**TOPIC: - LANDSCAPE CLASSIFICATION**

**About Model**

This model predicts type of landscape like sea, building, street etc.

**Dataset**

Name: Intel Image Classification

Classes: Buildings, Forest, Glacier, Mountain, Sea, Street

No. of instances: 14034

Source: Kaggle

**Results**

F1 Score: 0.82

Train Accuracy: 0.8506

Test Accuracy: 0.8161

Train Loss: 0.4503

Test Loss: 0.5660

Precision: 0.82

Recall : 0.81

AUC for :-

Class 0- 0.96

Class 1- 0.99

Class 2- 0.96

Class 3- 0.96

Class 4- 0.97

Class 5- 0.97

* **Screen Shots** of ROC curve and performance metrics are on the last page of the document

# Data Pre-processing :

* **Rescaling** : Rescaling in image classification is the process of normalizing pixel values in an image to a standard scale. This is often done as a preprocessing step to improve the performance of image classification models.

When an image is captured, the pixel values can range from 0 to 255, depending on the bit depth of the camera sensor. Rescaling involves transforming these pixel values to a standard range, such as between 0 and 1, or between -1 and 1. This makes it easier for machine learning algorithms to process the images, as the pixel values are in a more consistent and manageable range.

* **Augmentation** :
* Random flip
* Random rotate
* Random zoom

# Model Used (CNN) :

A Convolutional Neural Network (CNN) is a type of neural network that is commonly used for image classification, object detection, and image segmentation tasks. CNNs are designed to automatically learn hierarchical representations of the input data, which makes them well-suited for visual recognition tasks.

CNNs are composed of several types of layers, including convolutional layers, pooling layers, and fully connected layers. Each layer in a CNN performs a different type of operation on the input data, allowing the network to learn increasingly complex representations of the input.

Here's a more detailed explanation of each layer in a CNN:

**Convolutional layer**: A convolutional layer applies a set of filters (also called kernels) to the input image to produce a set of feature maps. Each filter is a small matrix of numbers that is convolved (i.e., slid) over the input image, performing a dot product at each position. The resulting output values are then combined into a feature map. The weights of the filters are learned during training through backpropagation.

**Activation layer:** An activation layer applies a non-linear function to the output of a convolutional layer. The most commonly used activation function in CNNs is the rectified linear unit (ReLU), which sets all negative values to zero and leaves positive values unchanged.

**Pooling layer:** A pooling layer reduces the spatial size (i.e., the width and height) of the feature maps by taking the maximum, average, or sum of a local region of the feature map. This helps to reduce the number of parameters in the model and make it more robust to small translations and distortions in the input image.

**Fully connected layer:** A fully connected layer takes the flattened output of the previous layer and applies a matrix multiplication to produce a set of output values. This layer is similar to the hidden layers in a traditional neural network, and is typically used to perform classification or regression tasks.

In addition to these basic layers, CNNs can also include other types of layers, such as dropout layers (which randomly remove some of the connections between neurons to prevent overfitting) and batch normalization layers (which normalize the inputs to each layer to improve the stability and speed of training).

The architecture of a CNN typically consists of several layers of convolutional and pooling layers, followed by one or more fully connected layers. The final layer of the network is typically a softmax layer, which computes the probabilities of each class based on the output of the previous layer.

During training, the weights of the network are learned using an optimization algorithm such as stochastic gradient descent (SGD) or Adam. The loss function used to train the network depends on the specific task at hand. For example, for classification tasks, the cross-entropy loss function is commonly used.

