Combining Physical and AI Models to Explore Climate Change

-- An ITCZ Detection Example

Research Report

Computer Science Department

Kean University

Student: Chen-Kai Tsai

Instructor: Juan J. Li, Ph.D

Date: November 24th, 2021

Introduction

What is ITCZ?

The Intertropical Convergence Zone (ITCZ) which appears as a band of clouds is an area where the northeast and the southeast trade winds converge. It encircles earth near the thermal equator but the exact location will vary every season and every year. The trade winds in the north hemisphere move south and in the south hemisphere they move toward north. Although the regain where the direct solar radiation is symmetric to the equator, the ITCZ is not symmetric relative to the equator. In the east Pacific Ocean, the ITCZ locates in the south hemisphere for the entire year. Sometimes a double ITCZ will form (Fig 1, Fig 2) with one located on north hemisphere and another on south hemisphere. When this occurs, a narrow area of high pressure appears between the two ITCZ structure.

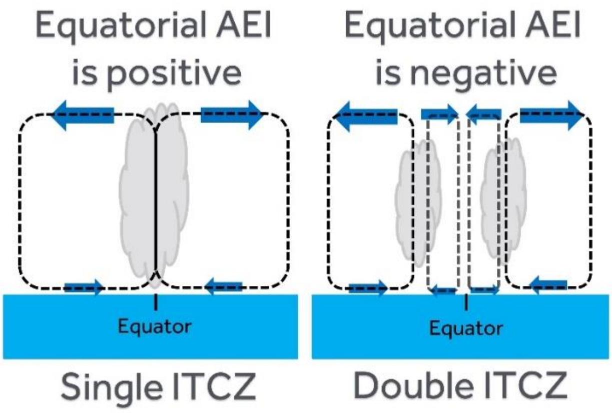


Fig 1. An illustration of the ITCZ and double ITCZ

Fig 2. An example of double ITCZ photo took in 3/10/2018

$$dI = \frac{1}{2}[H(5°\mathrm{N}) + H(5°\mathrm{S})] - [H(0°\mathrm{N}) + 1]$$

where

$$H = \begin{cases} 1 & (p \geq 100 \text{ mm}), \\ 0 & (70 < p < 100 \text{ mm}), \\ -1 & (p \leq 70 \text{ mm}), \end{cases}$$

Fig 3. Equation to detect ITCZ. If dI = 1, a well-defined ITCZ was present. If dI $\leq$ -1, when no double ITCZ present.

Effect on Weather

Global warming is causing more Inter-Tropical Convergence Zones (ITCZ) scenarios as shown in weather data, which makes manual detection infeasible. ITCZs play an important role in the global circulation system and even small changes in its patterns can cause severe droughts or flooding as well as other disasters including hurricanes. Thunderstorms along the ITCZ cause the loss of Air France Flight 447 in 2009. Therefore, detecting ITCZ and examine its activity is important for improve human's understanding of global climate and help human prepare of the upcoming weather change.

Motivation

The traditional method to detect a double ITCZ is based on the moisture profile retrieved from the satellite. In the ***Evolutionary Structure of the Eastern Pacific Double ITCZ Based on Satellite Moisture Profile Retrievals*** which publish on Journal of Climate, CHRISTOPHER E. LIETZKE propose mathematical equation (Fig 3) to detect a double ITCZ. Our research discovered that ITCZ detection based on physical model alone could misclassify some unexpected situations when the double bands occur at different places with similar intensity or at the same places with different intensities. Therefore, we want to design and train an AI model to detect the double ITCZ and compare the result from our model and the outcome from the physical model to find additional image that might contain double ITCZ but the physical model failed to detect it.

Dataset

We use the TRMM (The Tropical Rainfall Measuring Mission) data which was provided by the research satellite which provide 5 instrument Precipitation Radar (PR), TRMM Microwave Imager (TMI), Visible Infrared Scanner (VIRS), Clouds & Earths Radiant Energy System (CERES) and Lightning Imaging Sensor (LSI) operation from 1997 to 2015. The dataset is in NetCDF (Network Common Data Form) format which is hosted by UCAR (University Corporation for Atmospheric Research). NetCDF is a set of open standard software libraries which is commonly used in earth science and atmospheric sciences datasets. The data format is portable and support creating, accessing and sharing of array-oriented scientific data.

Plot NetCDF Data onto Map Image

In our research, we use image classification model to detect if there is a double ITCZ exist. Therefore, before we put the data into the model, we need to create image base on the NetCDF data. Figure 4 shows the result of transferring the NetCDF format data into an image.

