

```
In [6]: import pandas as pd # used for extracting various csv files
import numpy as np #used for various mathematical operations
import matplotlib.pyplot as plt # used for plating various graphs
%matplotlib inline
import matplotlib.image as mpimg
import seaborn as sns #used for making various statistical graphs
from tensorflow import keras # used for evaluating various Deep learning models
import tensorflow as tf #TensorFlow is an open-source library developed by Google primarily for deep learning applications
from tensorflow.keras import Model, Input
from tensorflow.keras.layers import Dense, Flatten #Flatten layers are used when you got a multidimensional output and you want to make it linear to pass it o

from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator #ImageDataGenerator class allows you to randomly rotate images through any degree between
from tensorflow.keras.callbacks import LearningRateScheduler, ReduceLROnPlateau

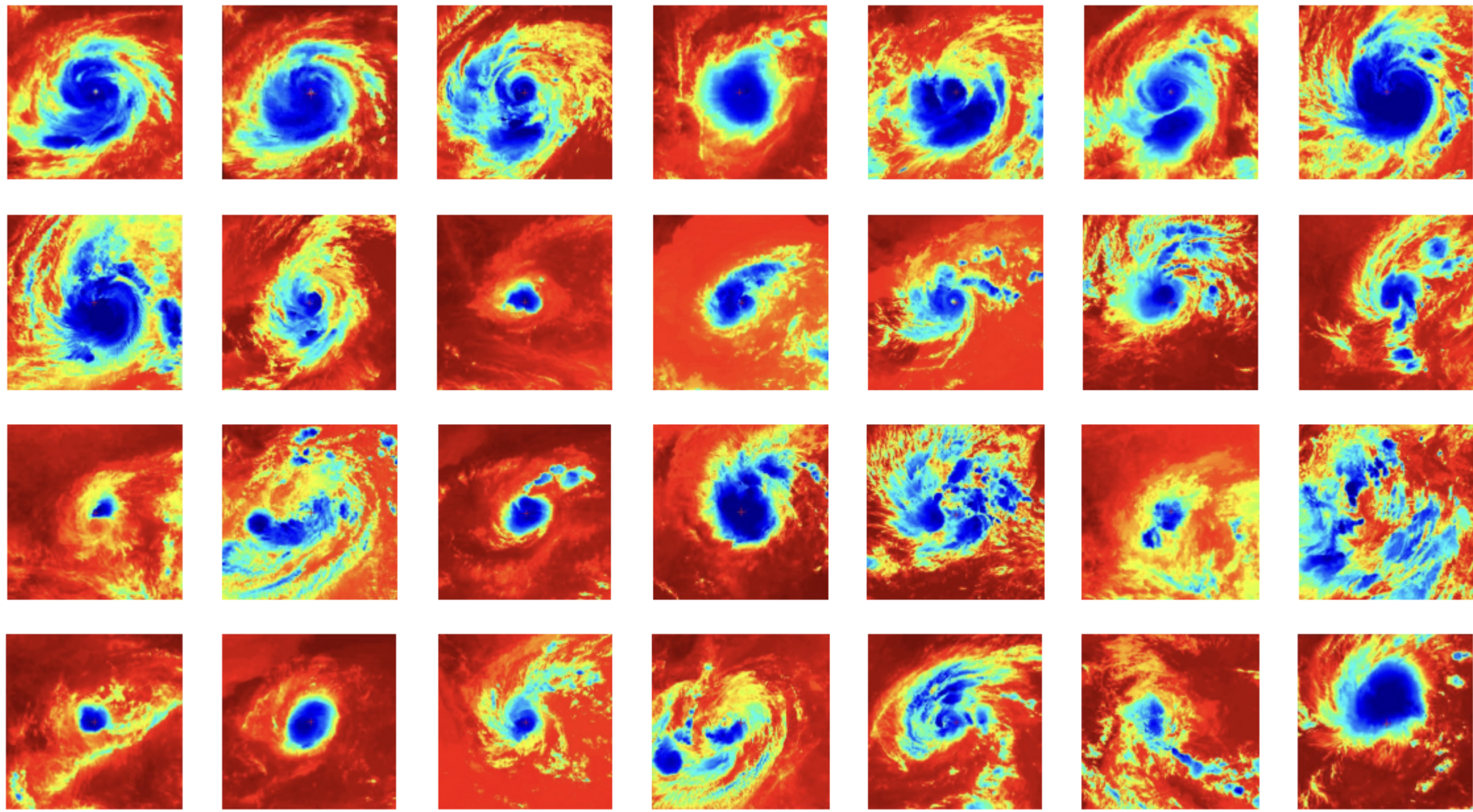
from tensorflow.keras import layers
from tensorflow.keras.applications import MobileNetV2, EfficientNetB0, EfficientNetB4, Xception
import warnings
import glob # It is global library
warnings.filterwarnings("ignore")
```

```
In [7]: data=pd.read_csv('insat_3d_ds - Sheet.csv') # The start data of csv file is read by this instruction
data.head()
```