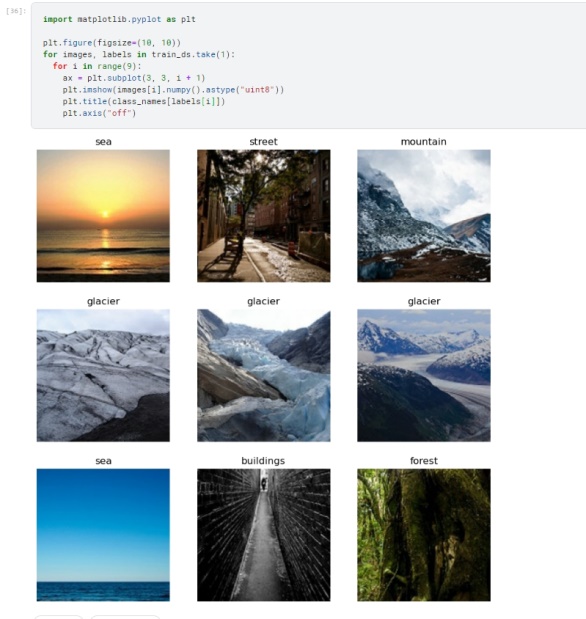
In summary, a CNN is a type of neural network that is designed to automatically learn hierarchical representations of the input data, making it well-suited for visual recognition tasks such as image classification, object detection, and image segmentation. The architecture of a CNN consists of several types of layers, each of which performs a different operation on the input data. The weights of the network are learned through backpropagation, using an optimization algorithm and a loss function that depend on the specific task at hand.

**SCREENSHOTS OF THE PROJECT :**

* Imported libraries :



* **Dataset (Images) with labels**

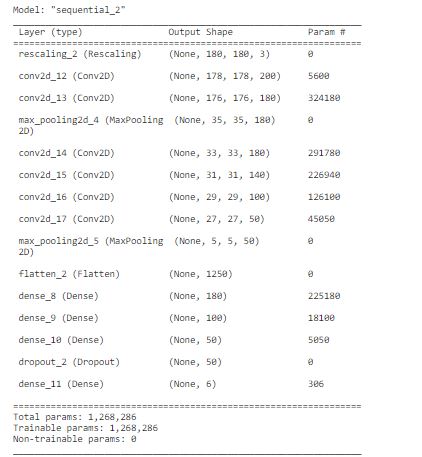


* **Pre-processing - Rescaling and Data Augmentation**

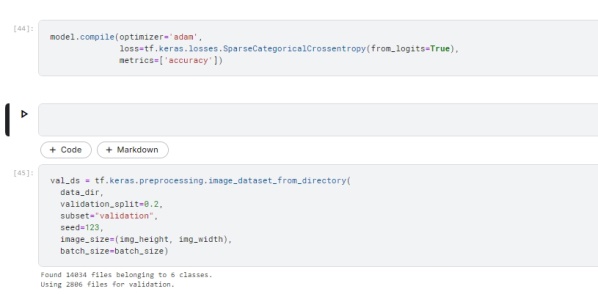




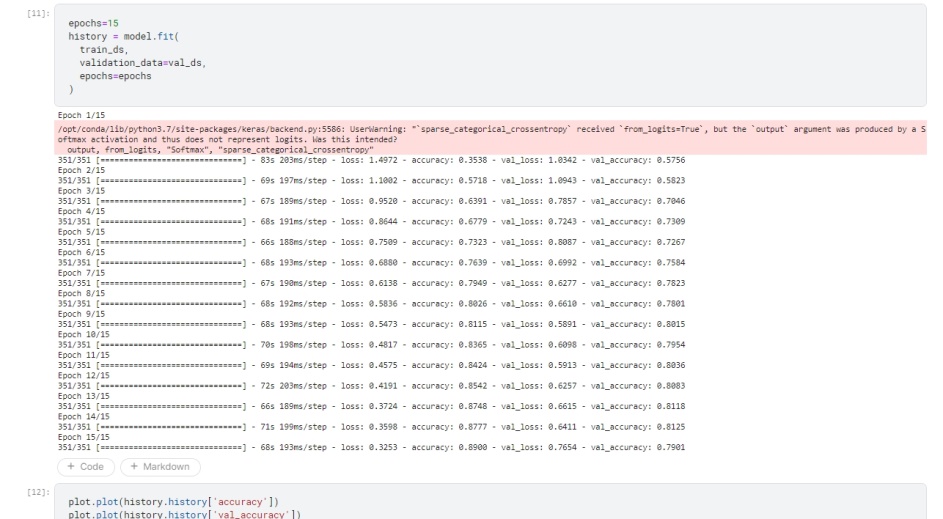
* **Model Summary:**



* **Compiling the Model and Splitting the dataset into Test(20%) and Train(80%)**

