

**19AIE205**

**PYTHON FOR ML**

**REMOTE SENSING IMAGE CLASSIFICATION USING ML**

**TEAM MEMBERS**

**BATCH – 13**

KARTHIKEYAN M - CB.EN.U4AIE20029​

PRAGADISHWARAN K - CB.EN.U4AIE20051​

KARTHIK RAJA B - CB.EN.U4AIE20028​

NALIN M R - CB.EN.U4AIE20041​

CHANDU JANAKIRAM - CB.EN.U4AIE20009​

**ACKNOWLEGEMENT**

We would like to thank to our professor Dr. Neethu Mohan who gave us her valuable suggestions and ideas when we needed them to work on our project

“**REMOTE SENSING IMAGE CLASSIFICATION USING ML**”

We are also grateful to our university, Amrita Vishwa Vidyapeetham for giving us the opportunity to work on this project. Last but not the least we would like to thank our group members, Nalin MR, Karthikeyan.M, Karthik Raja.B, Pragadishwaran K without their effort and co-operation, we would not have been able to complete the project within the prescribed time.

**ABSTRACT**

**Remote sensing** is the acquisition of information about an object or phenomenon without making physical contact with the object, in contrast to [in situ](https://en.wikipedia.org/wiki/In_situ) or on-site observation. The term is applied especially to acquiring information about the Earth and other planets. Remote sensing is used in numerous fields, including geography, land surveying and most Earth science disciplines (for example, hydrology, [ecology](https://en.wikipedia.org/wiki/Ecology), meteorology, oceanography, glaciology, geology); it also has military, intelligence, commercial, economic, planning, and humanitarian applications, among others.

In current usage, the term "remote sensing" generally refers to the use of satellite or aircraft-based sensor technologies to detect and classify objects on Earth. It includes the surface and the [atmosphere](https://en.wikipedia.org/wiki/Atmosphere) and [oceans](https://en.wikipedia.org/wiki/Oceans), based on [propagated signals](https://en.wikipedia.org/wiki/Wave_propagation) (e.g. [electromagnetic radiation](https://en.wikipedia.org/wiki/Electromagnetic_radiation)). It may be split into "active" remote sensing (when a signal is emitted by a satellite or aircraft to the object and its reflection detected by the sensor) and "passive" remote sensing (when the reflection of sunlight is detected by the sensor).

**TABLE OF CONTENTS**

**S. no TITLE Page No.**

1 INTRODUCTION 5

2 REMOTE SENSING IMAGE CLASSIFICATION 7

3 CONVOLUTIONAL NEURAL NETWORK 10

4 K- NEAREST NEIGHBOR 13

5 SUPPORT VECTOR MACHINE 18

6 DEEP NEURAL NETWORK 24

7 CODE IMPLEMENTATION 28

8 CONCLUSION 34

**INTRODUCTION**

Satellite imagery has a wide range of applications which is incorporated in every aspect of human life.​

Especially remote sensing has evolved over the years to solve a lot of problems in different areas.​

In Remote Sensing, hyperspectral remote sensors are widely used for monitoring the earth’s surface​with a high spectral resolution.​

**Hyperspectral Imaging** is an important technique in remote sensing, which collects the electromagnetic​ spectrum ranging from the visible to the near-infrared wavelength. Hyperspectral imaging sensors often​ provide hundreds of narrow spectral bands from the same area on the surface of the earth. In​ hyperspectral images (HSI), each pixel can be regarded as a high-dimensional vector whose entries​ correspond to the spectral reflectance in a specific wavelength.​

With the advantage of distinguishing subtle spectral differences, HSIs have been widely applied in​ diverse areas such as Crop Analysis, Geological Mapping, Mineral Exploration, Defence Research, Urban​ Investigation, Military Surveillance, Flood Tracking, etc.​

**REMOTE SENSING IMAGE CLASSIFICATION:**

Remote sensing image scene classification, which aims to classify remote sensing image into different types based on image content, has been attracted more and more attentions for its comprehensive application in fields of geography, ecology, city plan, forest monitor, military, etc .​

​

Remote sensing image scene classification essentially belongs to domains of machine learning and computer vision. With well-organized training dataset, models can be learned through minimizing loss functions between model output and ground-truth label.​

​

With explosive increasing of remote sensing data, analysis and processing remote sensing image effectively and efficiently becomes of great importance. ​

**What is Image Classification in Remote Sensing?**​

The 3 main **types of image classification** techniques in remote sensing are:​

​

* Unsupervised image classification​
* Supervised image classification​
* Object-based image analysis​

​

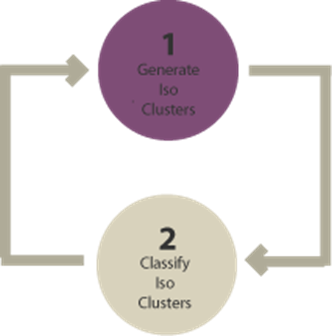
Unsupervised and supervised image classification are the two most common approaches.​

​

However, object-based classification has gained more popularity because it’s useful for high-resolution data.​

​

**Unsupervised Classification**​



In unsupervised classification, it first groups pixels into “clusters” based on their properties. Then, you classify each cluster with a land cover class.​

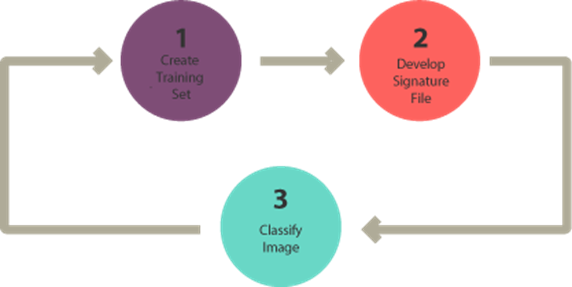
Overall, unsupervised classification is the most basic technique. Because you don’t need samples for unsupervised classification, it’s an easy way to segment and understand an image.​

​

* The two basic steps for unsupervised classification are:​
* Generate clusters​
* Assign classes​

**Supervised Classification**​

​

​

In supervised classification, you select representative samples for each land cover class. The software then uses these “training sites” and applies them to theentire image.​

The three basic steps for supervised classification are:​

​

* Select training areas​
* Generate signature file​
* Classify​ the images

​

​

For supervised image classification, you first create training samples. For example, you mark urban​ areas by marking them in the image. Then, you would continue adding training sites representative​

in the entire image.​

​

For each land cover class, you continue creating training samples until you have representative​samples for each class. In turn, this would generate a signature file, which stores all training samples​

spectral information.​

​

Finally, the last step would be to use the signature file to run a classification. From here, you would​

have to pick a classification algorithms such as:​

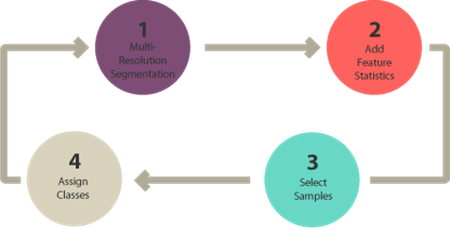
​

* Maximum likelihood​
* Minimum-distance​
* Principal components​
* Support vector machine (SVM)​
* Iso cluster​

​

**Object-Based Image Analysis (OBIA)**​

​



Supervised and unsupervised classification is pixel-based. In other words, it creates​

square pixels and each pixel​

has a class. But object-based image classification groups pixels into representative​

vector shapes with size and​

Geometry.​

​

Here are the steps to perform object-based image analysis classification:​

* Perform multiresolution segmentation​
* Select training areas​
* Define statistics​
* Classify​ the images

​

**CONVOLUTIONAL NEURAL NETWORK**

**Convolutional Neural Network** is one of the main categories to do image classification and image recognition in neural networks. Scene labeling, objects detections, and face recognition, etc., are some of the areas where convolutional neural networks are widely used.

In CNN, each input image will pass through a sequence of convolution layers along with pooling, fully connected layers, filters (Also known as kernels). After that, we will apply the Soft-max function to classify an object with probabilistic values 0 and 1.

**Convolution Layer**

Convolution layer is the first layer to extract features from an input image. By learning image features using a small square of input data, the convolutional layer preserves the relationship between pixels. It is a mathematical operation which takes two inputs such as image matrix and a kernel or filter.

## Strides

Stride is the number of pixels which are shift over the input matrix. When the stride is equaled to 1, then we move the filters to 1 pixel at a time and similarly, if the stride is equaled to 2, then we move the filters to 2 pixels at a time. The following figure shows that the convolution would work with a stride of 2.

**"Padding is an additional layer which can add to the border of an image."**

## **Pooling Layer**

Pooling layer plays an important role in pre-processing of an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "**downscaling**" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density.

## **Fully Connected Layer**

The fully connected layer is a layer in which the input from the other layers will be flattened into a vector and sent. It will transform the output into the desired number of classes by the network.