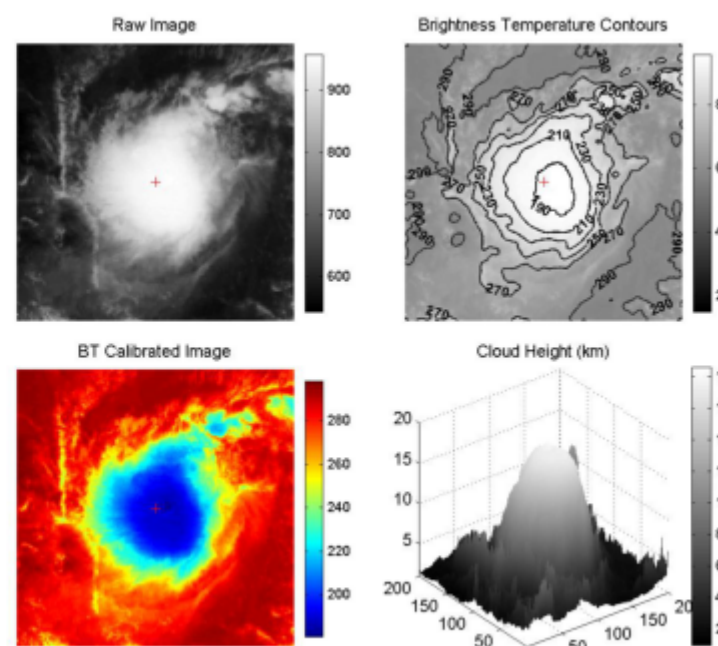
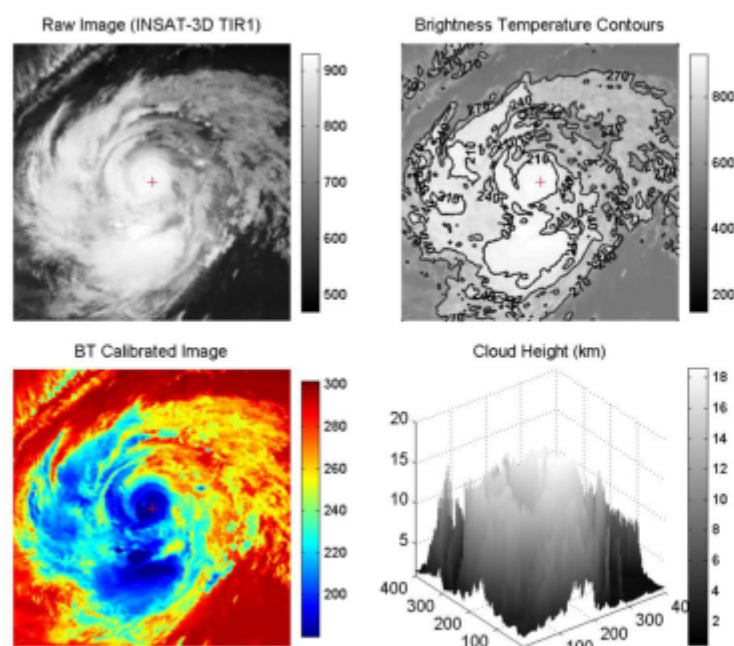
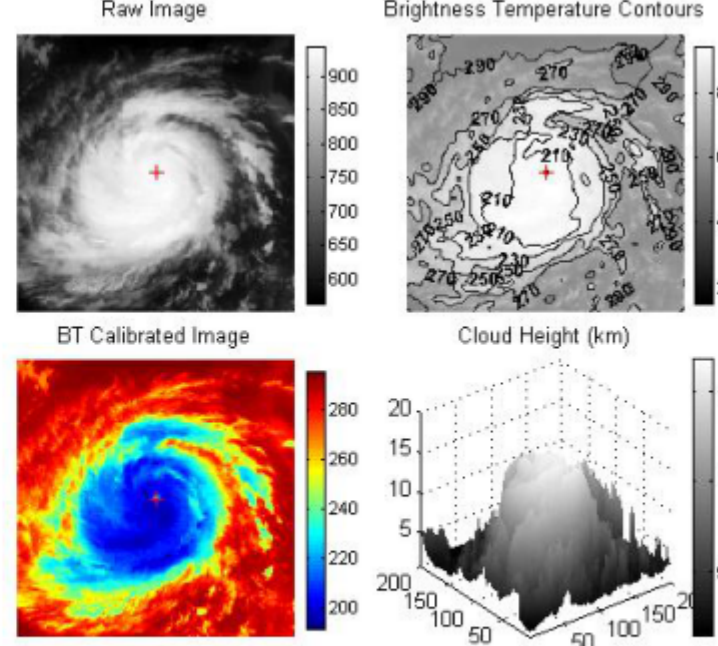
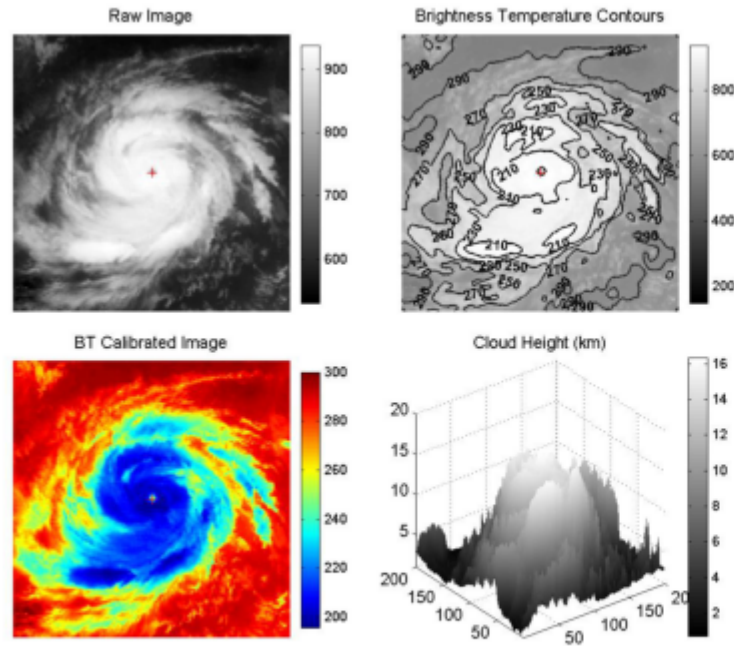
Out[7]:

	img_name	label
0	25.jpg	25
1	27.jpg	27
2	28.jpg	28
3	30.jpg	30
4	30(1).jpg	30

```
In [16]: paths=glob.glob('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED/*.jpg') #for extracting the dataset
plt.figure(figsize=(20, 20))
for i in range(28):
    cur_img = mpimg.imread(paths[i]) #used for reading images from a folder using matplotlib
    ax = plt.subplot(7, 7, i + 1) # number of images that can be seen in the output
    plt.imshow(cur_img.astype("uint8")) # unit 8 is a array which can store 8 bit numbers
    plt.axis("off")
```



```
In [9]: paths=glob.glob('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET/*.jpeg')
plt.figure(figsize=(20, 14))
for i in range(4):
    cur_img = mpimg.imread(paths[i]) # same as above code only difference is of data
    ax = plt.subplot(2, 2, i + 1)
    plt.imshow(cur_img.astype("uint8"))
    plt.axis("off")
```



```
In [10]: train_datagen = ImageDataGenerator(rescale=1.0/255.0) #Original datatype of uint8 stores 8 bit nos (0-255) but they are too high for training model so to
```

```
In [19]: import pandas as pd
from keras.preprocessing.image import ImageDataGenerator
train=pd.read_csv('C:/Users/Tanmay Shinde/Desktop/Major Project/insat_3d_ds - Sheet.csv')
train_data=train_datagen.flow_from_dataframe(train,directory='../input/insat3d-infrared-raw-cyclone-images-20132021/insat3d_ir_cyclone_ds/CYCLONE_DATASET_INFRA
y_col='label',subset='training',
target_size=(512,512),batch_size=16,shuffle=True,class_mode='raw') # used to extract data as the taining datas
```

