CLOUD CLASSIFICATION FOR WEATHER PREDICTION

Using deep learning

TABLE OF CONTENTS

01

AGENDA

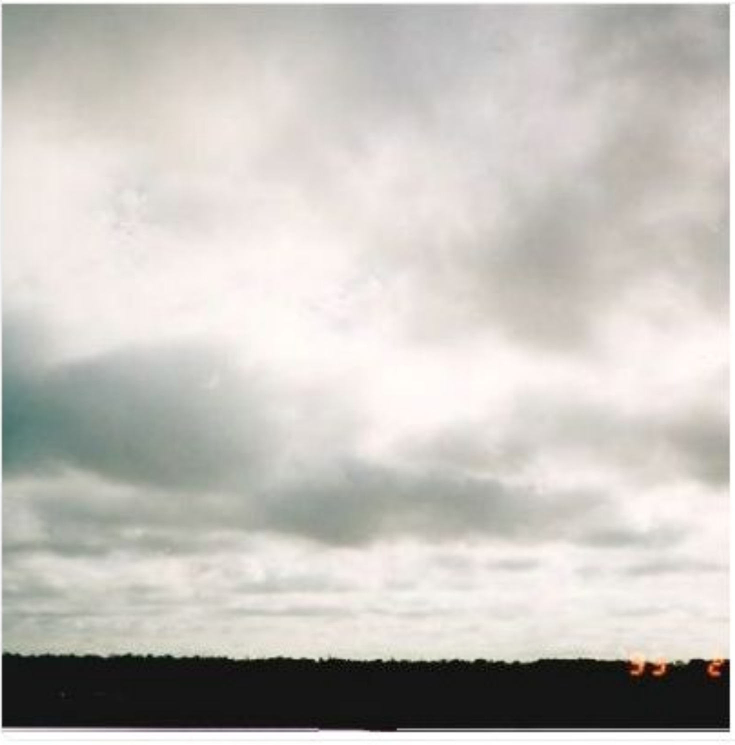
Weather condition is an important factor that is considered for various decisions.

Deep learning and computer vision can assist in classifying cloud images that do not depend on weather forecast information from the internet

This project aims to classify cloud images using deep learning to perform cloud image classification that is used predicting the weather condition

02

OBJECTIVE

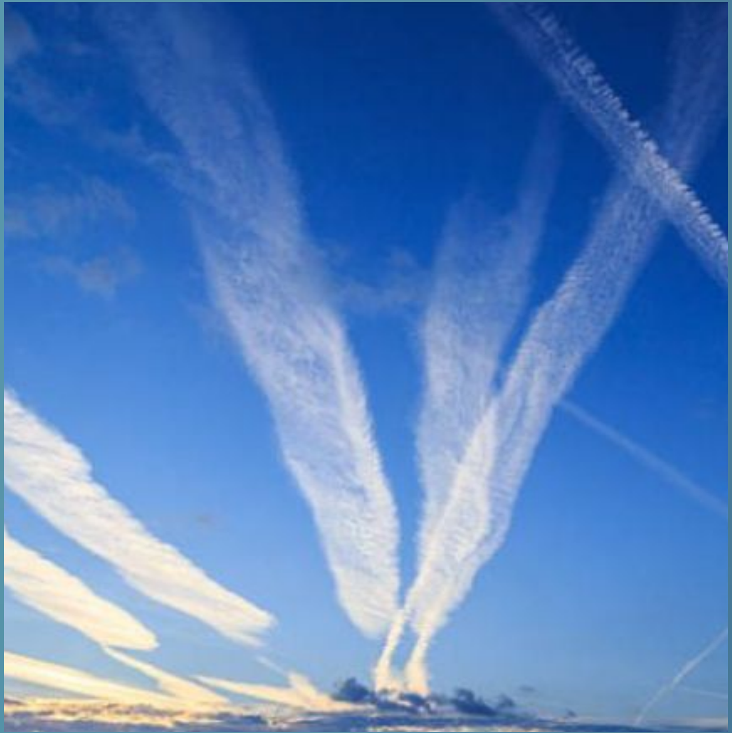


Stratus

Stratus is usually accompanied by little to moderate rainfall

Contrail

Indicates a fair and nice weather ahead



The modern classification scheme classifies clouds according to the altitude of cloud base, there being three altitude classes: low; mid-level and high. Within each altitude class additional classifications are defined based on four basic types and combinations there of.

Weather prediction from the type of cloud can predict upto immediate future and they also indicate change in weather pattern within 24 hours.