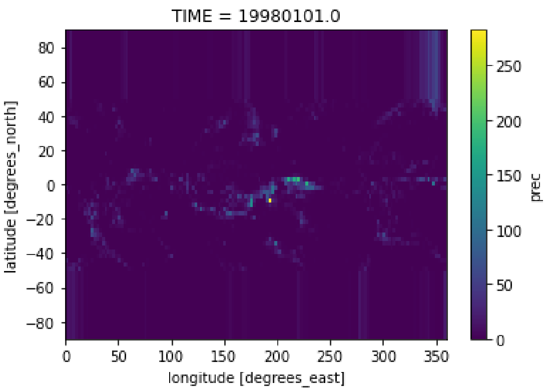
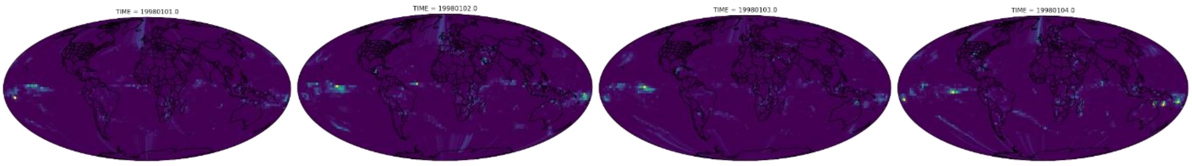


Fig 4 Transfer NetCDF format to image.

```
import cartopy.crs as ccrs
import cartopy.feature as cft

days = np.arange(0,5)#Set range of days
for i in days:
    plt.figure(figsize=(20,5))#Set figure size
    ax1 = plt.axes(projection=ccrs.Mollweide())#Set projection type
    ds2.PREC[i].plot(ax=ax1, transform=ccrs.PlateCarree(), add_colorbar=False)#Plot PREC at i
TIME
    #Add map features
    ax1.coastlines()
    ax1.add_feature(cft.BORDERS)
    ax1.add_feature(cft.STATES)
    #Save the figure
    plt.savefig(f'D:/Projects/mapstest/{i+1}.jpg')
    plt.pause(1)
```

Secondly, we need to combine the geospatial data to plot the NetCDF to a globe map image. We use a cartopy python library which is specifically designed to process geospatial data to process our dataset and product the image. The above code snippet shows how we use the library to generate the image (Fig 5).



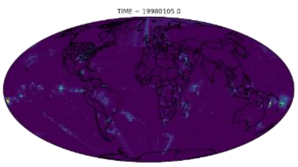
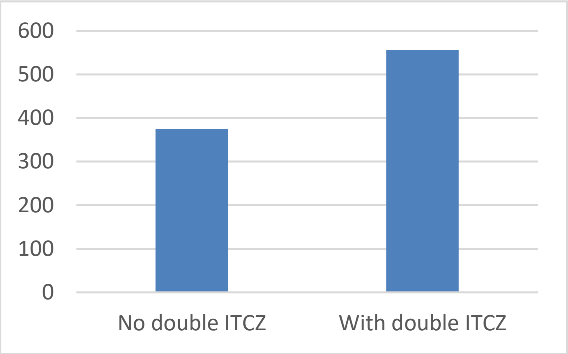
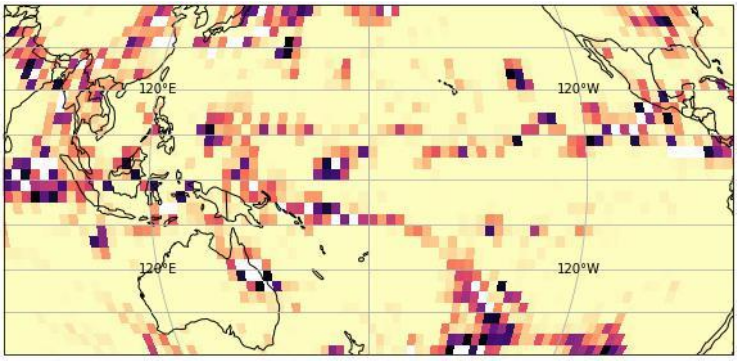


Fig 5 Results of the above code snippet which are data from 01/01/1998 to 05/01/1998

Summary of our dataset

The following diagram shows the number of images we have for containing double ITCZ and no double ITCZ. Additionally, the sample training image is provided.





