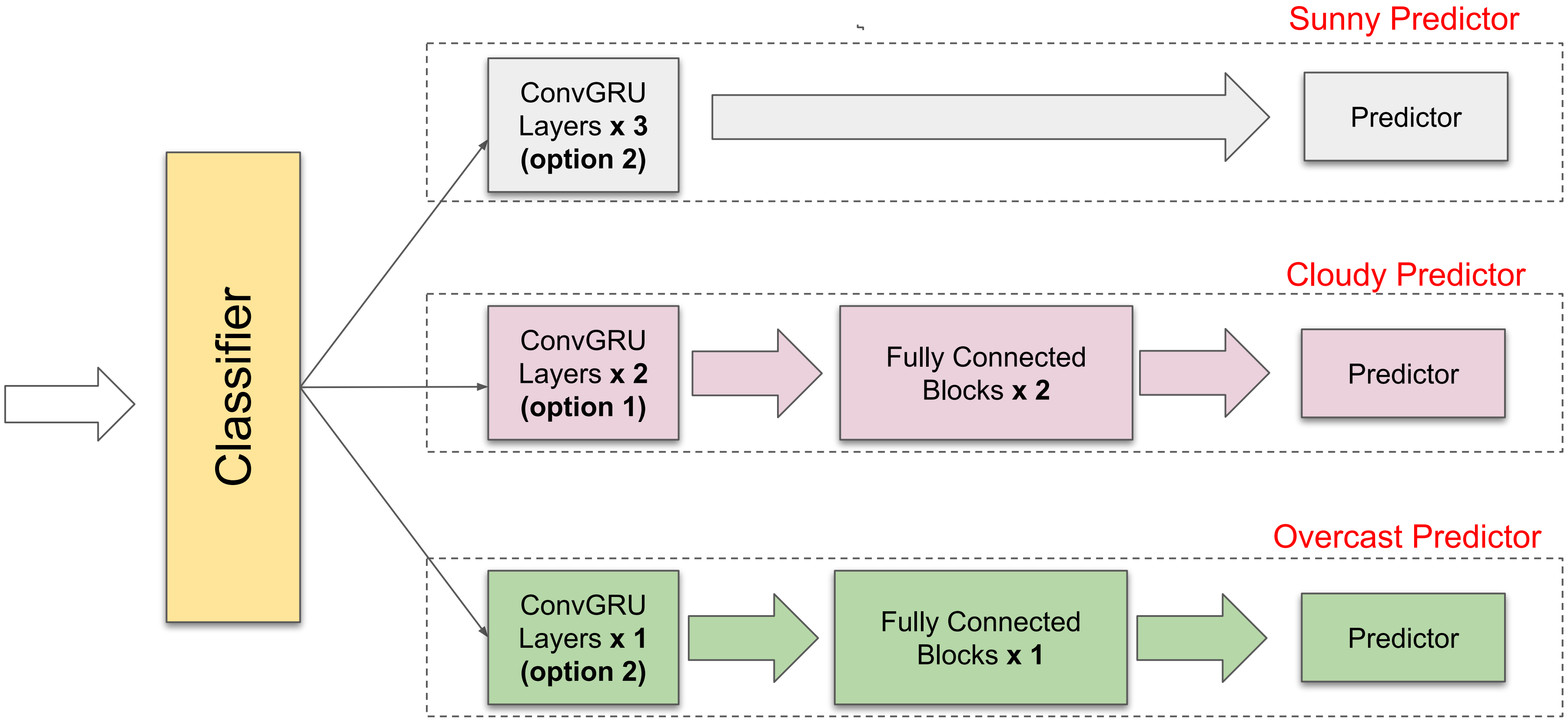
	Seq length	# ConvGRU Layers	FC = 0	FC = 1	FC = 2
Sunny	2	2	1.38	1.32	**1.3**
Cloudy	2	2	4.053	3.64	**3.5**
Overcast	2	2	2.121	1.79	**1.72**

Option 2:
Returning all layers

	Seq length	# ConvGRU Layers	FC = 0	FC = 1	FC = 2
Sunny	2	3	**1.286**	1.320	1.359
Cloudy	2	3	3.577	3.579	**3.559**
Overcast	2	1	1.951	**1.682**	2.012

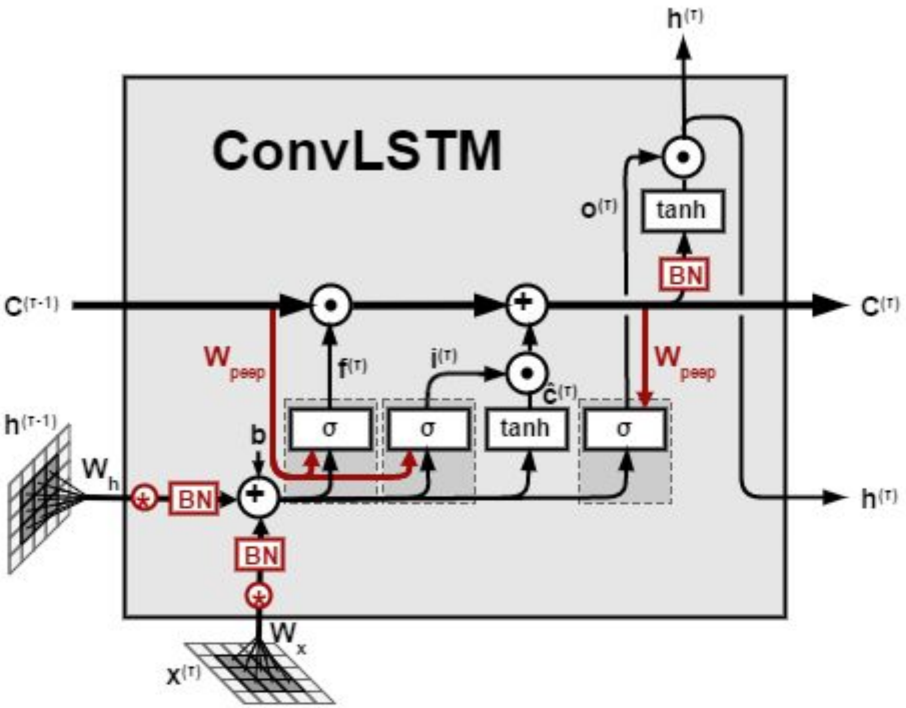
Overall RMSE = 2.221

ConvGRU PV Prediction Architecture



ConvLSTM

- LSTMs: similar to GRUs
- Cell state updated over time
- Modified to accommodate 2D images:
  - Convolution instead of matrix multiplication
- The state of each pixel depends on:
  - Local neighbours (present and past)