To build a multi class classification model for predicting weather from still image provided by the user which has crucial real life application such as environmental monitoring,distinguishing weather phenomenon that can improve local agriculture

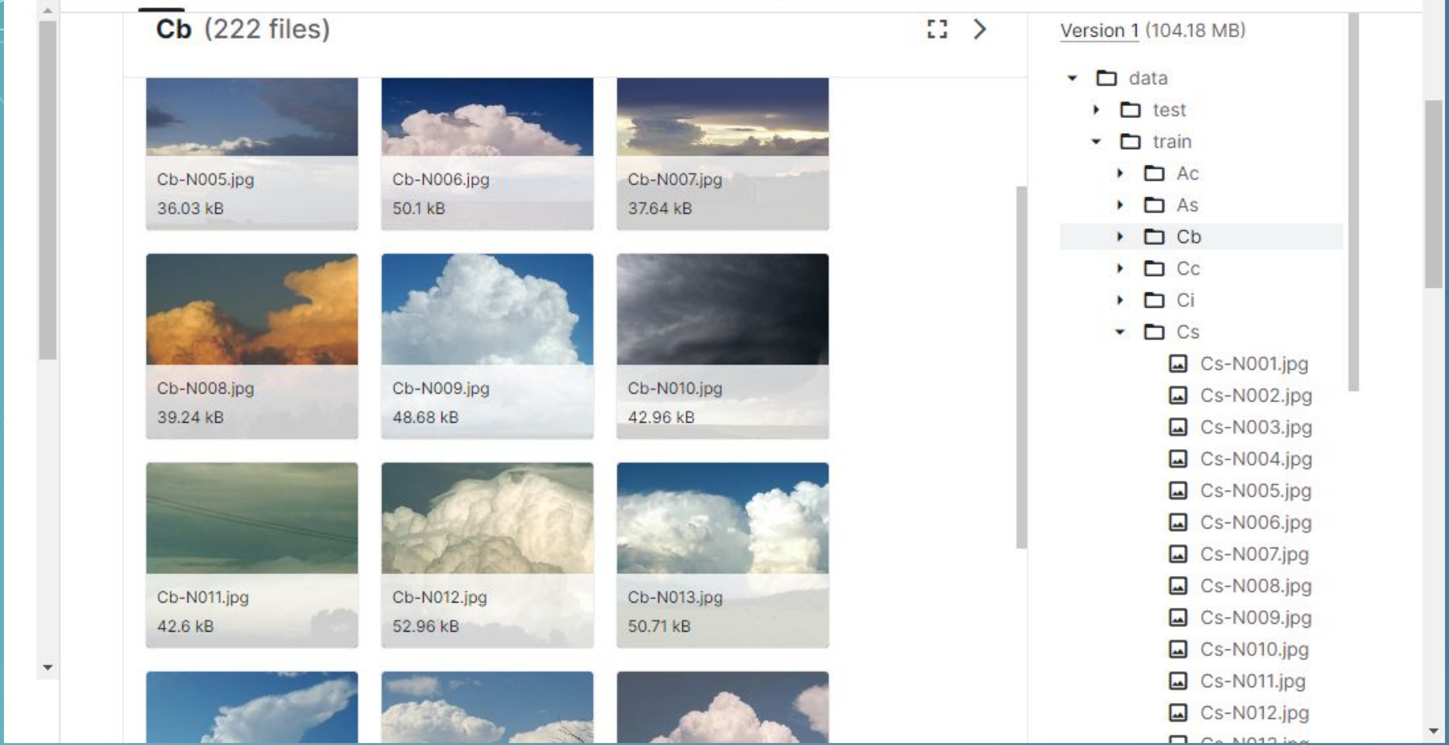
03

METHODOLOGY

Dataset Overview

The dataset has features 11 different classes of clouds collected from the Harvard data source.

The dataset contains about 2543 cloud labelled images. Images are of fixed dimensions and the photos are of same sizes 256x256 pixels.Each image has only one cloud category and are saved in separate folder as of the labelled class. Such as Ac, Sc,Ns,Cu,Ci,Cc,Cb,Co etc.



Cb (222 files)

Cb-N005.jpg
36.03 kB

Cb-N006.jpg
50.1 kB

Cb-N007.jpg
37.64 kB

Cb-N008.jpg
39.24 kB

Cb-N009.jpg
48.68 kB

Cb-N010.jpg
42.96 kB

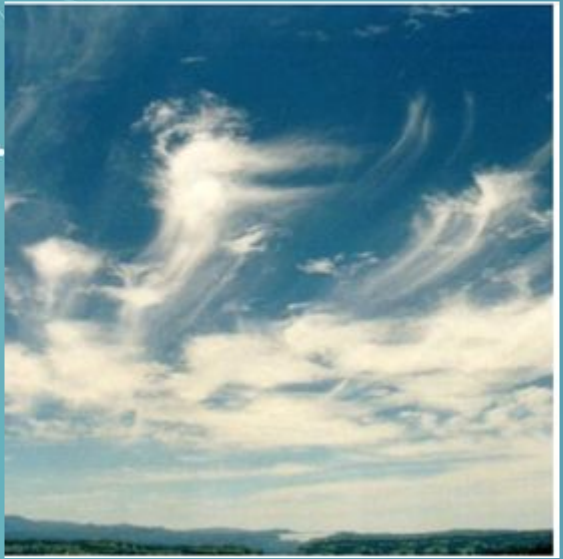
Cb-N011.jpg
42.6 kB

Cb-N012.jpg
52.96 kB

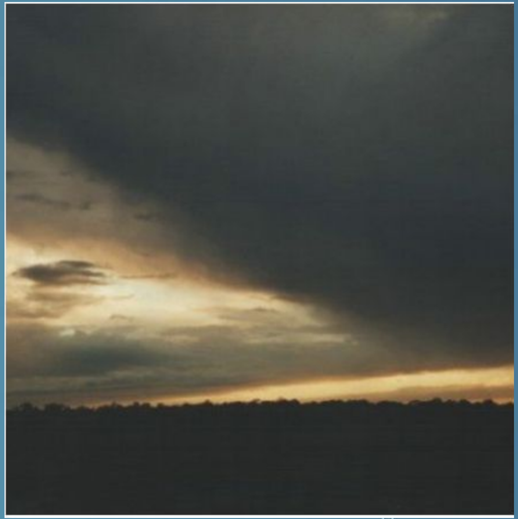
Cb-N013.jpg
50.71 kB

Version 1 (104.18 MB)

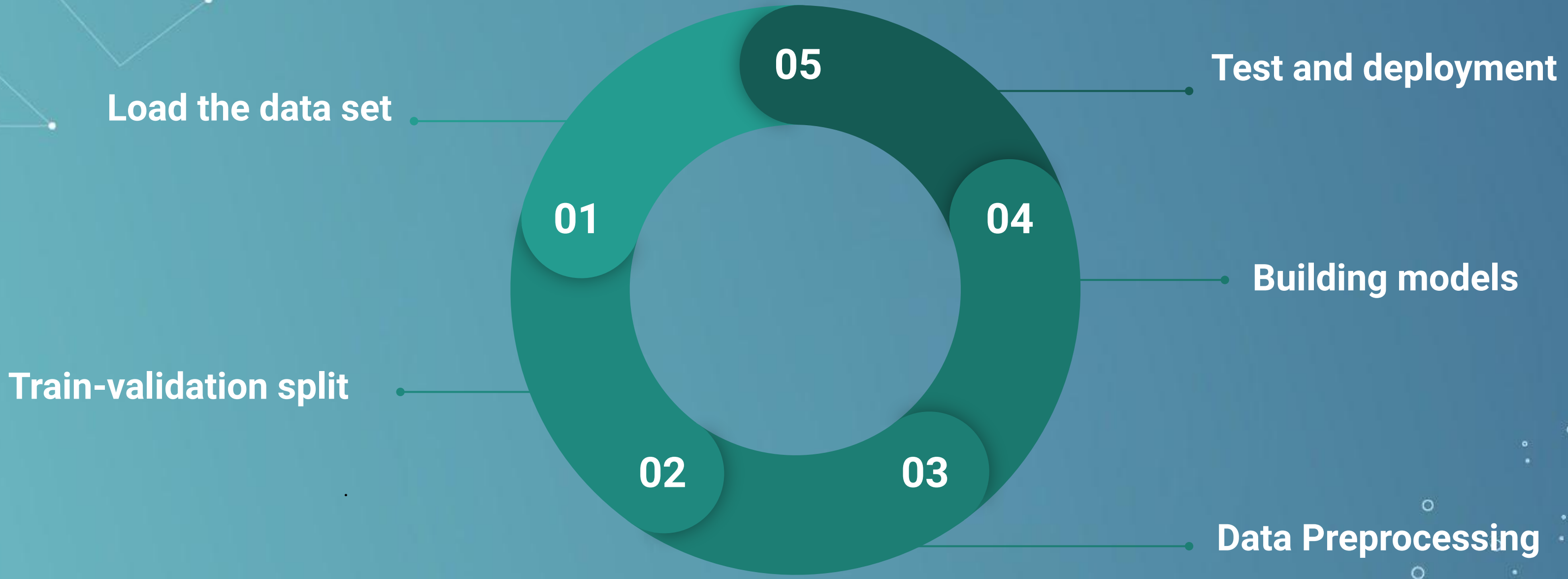
- data
  - test
  - train
    - Ac
    - As
    - Cb
    - Cc
    - Ci
    - Cs
      - Cs-N001.jpg
      - Cs-N002.jpg
      - Cs-N003.jpg
      - Cs-N004.jpg
      - Cs-N005.jpg
      - Cs-N006.jpg
      - Cs-N007.jpg
      - Cs-N008.jpg
      - Cs-N009.jpg
      - Cs-N010.jpg
      - Cs-N011.jpg
      - Cs-N012.jpg