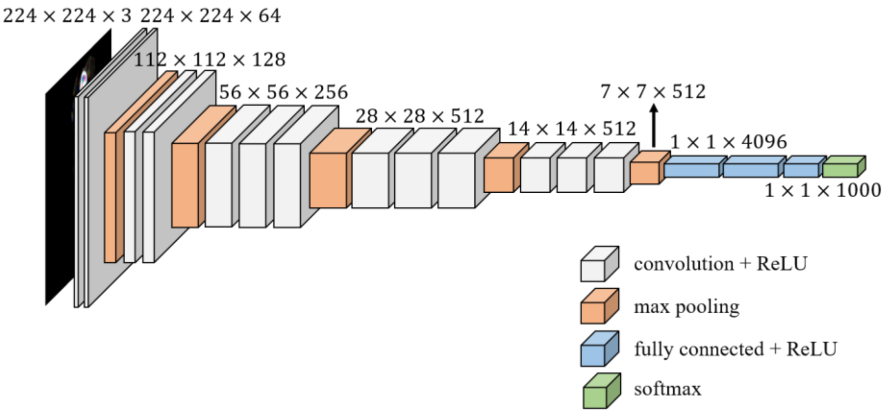
Solving with Deep Learning Models

TensorFlow 2.X Framework

Google's TensorFlow is popular open-source deep learning library and Keras is a high-level API with TensorFlow as its backbone. One of the important features of TensorFlow 2.X is that it fully incorporates Keras as its built-in high-level API. In tf.keras library, a model layer is similar to a piece of Lego. Therefore, models are easy to understand. Additionally, TensorFlow provide TensorFlow Hub which includes many pretrained model for many different tasks and further increase the productivity of the framework.

Transfer Learning with VGG16

In this research, we use transfer learning on VGG16 (Very Deep Convolutional Networks for Large-Scale Image Recognition) is presented to imagenet challenge by Karen Simonyan, Andrew Zisserman in 2014. The convolutional network is commonly used on computer vision or image processing and being used by VGG model as its main component. The main contribution of VGG is steadily increasing the depth of the model by using convolutional layer with size 3 * 3. Comparing to the AlexNet with 5 convolutional layers and kernel size start from 11 * 11, VGG16 has 13 convolutional layers with 3* 3 kernel size. The following picture is the architecture of the VGG16.



We can see that there are 5 blocks and, in each block, there are 2 to 3 convolutional layers. Between different blocks there is a max pooling layer. The image size will be reduced to half after passing through a max pooling layer.

Transfer Learning

In order to train a deep neural network, we need to have sufficient data. However, mostly this is not the case. Collecting data can be very expensive and, sometimes, impossible. Furthermore, deep neural networks contain many hyper parameters that require tuning. A change on the hyper parameters can affect the performance of the model. Last but not least, deep neural network often requires to train on high-end GPU or TPU which are expensive and, sometimes, impossible to obtain. With transfer learning, we can apply the pretrained networks and prebuilt models on our datasets. In our case, the training dataset is considerably small compare to imagenet. However, the first few convolutional layers are learning simple patterns such as lines and objects outline. The part that will have most differences is at the last few layers. Therefore, by using the weight from imagenet in the first few layers, we can harness the rich amount of knowledge learned from imagenet and apply that knowledge to our dataset.