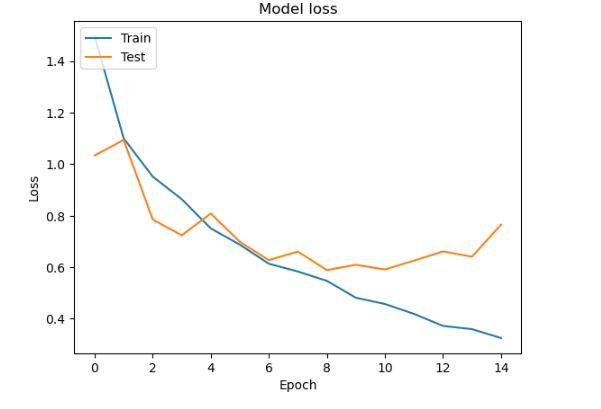
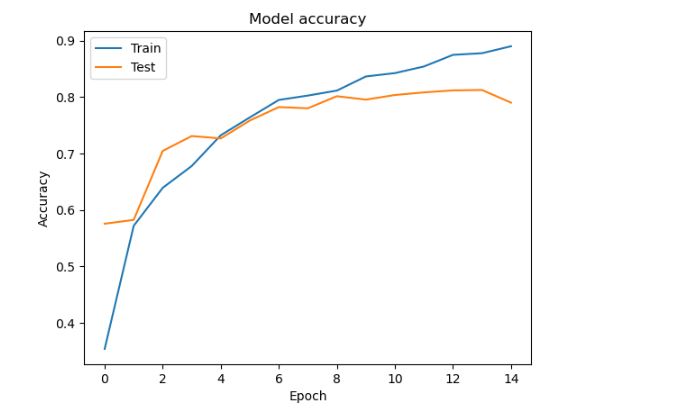


* **Epochs-15**



* **Plot for Accuracy and Loss**

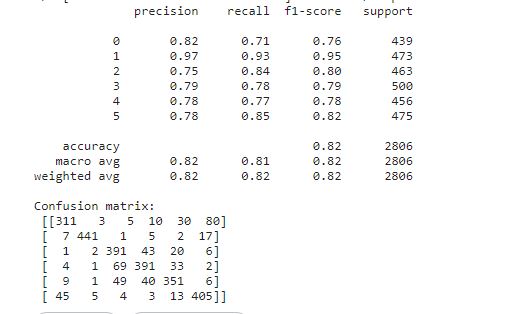




* **DIFFERENT PERFORMANCE METRICS:**

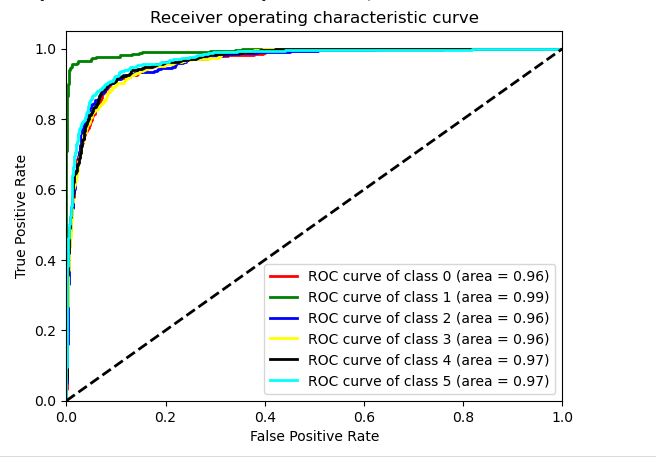
1) CLASSIFICATION REPORT:





1. ROC CURVE:





* GUI for Running the prediction results:



