# Detecting anomalous precipitation events using a deep learning autoencoder

**Hiroyuki Murakami**

**Geophysical Fluid Dynamics Laboratory**
**Hiroyuki.Murakami@noaa.gov**

Tutorial Lecture at the 2022 Princeton AOS workshop

# Motivation

## How do we define extreme hydroclimate events?

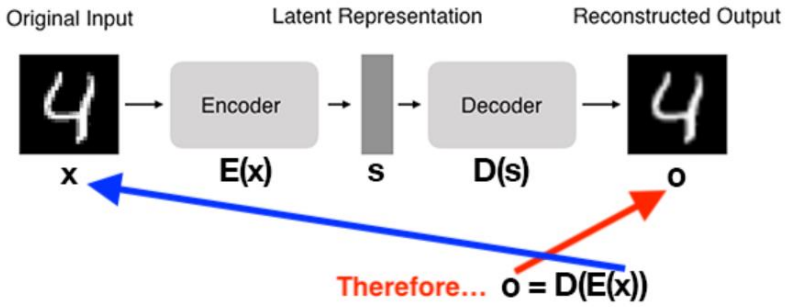

We tend to define them as extremely “intense” events (e.g., 100F temperature, 100 mm/day rainfall, etc.).

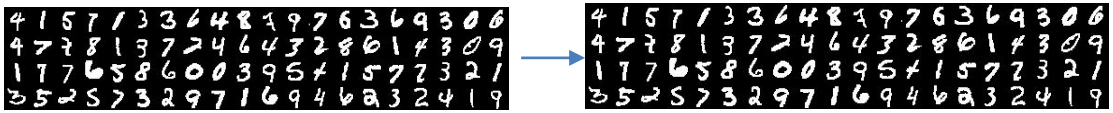
But extreme events should be defined not only by the intensity but also by the ***rarity*** of the events considering both spatial pattern and intensity.

Unlike the events exceeding the intensity threshold, there may be some rare events (or say outlier) that we didn’t recognize. But how can we objectively detect outlier events?

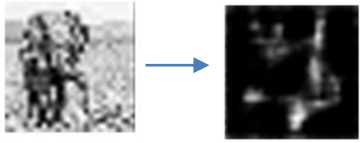
Murakami et al. (2022, *Earth’s Future*) developed a deep learning method “autoencoder” to objectively extract outlier hydroclimate events.

# What is autoencoder?



Input Images (X) D(E(x)) Output Images (o)



- Suppose there are many numeric images. In contrast, the elephant image is clearly an outlier. If an image other than a numeric image is shown, an autoencoder somehow recognizes the image as an outlier.
- Autoencoder doesn't do anything. But it compresses 2-D pixels to latent array s (E), then restore them to the original 2-D pixel (D). The parameters used for E and D are trained using a neural network using a lot of data.
- The trained autoencoder applied to numeric images can easily restore the original image.
- However, the autoencoder failed to restore the elephant image because it is not trained well to restore the rare image. The image that failed to restore can be considered as an outlier.

# Other examples of anomaly detections using Autoencoder

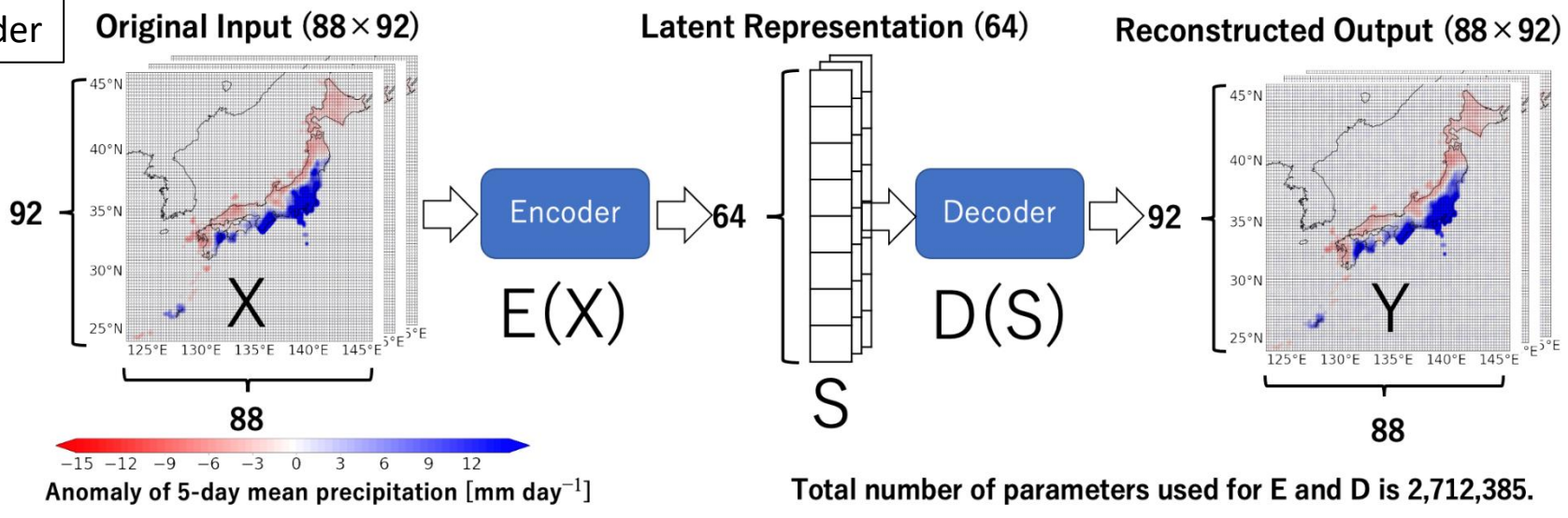
Ex1) Forest Dataset


Ex2)

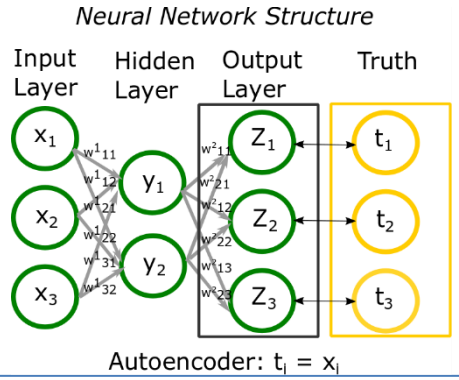
Ex3)

# Autoencoder applied to precipitation events

## Autoencoder



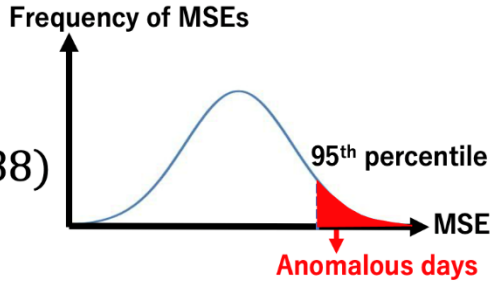

## Applied Convolutional Neural Network




## Definition of Outliers

$$Autoencoder\text{: } Y = D(E(X))$$

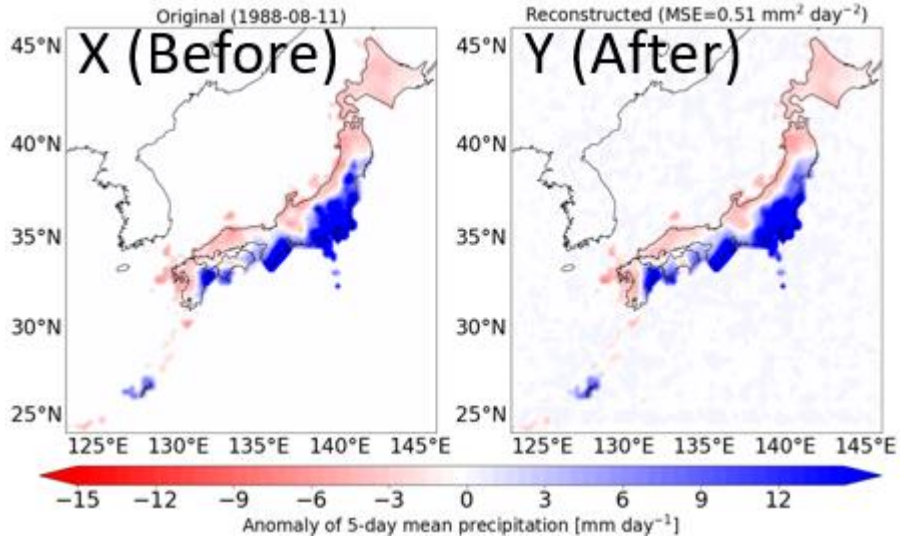
$$MSE(t) = \sum_{j=1}^{92}\sum_{i=1}^{88}\left(Y_{i,j,t} - X_{i,j,t}\right)^2/(92 * 88)$$



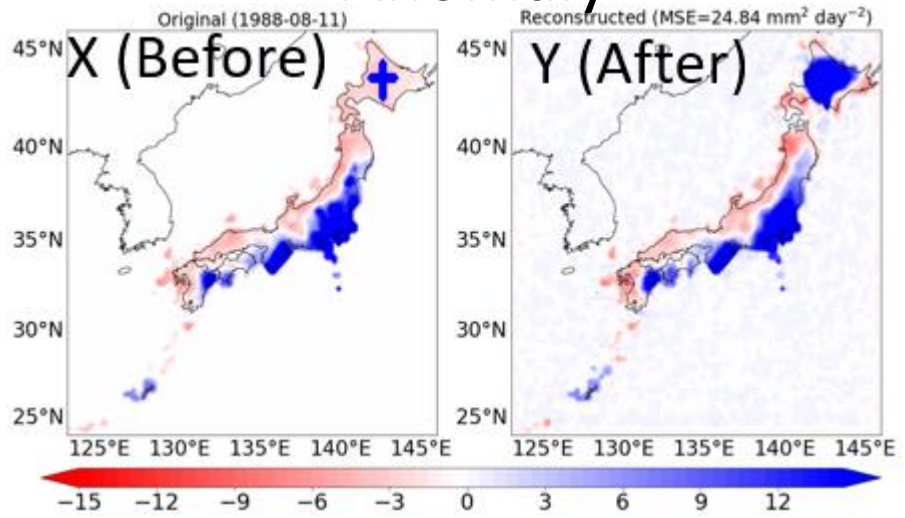

MSE greater than or equal to the 95 percentile are considered as outliers.

# Autoencoder applied to precipitation events

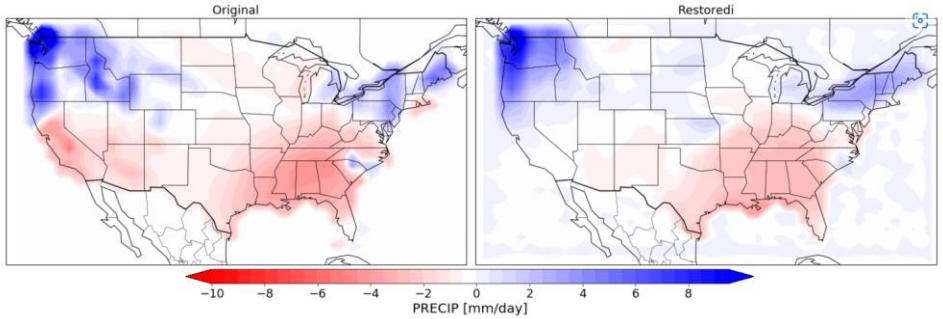
## Normal




## Anomaly

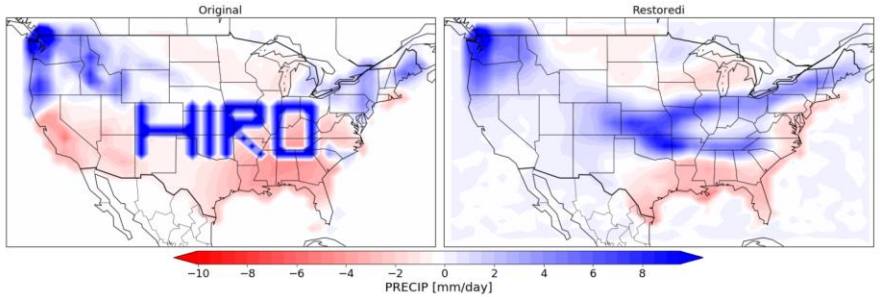



## Normal




X (Before) Y (After)