Cirrus



Nimbostratus



01 Load the data set

02 Train-validation split

03 Data Preprocessing

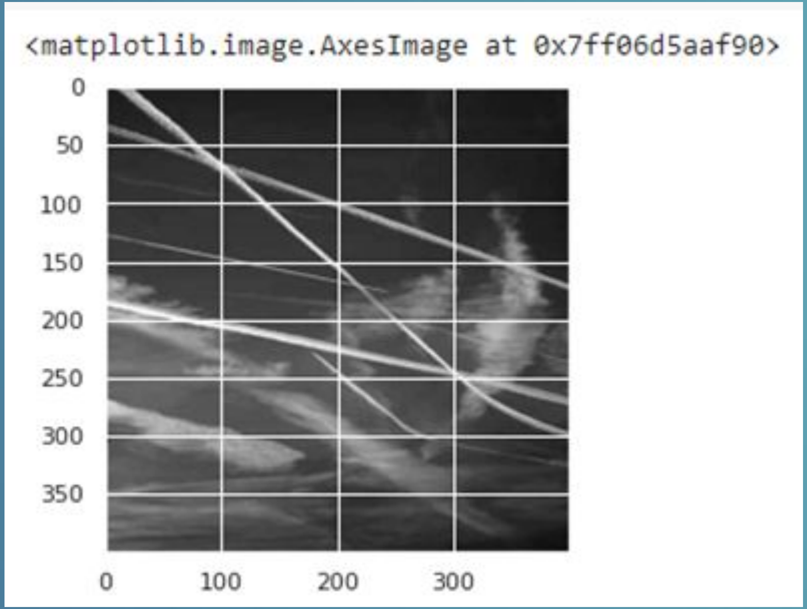
04 Building models

05 Test and deployment

Data Preprocessing

- Grayscale conversion
- Rotation
- Flipping
- Scaling and Standardizing images

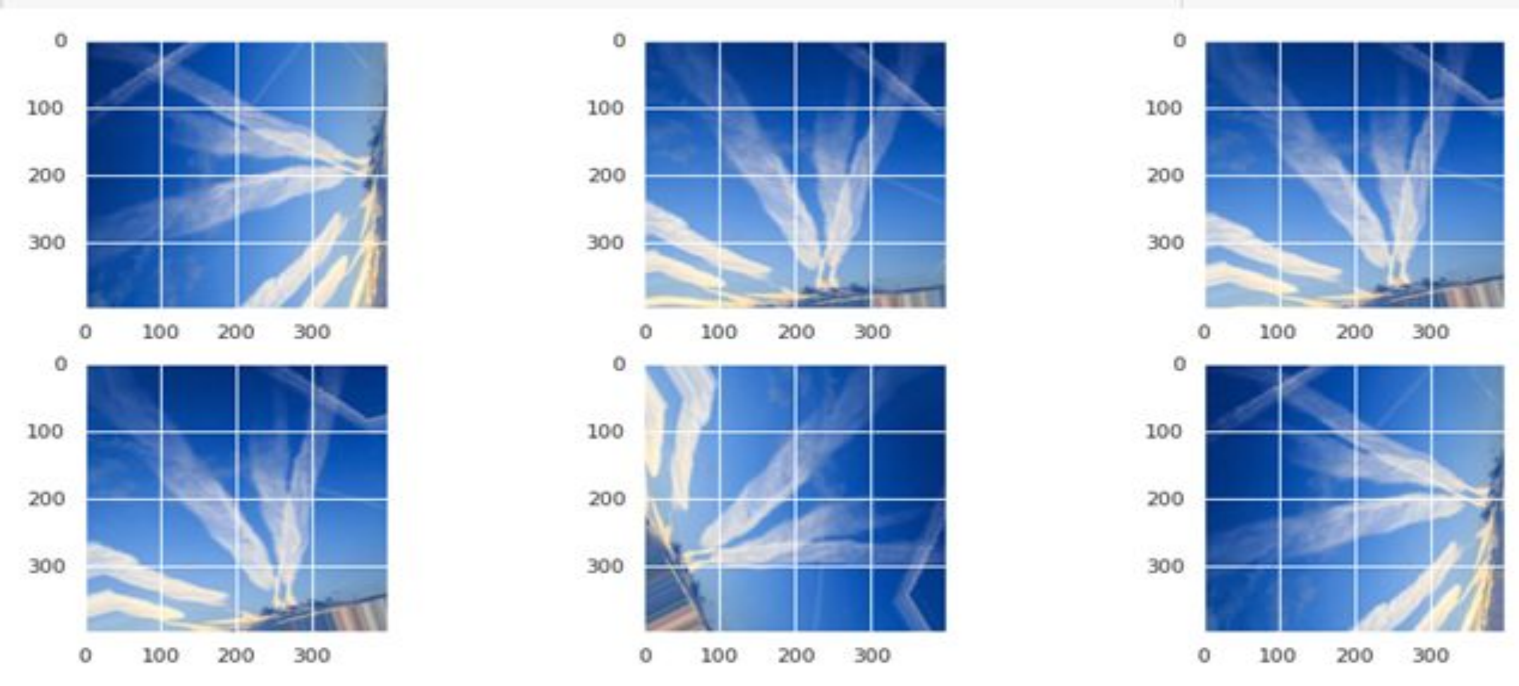
Grayscale conversion



```
<matplotlib.image.AxesImage at 0x7ff06d5aaf90>
```

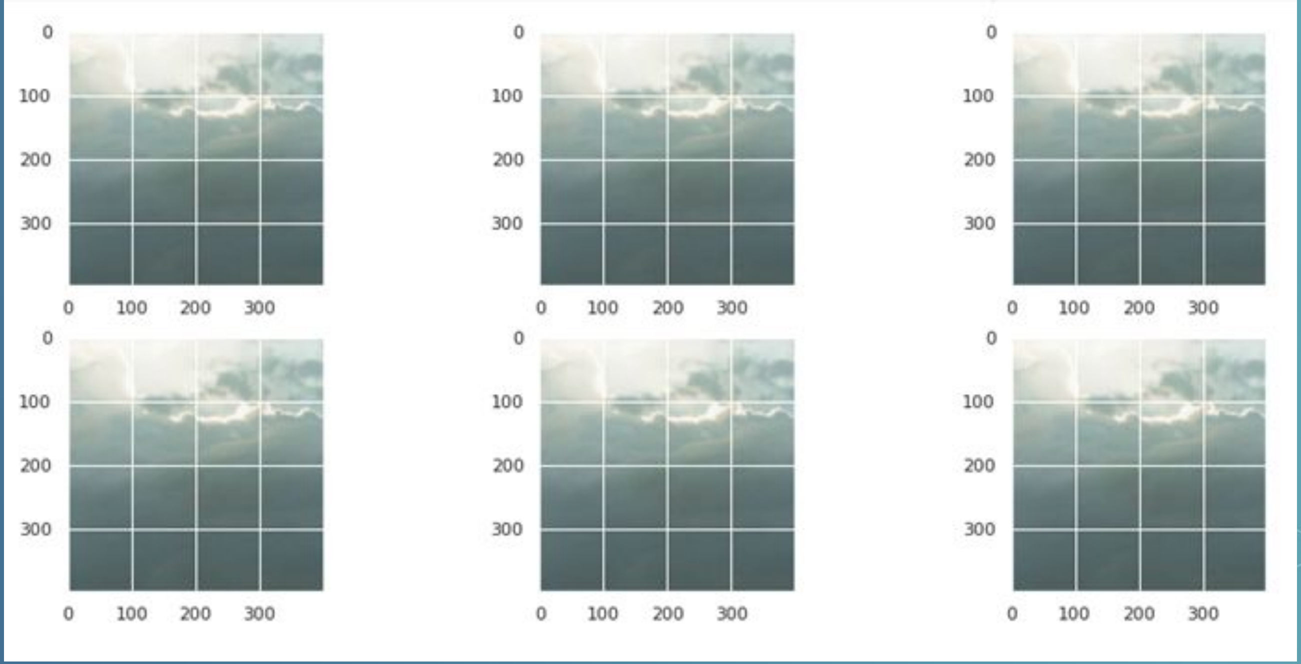
Simplifies the algorithm and reduces computational requirements

ROTATION



Rotates the image to a specific degree

FLIPPING



Reversing the entire rows and columns of an image pixels

Scaling and Standardizing images

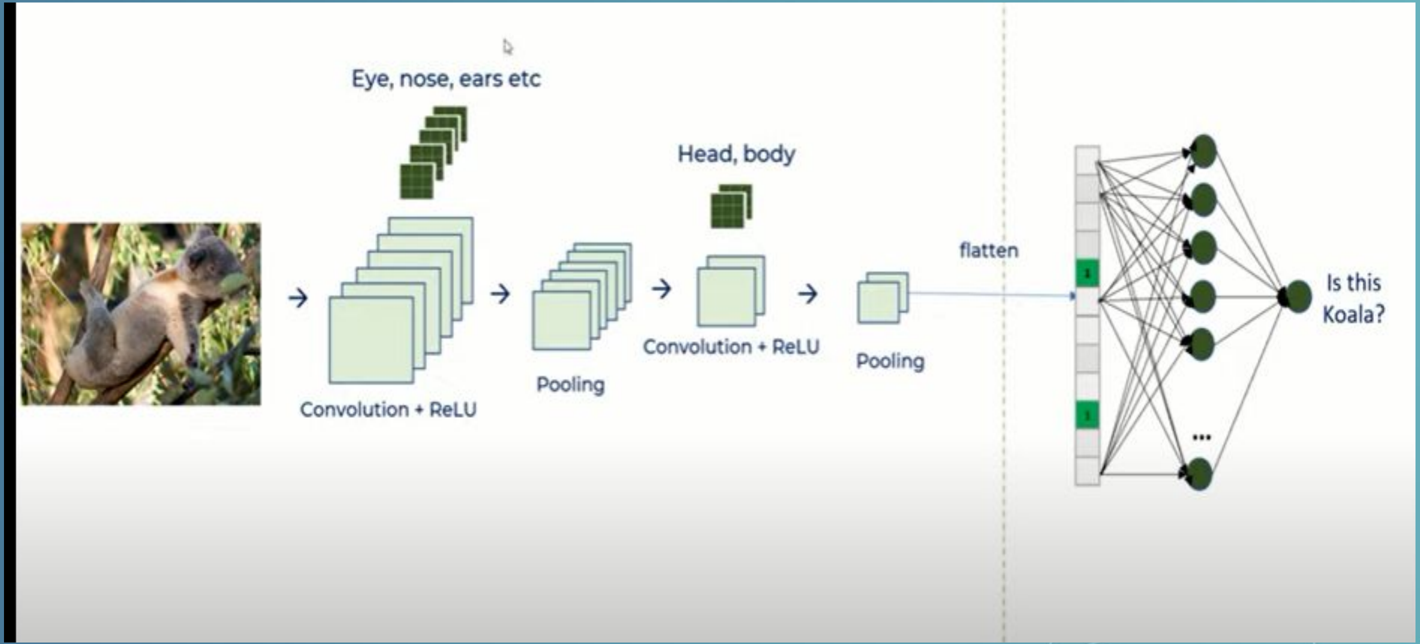
- **Standardization** is a method that scales and preprocesses images to have similar heights and widths. It re-scales data to have a standard deviation of 1 and a mean of 0
- Data scaling involves projecting image pixels into a predefined range

Building a CNN model

A convolution extracts tiles of input feature map and apply filters to then to compute new features , producing an output feature map.

During training the CNN learns the optimal values for filter matrices that enable it to extract meaningful features from input feature

With the help of CNN, We can build a model that classifies the given still image of a cloud to its correct type and its associated weather pattern



CNN Architecture

The CNN model is built with these layers

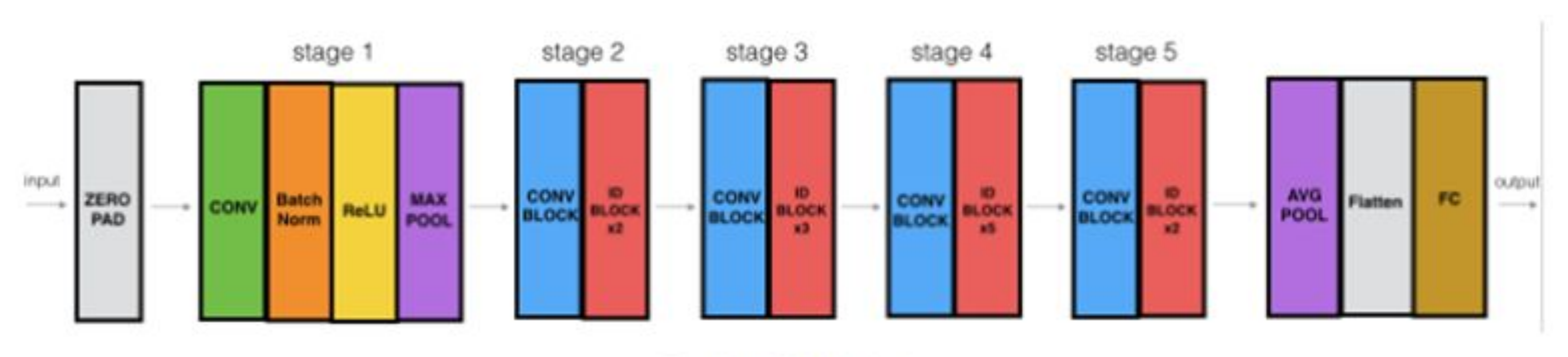
1. Conv2D
2. Maxpooling2D
3. Flatten
4. Relu as activation function
5. Softmax

Then the model is compiled with

1. optimizer: Adam
2. Metrics : Accuracy

Building ResNet50 model

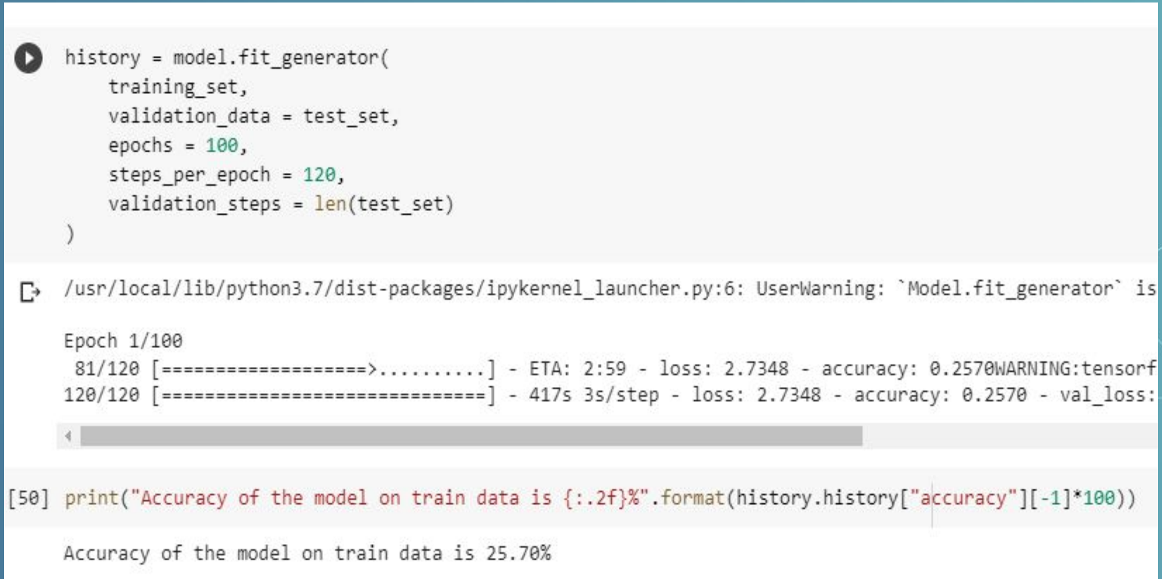
A ResNet50 is CNN that is 50 layers deep along with 1 Maxpool and 1 Average pool layer.The ResNet50 model has 5 stages each with convolution and identity block . each convolution block has 3 convolutional layers nd each identity block also has 3 convolutional layers



ResNet-50 Model

The ResNet50 model is trained with same image preprocessing techniques that were followed on CNN model.

The models yields as accuracy of 25.70% in training set



```
history = model.fit_generator(
    training_set,
    validation_data = test_set,
    epochs = 100,
    steps_per_epoch = 120,
    validation_steps = len(test_set)
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: `Model.fit_generator` is

Epoch 1/100
 81/120 [===================>..........] - ETA: 2:59 - loss: 2.7348 - accuracy: 0.2570WARNING:tensorf
120/120 [==============================] - 417s 3s/step - loss: 2.7348 - accuracy: 0.2570 - val_loss:
```

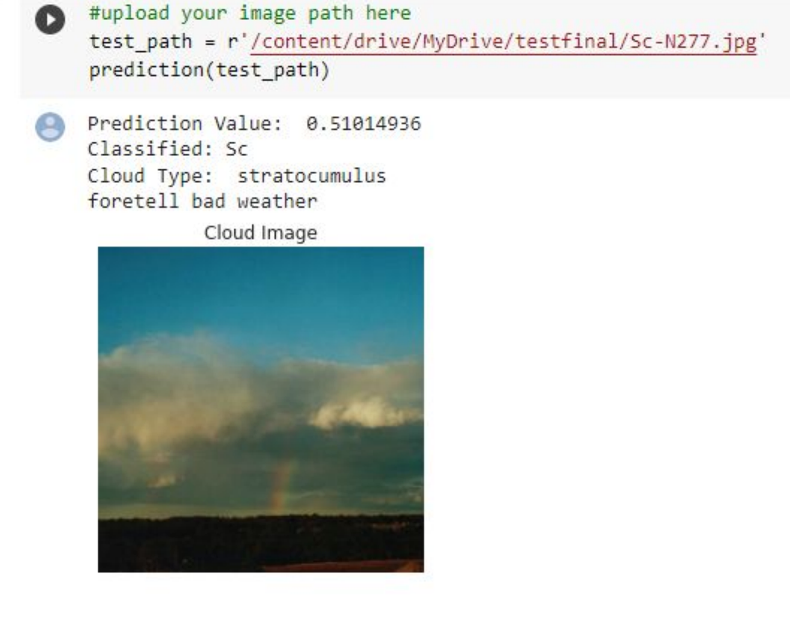
```
[50] print("Accuracy of the model on train data is {:.2f}%".format(history.history["accuracy"][-1]*100))
```

```
Accuracy of the model on train data is 25.70%
```

04

Test and Deployment

The model is finally tested with a test data and the results are obtained and deployed using vs code and streamlit.



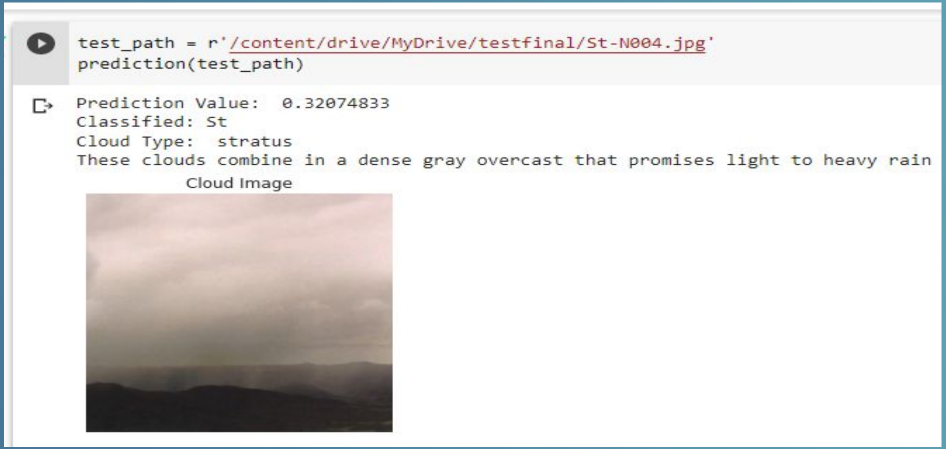
```
#upload your image path here
test_path = r'/content/drive/MyDrive/testfinal/Sc-N277.jpg'
prediction(test_path)
```

```
Prediction Value:  0.51014936
Classified: Sc
Cloud Type:  stratocumulus
foretell bad weather
```

Cloud Image

Test using CNN model

Test using ResNet50



```
test_path = r'/content/drive/MyDrive/testfinal/St-N004.jpg'
prediction(test_path)
```

```
Prediction Value:  0.32074833
Classified: St
Cloud Type:  stratus
These clouds combine in a dense gray overcast that promises light to heavy rain
```

Future works

- Applying various Transfer learning algorithm
- With object detection using computer vision for identification of cloud types can improve the accuracy of prediction with real life weather pattern and climate can be predicted more accurately.

THANK YOU