# Day After Tomorrow

*Release 0.1*

**Team Hugo**

**May 27, 2022**

# CONTENTS:

CHAPTER

ONE

# HURRICANEFORECAST

## 1.1 HurricaneForecast package

### 1.1.1 Submodules

### 1.1.2 HurricaneForecast.Model module

### 1.1.3 HurricaneForecast.StormData module

class HurricaneForecast.StormData.**DataChecks**(*download_dir*, *source*, *labels=None*, *provide_df=False*, *df=None*, *storm_id='*'*)

Bases: object

**average_wind_speeds**()

Returns average wind speed over dataframe

**check_shapes**(*verbose=False*)

Check size of images in df. Raise warning if shape is unequal.

**check_time_gaps**(*verbose=False*)

Check time gap between images. Raise warning if the gap is larger than 1 hour.

**count_images**()

Returns number of images in dataframe

**get_anomalous_shapes**()

Returns lists storing image id of anomalous images and shapes of those images

**get_image**(*image_id*)

A function that returns the image represented by image_id as a tensor.

**get_time_gaps**()

Return lists storing index of large time gap and the magnitude of the time gap

**get_warnings**()

Print out warning if image sizes are unequal or large time gap is found.

**plot_wind_speeds**()

Create plot of wind speed over relative time

### 1.1.4 HurricaneForecast.StormTensor module

class HurricaneForecast.StormTensor.**StormTensorDataset**(*train_df*, *storm_id*, *num_sequence*, *download_dir*, *train_source*)

Bases: torch.utils.data.dataset.TensorDataset

**Resize_space**()

Resize all the images in the train_df by (366, 366).

**calculate_mean_std**(*storm_id*)

Calculate the mean and standard deeviation of all the pixels in the training dataset This will be used for normalisation

**get_image_tensor**(*image_id*)

Obtain image tensor from Image ID.

**image_id** [str] Image ID from self.df

**img** [torch.tensor] Tensor of the image specified by image_id

**get_last_10_img**()

Get the last tem 10 imgs to predict the last five

**get_last_five**()

**get_storm_images**(*storm_id*)

Create a tensor of all images in the data for storm with Storm ID = storm_id

**get_tensor_combination**(*start_pos*, *number*)

Obtain the combination tensor of several images from the given start point and number of images required.

**start_pos** [int] The start index to combine the images.

**number: int** Combined the number of images in one tensor

**img** [torch.tensor] The combination tensor for numbers of images.

**tensor_to_image**(*tensor*)

Convert image tensor to a 2D array

**tensor** [torch.tensor] Image tensor with one channel

**tensors: Tuple[torch.Tensor, ...]**

### 1.1.5 HurricaneForecast.Train_Validate module

### 1.1.6 Module contents

CHAPTER

TWO

# INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

## h

# INDEX