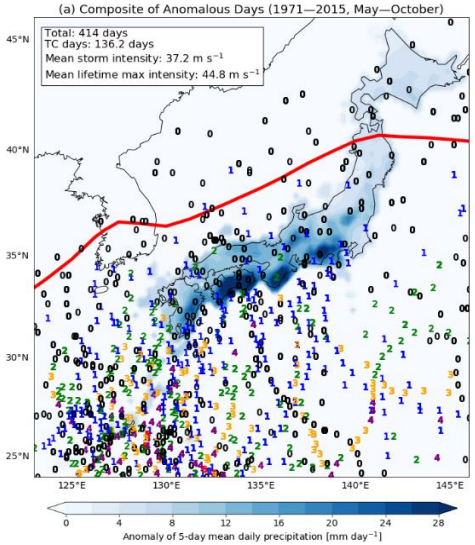
## Anomaly



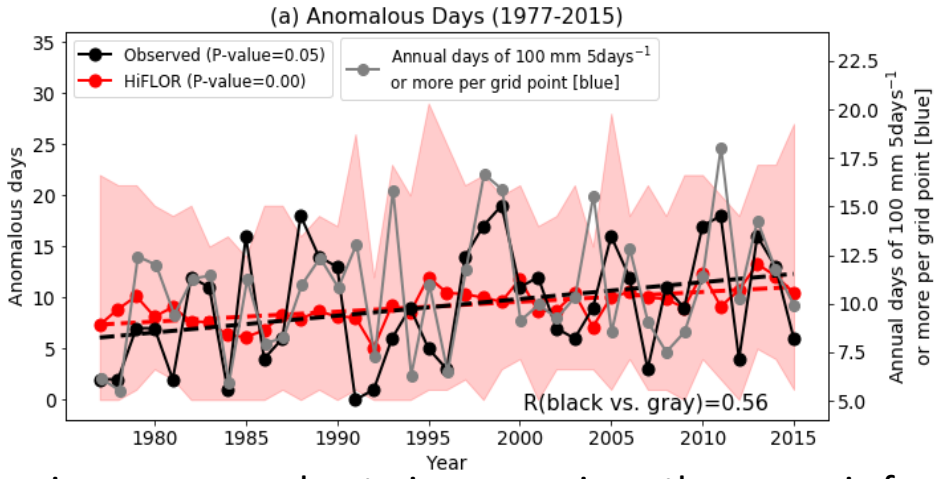

X (Before) Y (After)

# Results in Murakami et al. (2022)

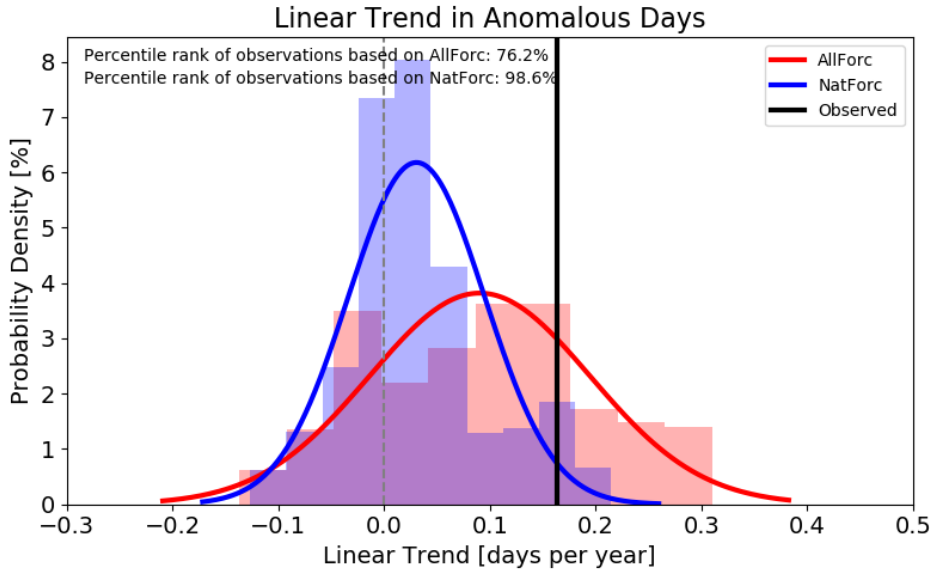
Detected anomalous precipitation events by the autoencoder are related to frontal structure and it accompanied with typhoons.