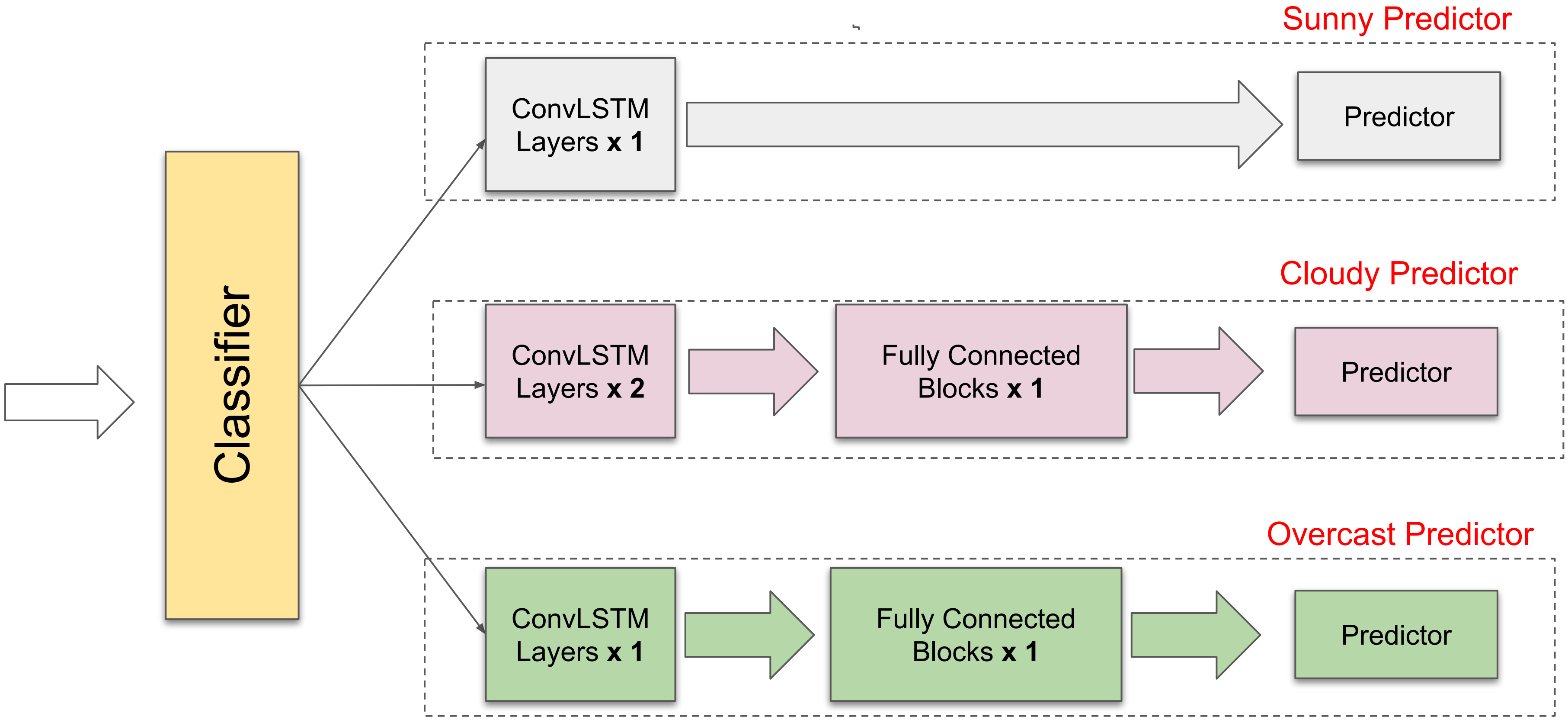


ConvLSTM - Results

	# ConvLSTM layers	# FC = 0	# FC = 1	# FC = 2
Sunny	1	**1.275**	1.281	1.338
	2	1.440	1.325	1.343
	3	1.522	1.330	1.295
Cloudy	1	3.663	3.573	3.538
	2	3.963	**3.524**	3.559
	3	4.207	3.656	3.567
Overcast	1	1.978	**1.831**	1.864
	2	2.010	1.937	2.126
	3	2.16	2.125	2.008

Overall RMSE = 2.238

ConvLSTM - Final Architecture



5 - Data Augmentation

5. Data Augmentation using GAN

- **The model** : basic Wasserstein GAN with gradient penalty
- **The process**:

  1- Training and generation of 10000 fakes sky images with GAN.

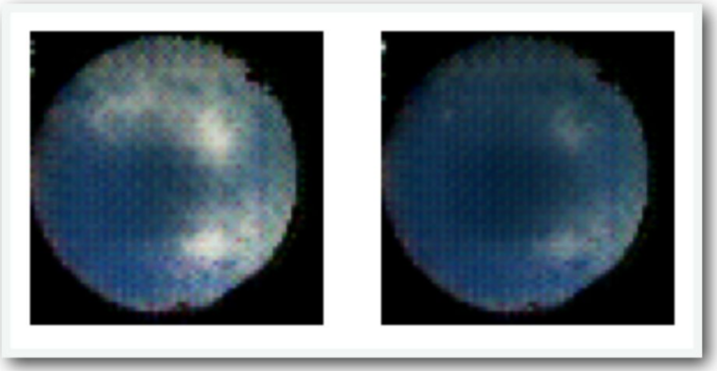
  2- Prediction of the pv log of the 10000 fakes images with Sunset

  3- Training / validation of the Sunset model with the augmented dataset (reals + fakes).