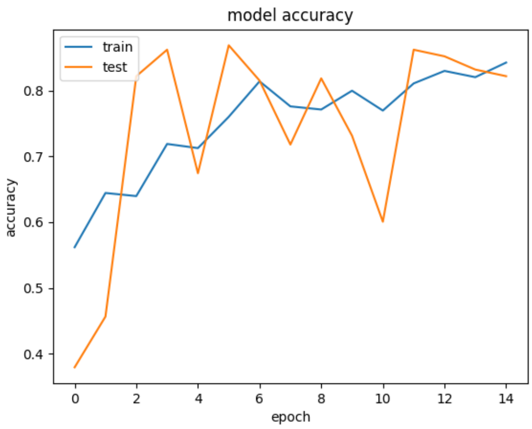
VGG16 training and tuning

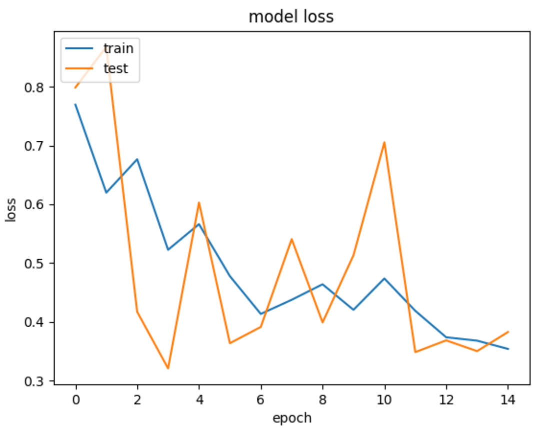
In this case, we start with freezing the top 3 blocks in the VGG16 model. Therefore, we only train on block4, 5, a flatten layer that we add. The original imagenet has 1000 classes, but we only need 2 classes in our research. By adding a dense layer with 2 units, we indicate that we only have 2 classes for our dataset. The following diagrams include the model summary, training accuracy and training loss.

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 575, 280, 3)]	0
block1_conv1 (Conv2D)	(None, 575, 280, 64)	1792
block1_conv2 (Conv2D)	(None, 575, 280, 64)	36928
block1_pool (MaxPooling2D)	(None, 287, 140, 64)	0
block2_conv1 (Conv2D)	(None, 287, 140, 128)	73856
block2_conv2 (Conv2D)	(None, 287, 140, 128)	147584
block2_pool (MaxPooling2D)	(None, 143, 70, 128)	0
block3_conv1 (Conv2D)	(None, 143, 70, 256)	295168
block3_conv2 (Conv2D)	(None, 143, 70, 256)	590080
block3_conv3 (Conv2D)	(None, 143, 70, 256)	590080
block3_pool (MaxPooling2D)	(None, 71, 35, 256)	0
block4_conv1 (Conv2D)	(None, 71, 35, 512)	1180160
block4_conv2 (Conv2D)	(None, 71, 35, 512)	2359808
block4_conv3 (Conv2D)	(None, 71, 35, 512)	2359808
block4_pool (MaxPooling2D)	(None, 35, 17, 512)	0
block5_conv1 (Conv2D)	(None, 35, 17, 512)	2359808
block5_conv2 (Conv2D)	(None, 35, 17, 512)	2359808
block5_conv3 (Conv2D)	(None, 35, 17, 512)	2359808
block5_pool (MaxPooling2D)	(None, 17, 8, 512)	0
flatten (Flatten)	(None, 69632)	0
dense (Dense)	(None, 2)	139266

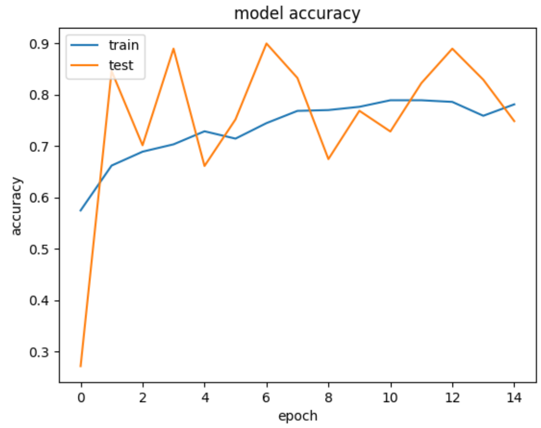
Total params: 14,853,954
Trainable params: 13,118,466
Non-trainable params: 1,735,488

