# **CSYE 7245 - Big Data Sys and Int Analytics**

## **Lab - SEVIR Notebook**

### **Team Information:**

|  |  |
| --- | --- |
| **Name** | **NUID** |
| Tanvi Gurav | 001306848 |
| Keerti Ojha | 001050173 |
| Priyanka Malpekar | 001302741 |

**Lab Completion Date:** 12th February’21

# **About**

This lab demonstrates analyzing the results of training, visualizing the results on Synthetic Radar Data from SEVIR Dataset (Storm Event Imagery Dataset for Deep Learning Applications in Radar and Satellite Meteorology) using the pretrained models.

**Why this use case?**

The Earth’s weather is continuously monitored by sensors that collect terabytes of data every day. Over the US, satellite observations provided by GOES-R series satellites (GOES-16 & GOES-17), and weather radar data provided by the national network of WSR-88D (NEXRAD) radars are two major sources of weather sensing used by forecasters, decision makers, and the general public.

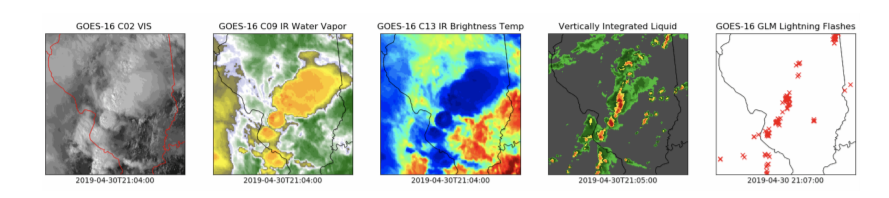
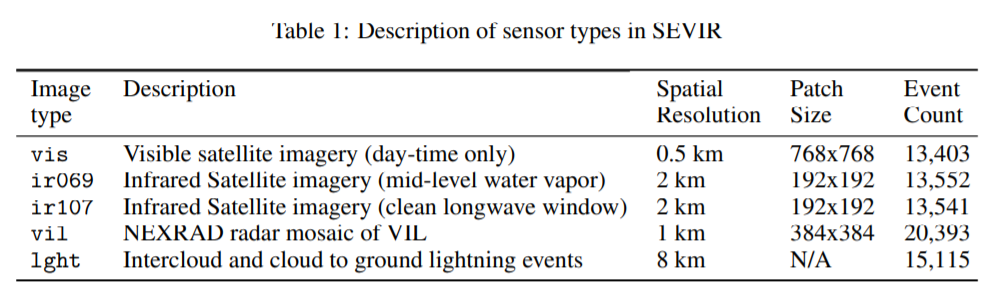


Figure 1: The Storm EVent ImagRy (SEVIR) dataset contains over 10,000 spatially and temporally aligned sequences across the five image types.

Archives of these two sensing modalities make up petabytes of data that include images of clouds in both visible and infrared, depictions of precipitation intensity, and detection of lightning. Recently, there has been a great deal of work to use deep learning to better leverage these rich data sources, specifically for applications like short term weather forecasting, synthetic weather radar for areas lacking traditional weather radar, improved data assimilation for improved numerical weather prediction, and many others. Furthermore, access to datasets like GOES & NEXRAD is becoming easier as cloud services such as Google Earth Engine, Amazon Open Data Registry, IBM’s PAIRS and others provide access to Earth system datasets.