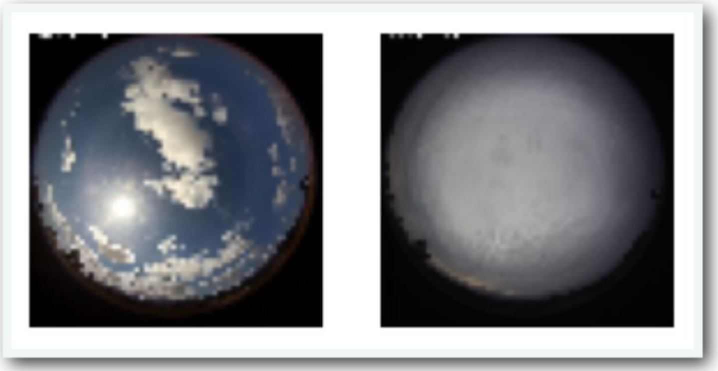
  4- Use of the new model for forcasting on testset data.

- **Fakes generated sky images samples:**

fake sky images samples



real sky images samples



5. Data Augmentation using GAN

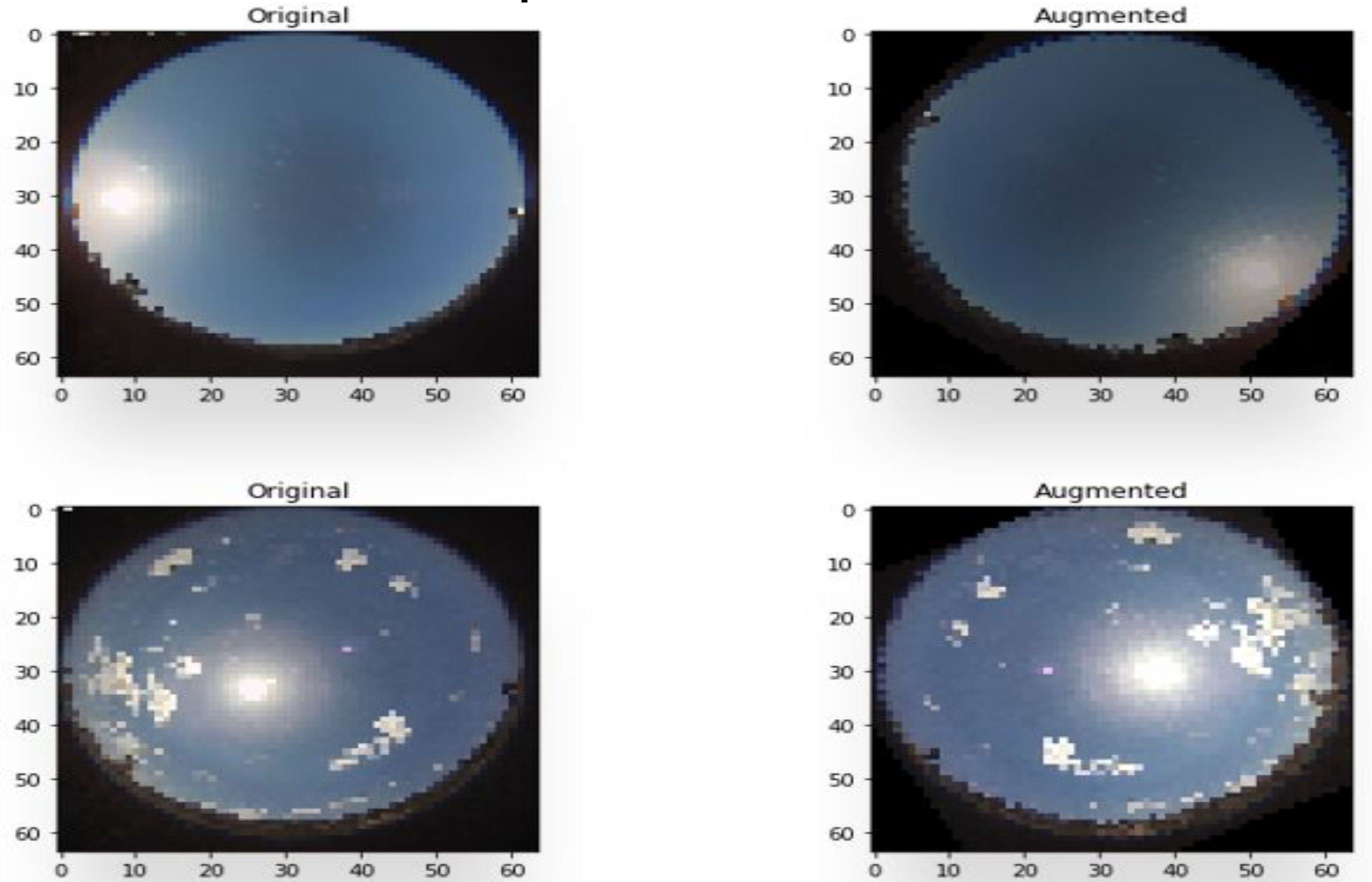
- **Results**: **the rmse has degraded.**
- **Possible reasons for this degradation**:
    1. Failure to take into account the datetime features during gan training process.
    2. Use of a basic Wasserstein GAN with a gradient penalty model.
- **Possible solutions**:
    1. Integration of date time feature for the gan training process.
    2. Use of a Conditional GAN model that allows to get generated and diversified examples in a specific class limited only to this class (ie sunny or cloudy or overcast) rather than a random generation.
    3. Using a controllable GAN model that allows us to control the amount of a particular feature we want, like sky brightness or sun positions etc ...