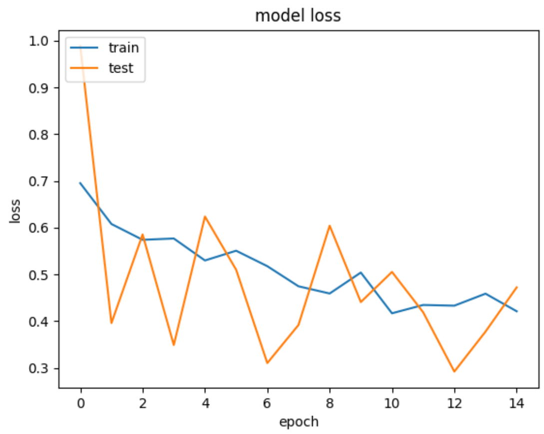




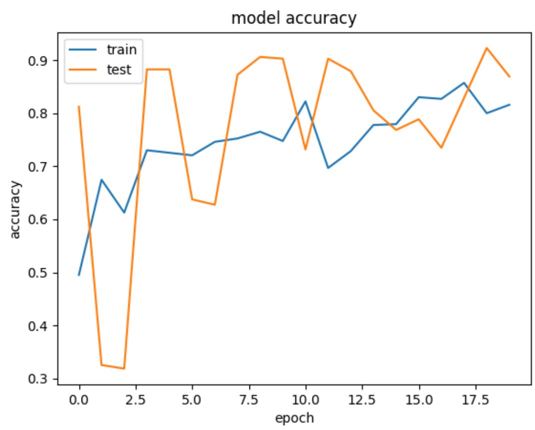
In the above diagram, we can find that the validation set is unstable and the training accuracy reaches 86%. The model can still be improved. One of the possible can be done is reduce the trainable parameters which still up to 88% of the total parameters in our model. Therefore, we freeze most of the model left with only one convolutional layer which in our model summary is named ‘*block5_conv3*’. The following diagrams include the model summary, training accuracy and training loss.