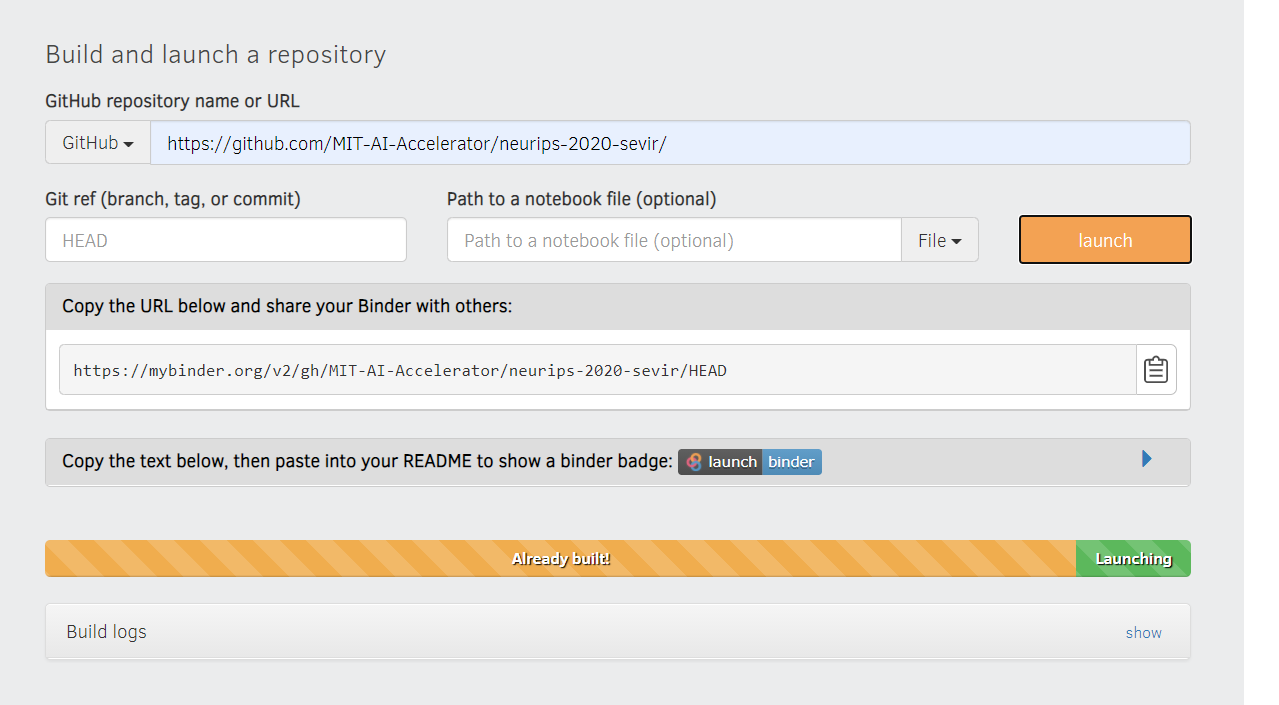
# **Dataset**

SEVIR is a collection of temporally and spatially aligned image sequences depicting weather events captured over the contiguous US (CONUS) by GOES-16 satellite and the mosaic of NEXRAD radars. Figure 1 shows a set of frames taken from a SEVIR event. SEVIR contains five image types: GOES-16 0.6 µm visible satellite channel (vis), 6.9 µm and 10.7 µm infrared channels (ir069, ir107), a radar mosaic of vertically integrated liquid (vil), and total lightning flashes collected by the GOES-16 geostationary lightning mapper (GLM) (lght). See Table 1 for details. Each event in SEVIR consists of a 4-hour length sequence of images sampled in 5 minute steps. The lightning modality is the only non-image type, and is represented by a collection of GLM lightning flashes captured in the 4 hour time window. SEVIR events cover 384 km x 384 km patches sampled at locations throughout the continental U.S. (CONUS). The pixel resolution in the images differ by image type, and were chosen to closely match the resolution of the original data. Since the patch dimension of 384 km is constant across sensors, the size of each image differs (as shown in Table 1).