Exploring data augmentation

Augmentation techniques used:

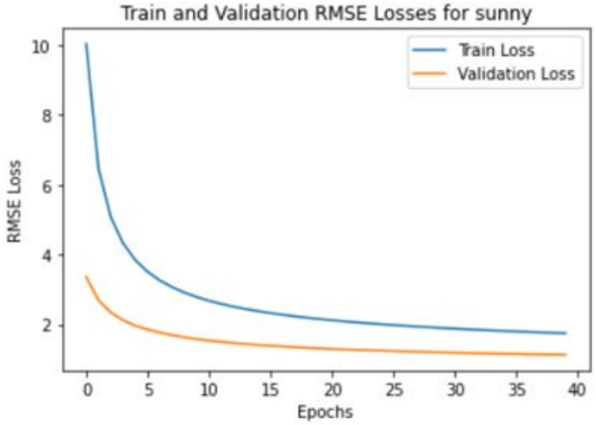
- Spatial Augmentation (rotation + flips)
- Pixel Augmentation (contrast + brightness)
- Spatial + Pixel Augmentation

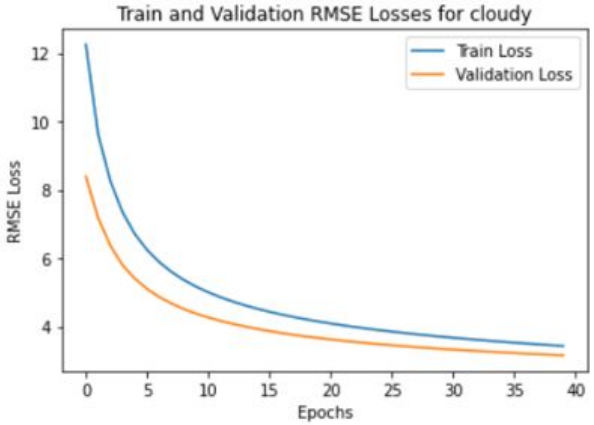
Augmentation at a Glimpse

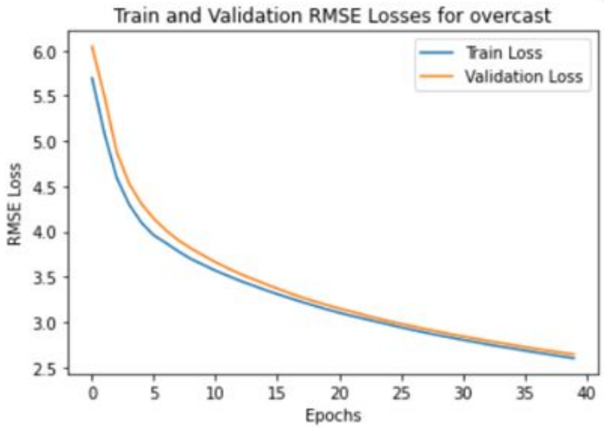


Results without augmentation