Found 0 validated image filenames.

```
In [12]: tf.keras.preprocessing.image_dataset_from_directory('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED',
labels="inferred",
label_mode="int",
class_names=None,
color_mode="rgb",
batch_size=32,
image_size=(256, 256),
shuffle=True,
seed=None,
validation_split=None,
subset=None,
interpolation="bilinear",
follow_links=False,
crop_to_aspect_ratio=False
)
```

Found 0 files belonging to 0 classes.

```
ValueError                                Traceback (most recent call last)
C:\Users\TANMAY-1\AppData\Local\Temp\ipykernel_17016\4132589452.py in <module>
----> 1 tf.keras.preprocessing.image_dataset_from_directory('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED',
      2 labels="inferred",
      3 label_mode="int",
      4 class_names=None,
      5 color_mode="rgb",

~\Anaconda\lib\site-packages\keras\utils\image_dataset.py in image_dataset_from_directory(directory, labels, label_mode, class_names, color_mode, batch_size,
image_size, shuffle, seed, validation_split, subset, interpolation, follow_links, crop_to_aspect_ratio, **kwargs)
    207 image_paths, labels, validation_split, subset)
--> 209     if not image_paths:
    210         raise ValueError(f'No images found in directory {directory}. '
    211                        f'Allowed formats: {ALLOWLIST_FORMATS}')

ValueError: No images found in directory C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED. Allowed formats: ('.bmp', '.gif', '.jpeg', '.j
pg', '.png')
```

```
In [17]: from keras.preprocessing.image import ImageDataGenerator
train_data = train_datagen.flow_from_directory('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED/*.jpg',subset="training",x_col="img_name")

-----
TypeError                                Traceback (most recent call last)
C:\Users\TANMAY-1\AppData\Local\Temp\ipykernel_17016\673283414.py in <module>
      1 from keras.preprocessing.image import ImageDataGenerator
----> 2 train_data = train_datagen.flow_from_directory('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED/*.jpg',subset="training",x_col="i
mg_name",y_col="label",target_size=(512, 512),batch_size=16,class_mode='raw')
      3

TypeError: flow_from_directory() got an unexpected keyword argument 'x_col'
```

```
In [20]: from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2, #it is used to show a image from different angles
    zoom_range=0.2, # by this we can zoom on the images automatically
    horizontal_flip=True) #it flips rows and columns horizontally

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory('C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED',target_size=(512, 512),batch_size=
```

Found 0 images belonging to 0 classes.

```
In [21]: for image_batch, labels_batch in train_generator: # this gives number of images belonging to various classes
print(image_batch.shape)
print(labels_batch.shape)
break
```

(0, 512, 512, 3)
(0,)

```
In [22]: train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
    'C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED',
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary')
validation_generator = test_datagen.flow_from_directory(
    'C:/Users/Tanmay Shinde/Desktop/Major Project/CYCLONE_DATASET_INFRARED',
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary')
```

Found 0 images belonging to 0 classes.
Found 0 images belonging to 0 classes.

```
In [ ]: def build_model():
base = tf.keras.applications.Xception(weights="imagenet", include_top=False, input_tensor=Input(shape=(512, 512, 3)))

base.trainable = False

flatten = base.output
flatten = Flatten()(flatten)

bboxHead = Dense(64, activation="relu")(flatten)
bboxHead = Dense(32, activation="relu")(bboxHead)
bboxHead = Dense(1, activation="linear")(bboxHead)

model = Model(inputs=base.input, outputs=bboxHead)
return model
```

```
In [ ]: model = build_model()
```

```
In [ ]: model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss='mae', metrics=[tf.keras.metrics.RootMeanSquaredError()])

save_best = tf.keras.callbacks.ModelCheckpoint("Model.h5", monitor='loss',save_best_only=True, verbose=1)
```

```
In [ ]: from tensorflow_core.python.keras.utils.data_utils import Sequence
model.fit(train_datagenerator, epochs=50, callbacks=[save_best])
```

```
In [ ]:
```