# **Experiment Setup**

We implemented this lab using <https://mybinder.org/> which helps to open notebooks in an executable environment, making your code immediately reproducible and launch our [neurips-2020-sevir](https://github.com/MIT-AI-Accelerator/neurips-2020-sevir) repository as follows:



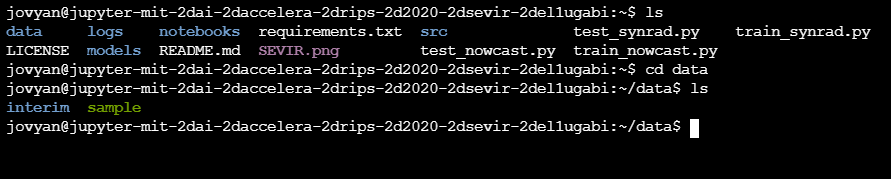
Once the repository is build, we can access it in a Juypter Dashboard environment with the folder structure similar to that in **neurips-2020-sevir** repository



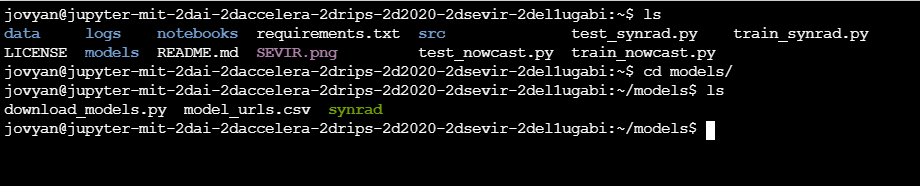
**Downloading pretrained models**

Before downloading the models, we created the following folders for downloaded data:

1. /data/sample: Location for placing the synrad testing sample data



2. models/synard: Location for placing the pretrained models



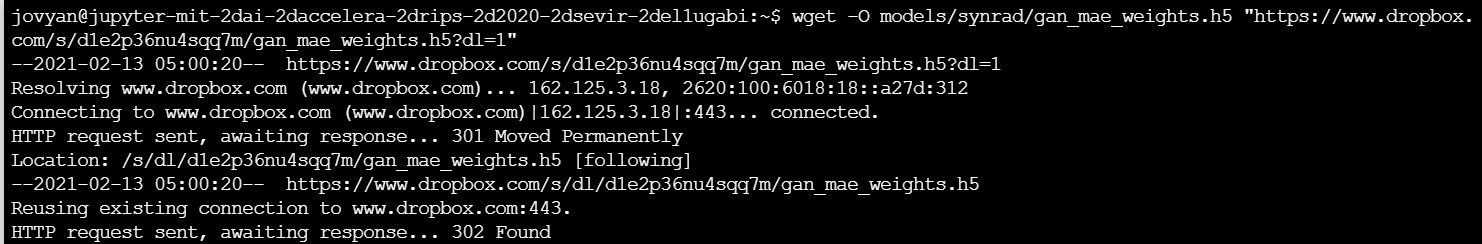
To download the models trained in the paper, we can execute the following:

cd models/

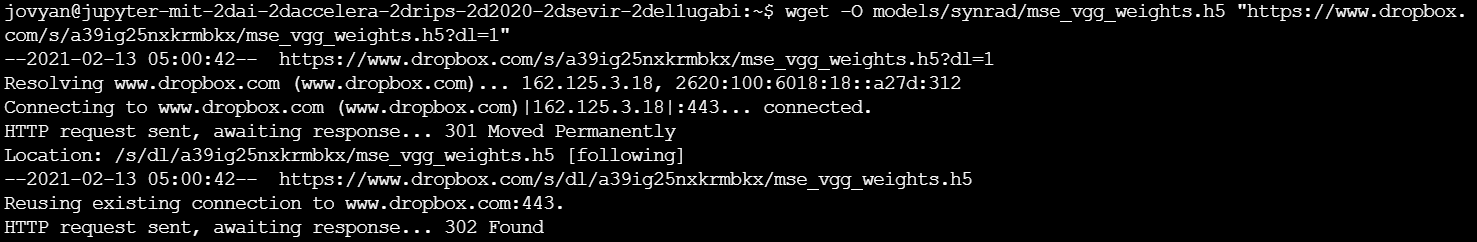
python download\_models.py

**Alternatively, we used the below command to download the pretrained models:**

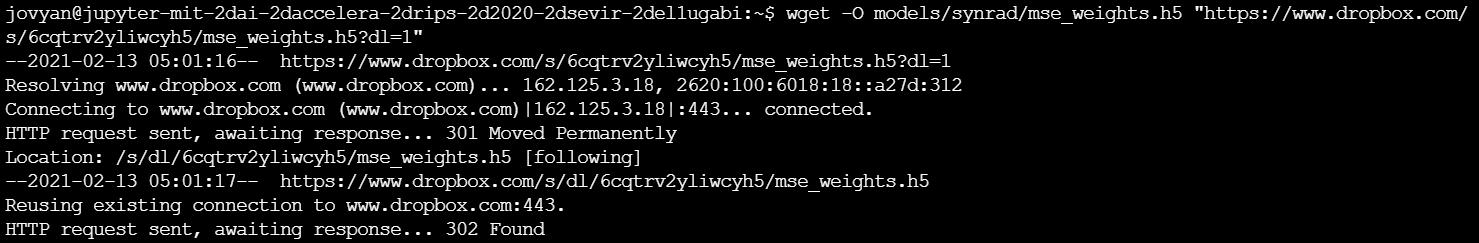
wget -O models/synrad/gan\_mae\_weights.h5 "<https://www.dropbox.com/s/d1e2p36nu4sqq7m/gan_mae_weights.h5?dl=1>"



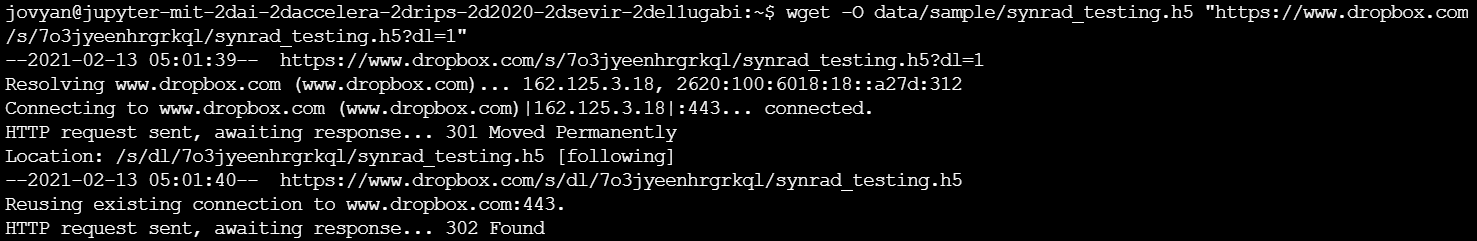
wget -O models/synrad/mse\_vgg\_weights.h5 "<https://www.dropbox.com/s/a39ig25nxkrmbkx/mse_vgg_weights.h5?dl=1>"



wget -O models/synrad/mse\_weights.h5 "<https://www.dropbox.com/s/6cqtrv2yliwcyh5/mse_weights.h5?dl=1>"



wget -O data/sample/synrad\_testing.h5 "<https://www.dropbox.com/s/7o3jyeenhrgrkql/synrad_testing.h5?dl=1>"