Model	RMSE
Sunny	1.344
Cloudy	3.626
Overcast	1.793
Final_result	2.310



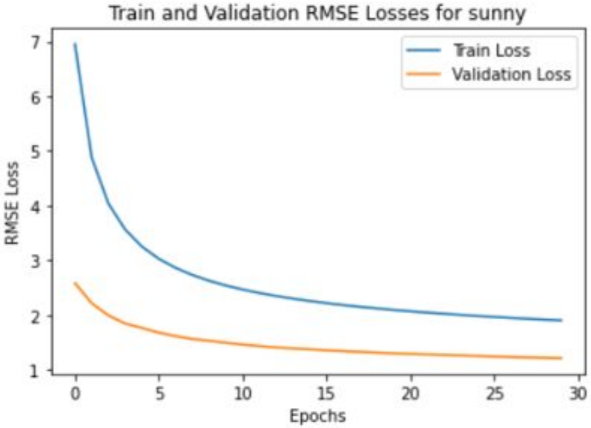


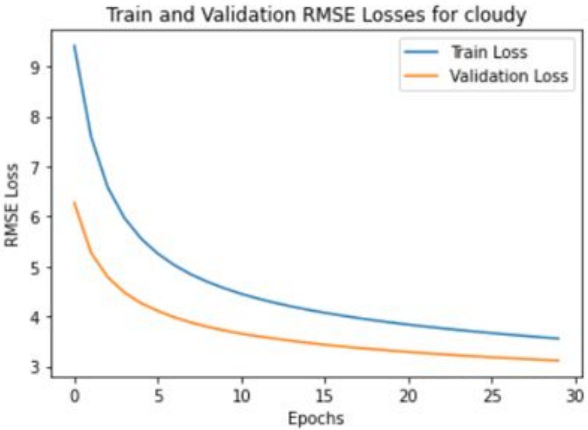


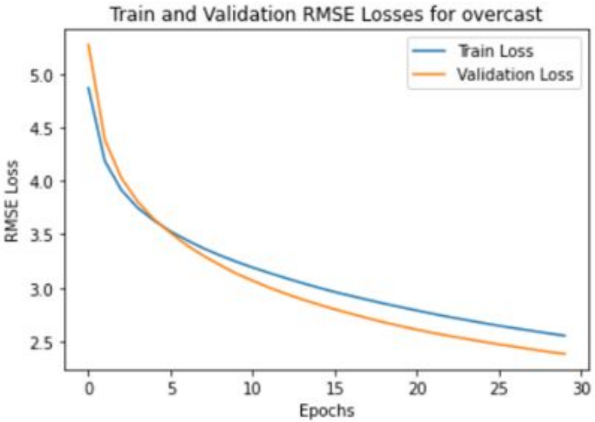
Without Augmentation

Spatial Augmentation

Model	Without Augmentation	With Augmentation
Sunny	1.344	1.335
Cloudy	3.626	3.532
Overcast	1.793	1.824
Final	2.310	2.262