Frequency of extreme events increases in observations, present, and future projections.

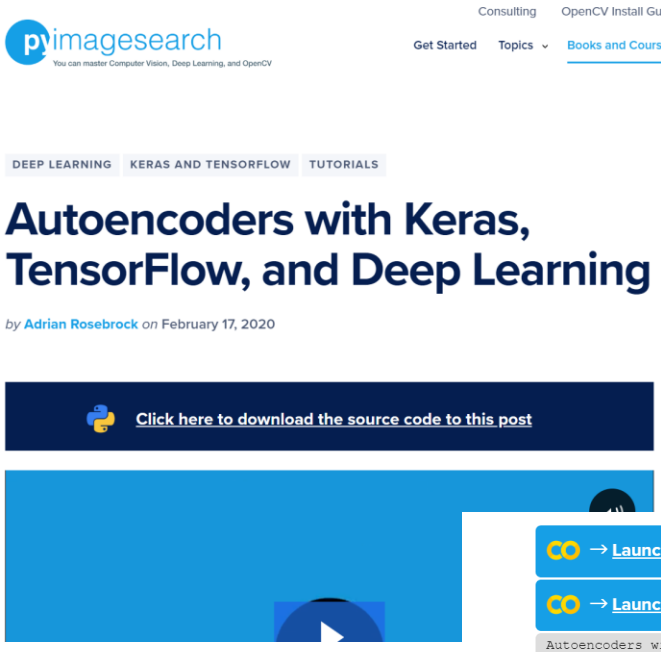



These increases are due to increases in anthropogenic forcing.

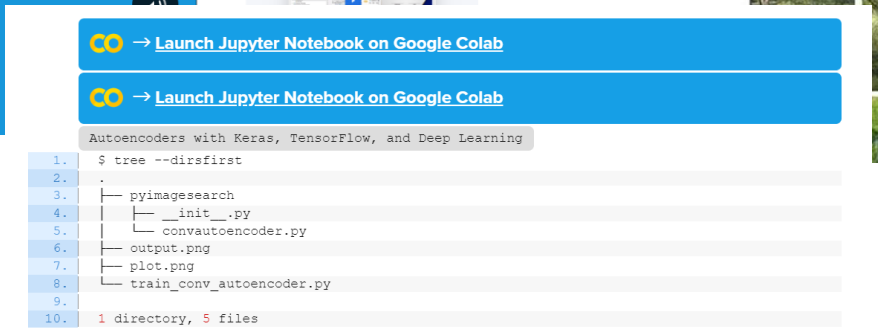


# pyimagesearch

- Intro to anomaly detection with OpenCV, Computer Vision, and scikit-learn
- Anomaly detection with Keras, TensorFlow, and Deep Learning
- Autoencoders with Keras, TensorFlow, and Deep Learning








We will review two Python scripts today:

- `convautoencoder.py`: Contains the `ConvAutoencoder` class and `build` method required to assemble our neural network with `tf.keras`.
- `train_conv_autoencoder.py`: Trains a digits autoencoder on the MNIST dataset. Once the autoencoder is trained, we'll loop over a number of output examples and write them to disk for later inspection.