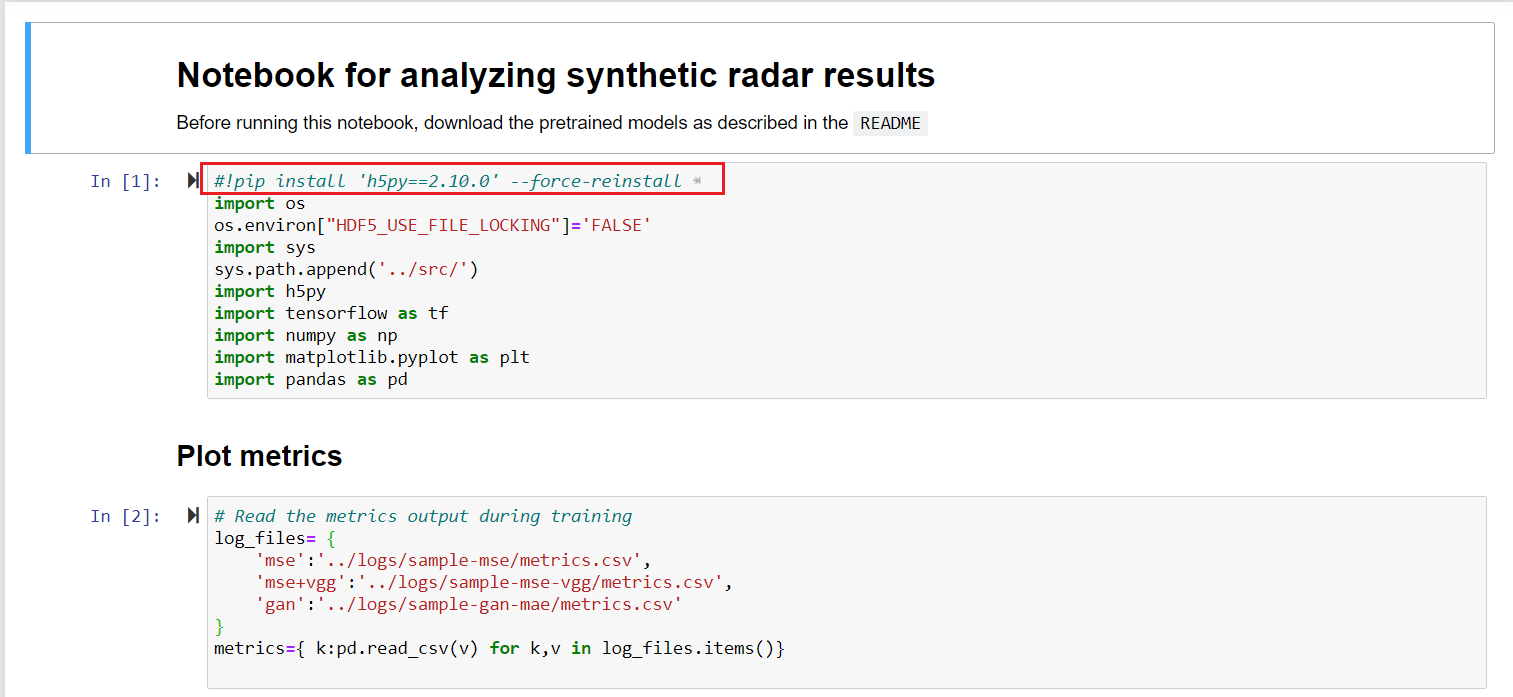


Navigate through the notebooks folder and access the **AnalyzeSyntheticRadar.ipynb** notebook which we implemented analysis of synthetic radar data