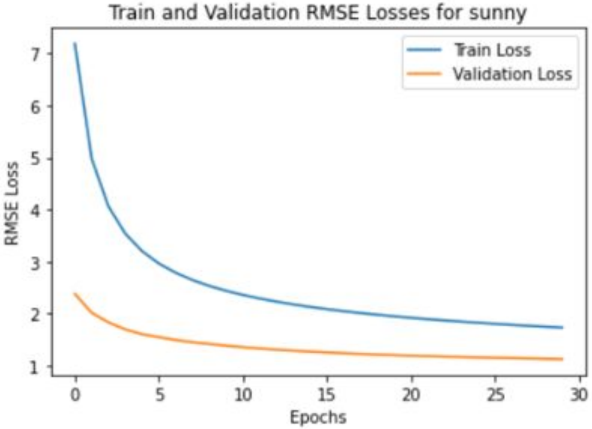


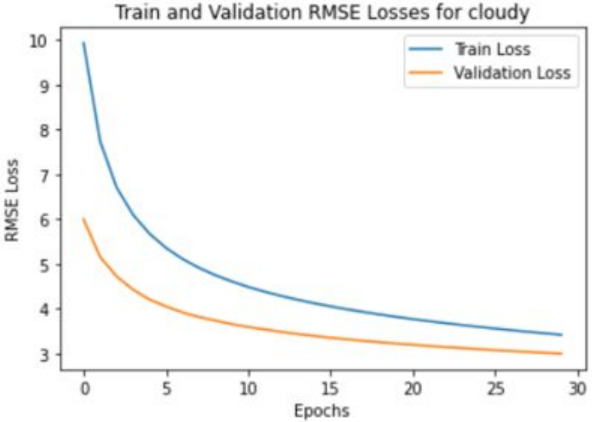


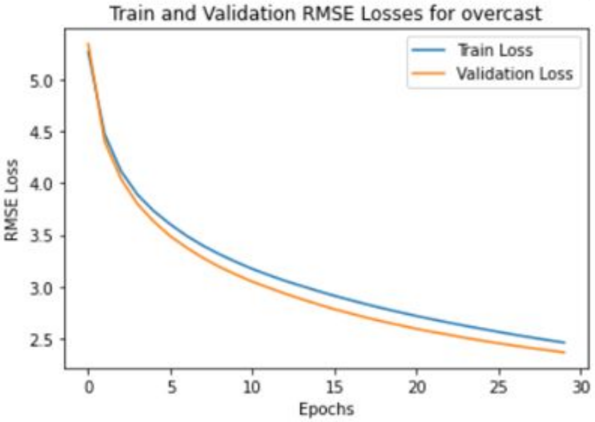
Spatial Augmentation

Pixel Augmentation

Model	Without Augmentation	With Augmentation
Sunny	1.344	1.413
Cloudy	3.626	3.637
Overcast	1.793	1.841
Final	2.310	2.340



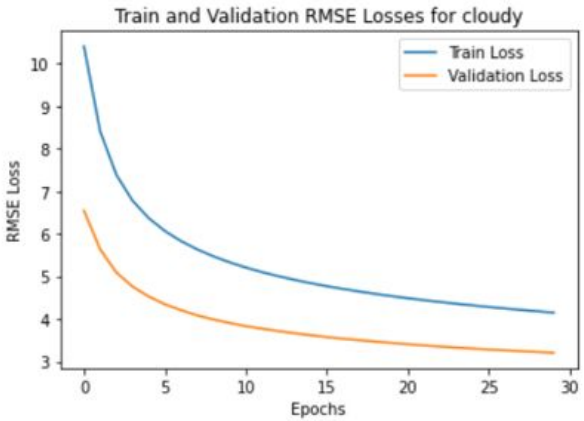


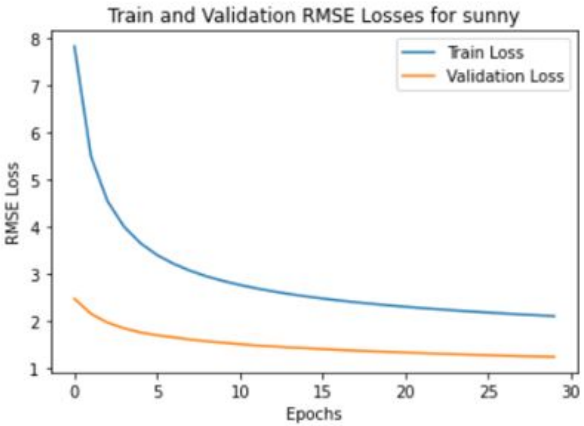


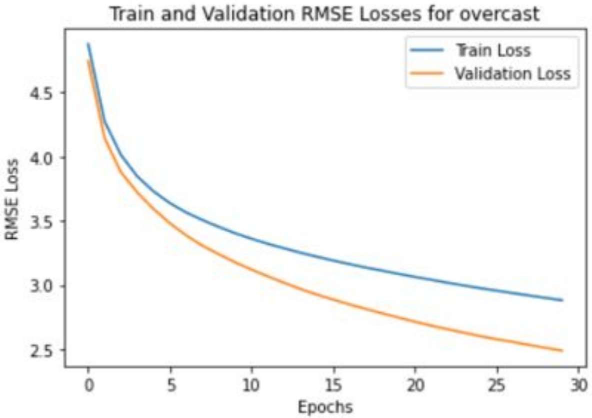
Pixel Augmentation

Pixel + Spatial Augmentation

Model	Without Augmentation	With Augmentation
Sunny	1.344	1.429
Cloudy	3.626	3.694
Overcast	1.793	2.844
Final	2.310	2.372







Spatial + Pixel Augmentation