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 575, 280, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 575, 280, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 575, 280, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 287, 140, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 287, 140, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 287, 140, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 143, 70, 128)      0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 143, 70, 256)      295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 143, 70, 256)      590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 143, 70, 256)      590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 71, 35, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 71, 35, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 71, 35, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 71, 35, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 35, 17, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 35, 17, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 35, 17, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 35, 17, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 17, 8, 512)        0
_________________________________________________________________
flatten (Flatten)            (None, 69632)             0
_________________________________________________________________
dense (Dense)                (None, 2)                 139266
=================================================================
Total params: 14,853,954
Trainable params: 2,499,074
Non-trainable params: 12,354,880
_________________________________________________________________
```

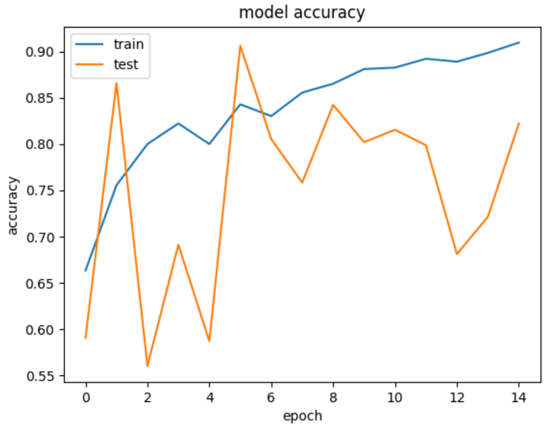


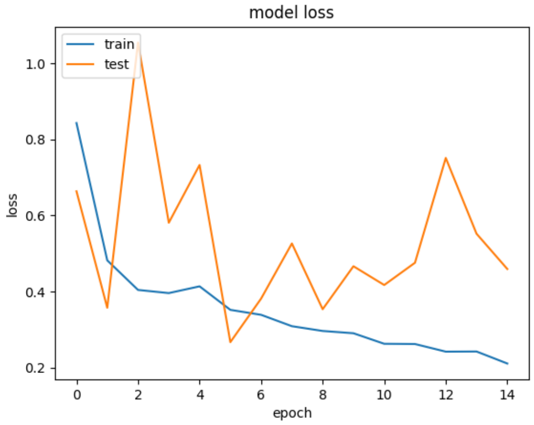


We can see that the model does not improve after we freeze more layers. However we can see that the model improve slower than the previous model. Therefore, we can try to train it with more epochs. The following result shows the model accuracy and loss after training for 20 epochs.

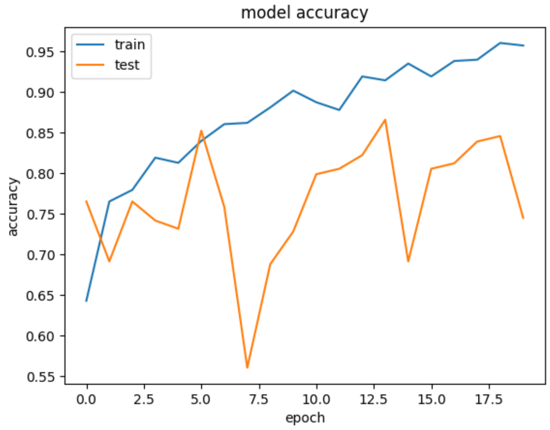


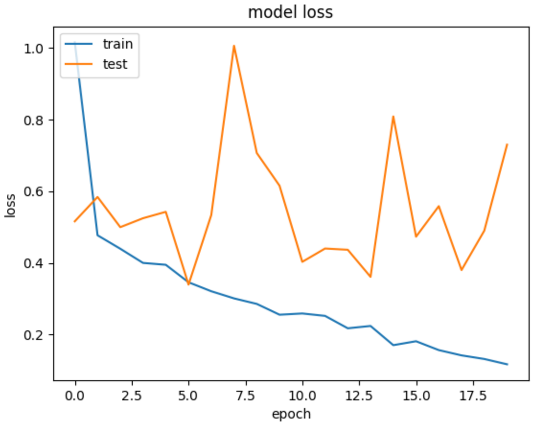
The model now reaches similar accuracy with considerable less trainable parameters. In the next experiment, we change the optimizer to Adam (Adaptive Momentum) which is currently the most widely-used optimizer. Compare to the SGD which is used for previous experiments, Adam calculate learning rate for every parameters instead of using one learning rate for all the parameters and add an additonal correction mechanism. The following diagram shows the result with 15 training epochs.





The training accuracy grow steadily and reaches 93%. However, the validation set accuracy become more unstable and perform worst than the previous experiment. We than trained the model with 20 epoch to see if this phonomenon still exist.

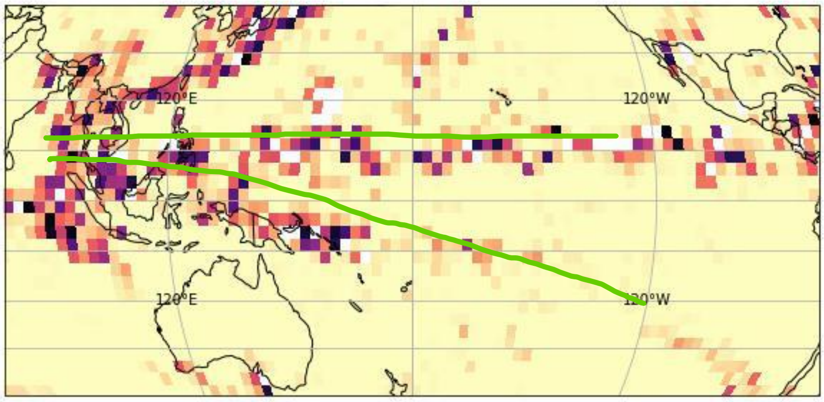




This time, we reaches 95% of training accuracy. However, the validation set accuracy is still unstable and overfitting become prominont.

Result Review

The following example is the additional image found by the AI model but not detected by the physical model. Our program separate the validation set to two different folders by calculate the difference between the two methods. One of the folder store image that is not detected by the physical model but detected by the AI model and the other store image that is not detected by the AI model but detected by the physical model.



Conclusion

Human detect and labeling the existance of double ITCZ for a TRMM data is time consuming or impossible to do. In this research, our goal is to find the images that contain double ITCZ that is not in the original training set or in the training set but not classified as having double ITCZ by the tranditional physical model method. The original dataset is labeled by a physical model which is not perfect. By dicovering additional images that might contain a double ITCZ, the expert can review only on the additional image found by AI model thus save time and resources.

Future Works

Require Ground Truth Labels: Although in some cases, we can clearly identify the result from the model and from the physical model is correct or not. We will still require an expert to help identified the ground truth label which will take excessive time to do.

Take Time Squence into Consideration: The original dataset contain a time series data. We can use this as an additional feature to help our model improve its accuracy. Recurrent Neural Network (RNN) which is specialized in learning from sequential data such as time series, video and text can be combined with CNN to improve our result.

Referen

1. *illustration of the ITCZ and double ITCZ (Figure 1).* https://thesocialmetwork.files.wordpress.com/2018/07/blog_figure_itcz_simulation.jpg
2. *Double ITCZ Picture (Figure 2).* https://twitter.com/dscovr_epic/status/972973085275164672?s=20
3. CHRISTOPHER E. LIETZKE, CLARA DESER, THOMAS H. VONDER HAAR. *Evolutionary Structure of the Eastern Pacific Double ITCZ Based on Satellite Moisture Profile Retrievals*. 18 June 1999
4. Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 4 Sep 2014. arXiv:1409.1556
5. TensorFlow https://www.tensorflow.org/
6. The Tropical Rainfall Measuring Mission (TRMM) https://gpm.nasa.gov/missions/trmm
7. NetCDF https://www.unidata.ucar.edu/software/netcdf/
8. Stevo Bozinovski. *REMINDER OF THE FIRST PAPER ON TRANSFER LEARNING IN NEURAL NETWORKS, 1976*. June 10, 2019