**Path:** neurips-2020-sevir/notebooks/AnalyzeSyntheticRadar.ipynb

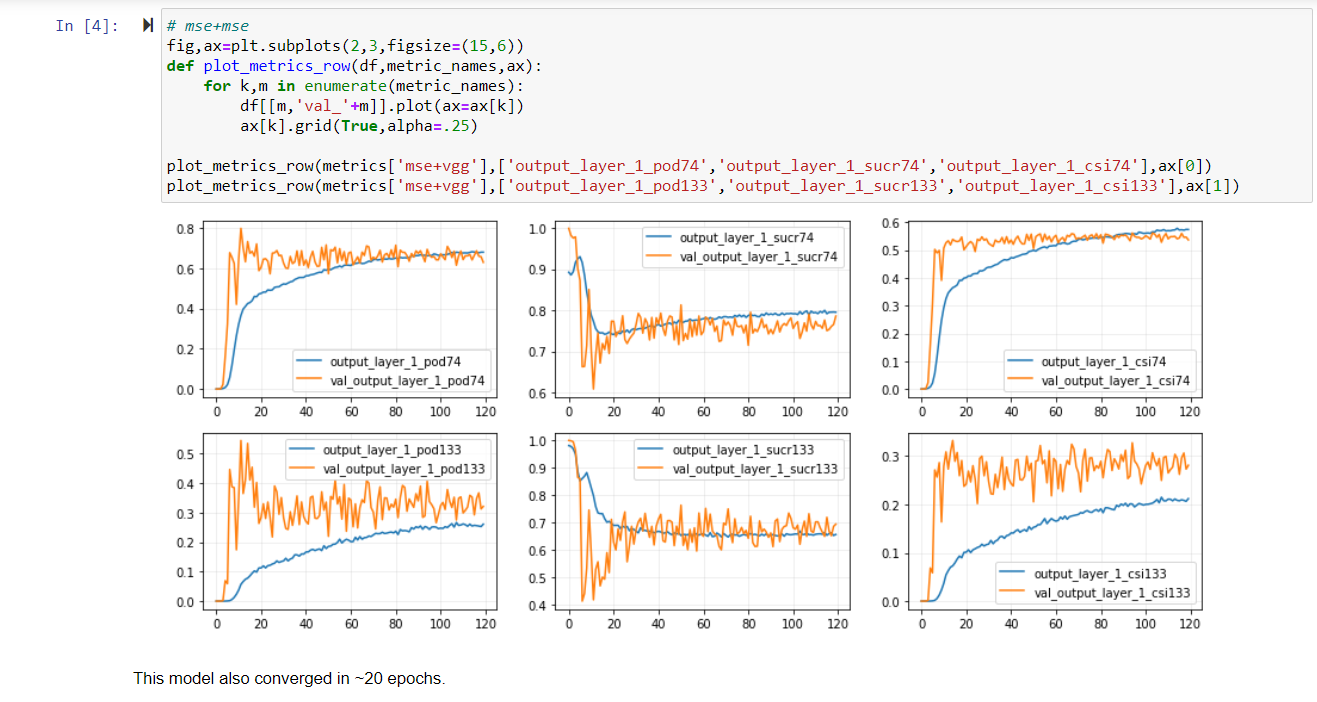
# **Tests**

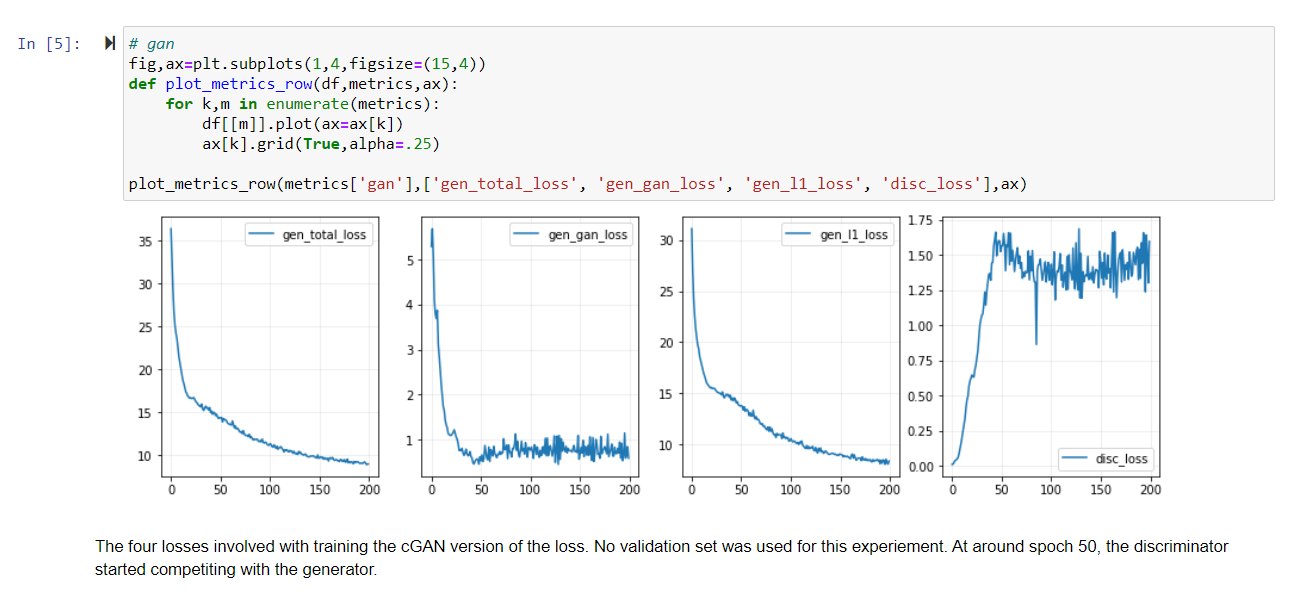
**Installed the required libraries:**

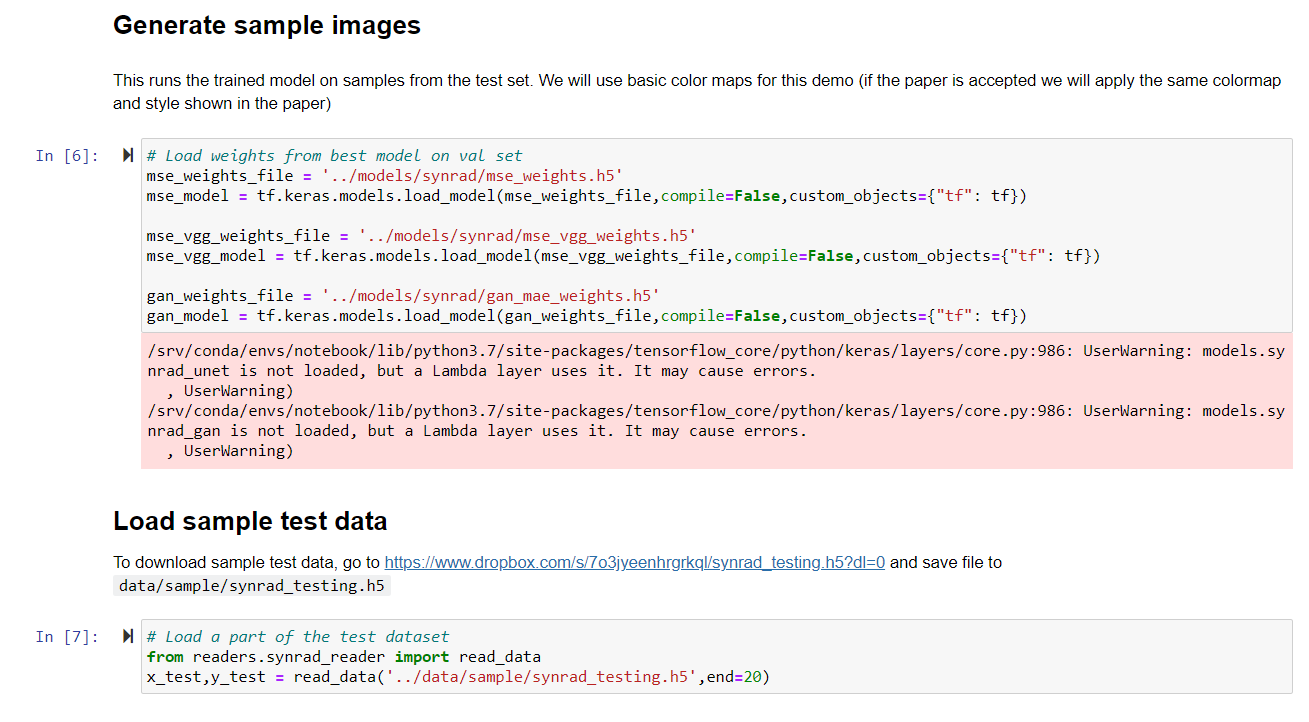


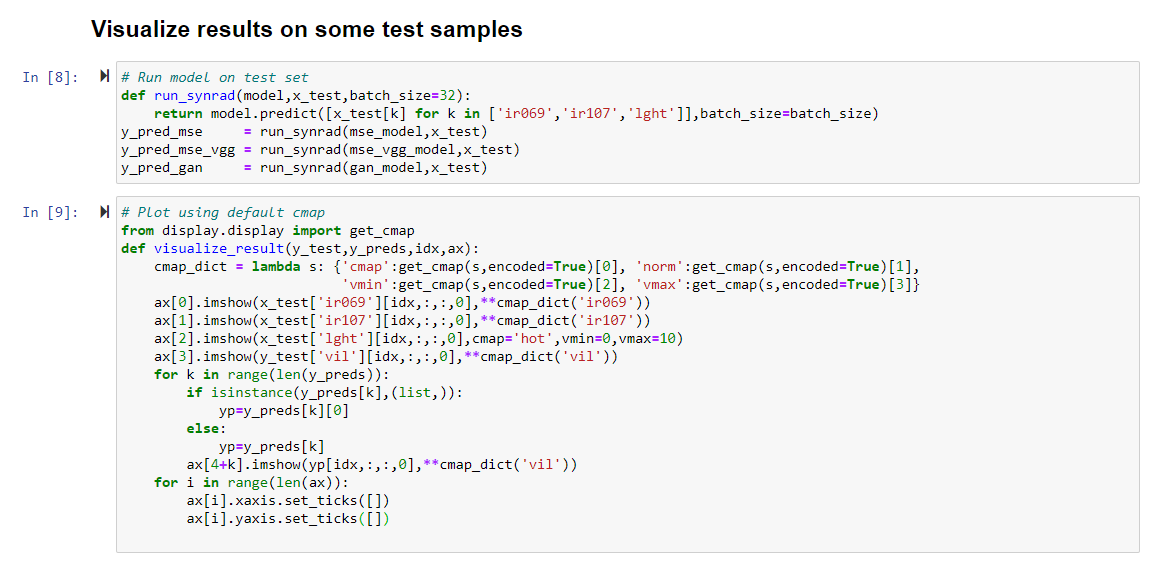
**Plotting the metrics:**





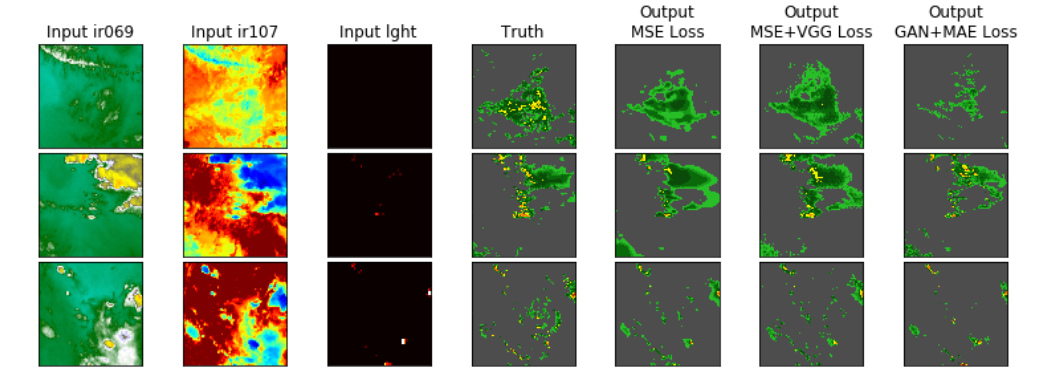






# **Results**

After running the trained models on samples from the test set, we used the basic color maps and loaded part (1000 data points) of the test dataset, ran the model and finally plotted output loss along with inputs using default cmap which ideally resulted in the following cmap:



Three examples of the synthetic weather radar model trained using three different loss functions. MSE leads to an accurate, albeit overly smoothed, prediction. The content and adversarial losses are able to provide additional textures that are visually more similar to the target.

# **Lessons learned**

1. Learned the concept of transferred learning that is utilizing pretrained model for sample test

2. Learned the usage of TensorFlow and its library called Keras to load pretrained models

3. Learned how to leverage executable environment tools like mybinder to replicate or reproduce code

# **References**

<https://proceedings.neurips.cc/paper/2020/file/fa78a16157fed00d7a80515818432169-Paper.pdf>

<https://github.com/MIT-AI-Accelerator/neurips-2020-sevir>