In the above diagram, the feature map matrix will be converted into the vector such as **x1, x2, x3... xn** with the help of fully connected layers. We will combine features to create a model and apply the activation function such as **softmax** or **sigmoid** to classify the outputs

Deep learning is a subset of machine learning that yields high-level abstractions by compositing multiple​

non-linear transformations. Among deep learning algorithms, Convolutional Neural Networks (CNNs) have​

gained popularity in computer vision and remote sensing fields, especially for image classification.​

​

Convolutional Neural Networks (CNN) is a type of deep learning method that uses convolutional​

multiplication based on artificial neural networks. Recently, CNN has been widely used in land cover​

classification, showing remarkable performance. Typical CNNs are composed of convolutional layers, pooling​ layers, and fully connected layers.​

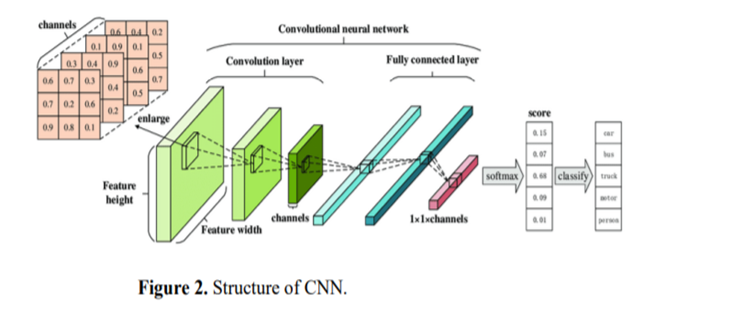
**WORKING OF CNN:**

**​**

CNN is a deep learning network with following benefits over models: CNNs uses a convolution procedure onthe pixels of an image to retrieve the image's features. CNN can display image data, and gather image information from large data easily. ​

​

CNNs specifically learn to analyze two-dimensional (2D) shapes that transform the original input into the output. Inside a CNN, each neuron is connected within the preceding layer neurons, thereby reducing weights in the network. As shown in  the Figure , this network consists of the convolutional layer, pooling layers and a fully connected output layer. The results of the classification are generated from the final output layer.​



​

A CNN's core layer is the convolutional layer . The convolution kernel/filter is an array of weight matrix. The characteristic map formed by convolving the input matrix and filter can be the input for the convolutional neural network's successive stages. Furthermore, the neurons in each input image convolve with the kernel in every convolutional layer. A CNN can have any number of convolutional layers.

**Steps Involved in CNN classification:**

Step 1: Choose a Dataset ​

Step 2: Prepare Dataset for Training ​

Step 3: Create Training Data ​

Step 4: Shuffle the Dataset ​

Step 5: Assigning Labels and Features ​

Step 6: Normalising X and converting labels to categorical data. ​

Step 7: Split X and Y for use in CNN ​

Step 8: Define, compile and train the CNN Model ​

Step 9: Accuracy and Score of model ​

**K-NEAREST NEIGHBOR**

# **K-Nearest Neighbor(KNN) Algorithm for Machine Learning**

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

**How does K-NN work?**

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

**How to select the value of K in the K-NN Algorithm?**

Below are some points to remember while selecting the value of K in the K-NN algorithm:

Skip Ad

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

**IMAGE CLASSIFICATION USING KNN**

K-nearest neighbor (KNN) is a common classification method for data mining techniques. It has been widely used in many fields because of the implementation simplicity, the clarity of theory and the excellent classification performance. But KNN will increase classification error rate when training samples distribute unevenly or sample number of each class is very different. So, learning from the idea of clipping-KNN, we adopt an improved KNN classification algorithm and applies it to

object-oriented classification of high resolution remote sensing image.

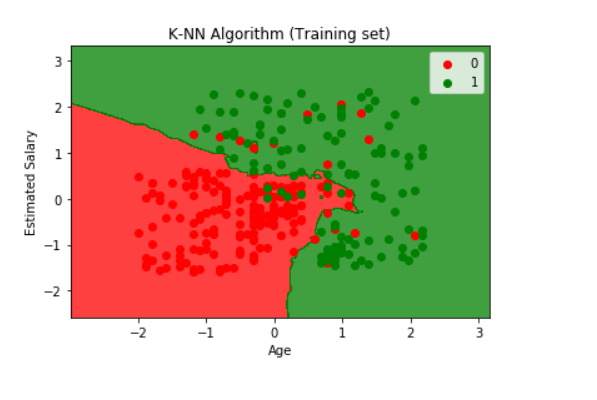
Firstly, as sample points,  image objects are obtained through image segmentation.

Secondly, original KNN, clipping-KNN and the improved KNN are introduced and used to classify those sample points respectively.

kNN provides objective, fast, transparent and produces good results over larger areas. The main advantage of kNN algorithm is its simplicity and lack of parametric assumptions

**WORKING OF KNN:**

After the model has stored the training set for prediction, it takes a test image to be predicted, calculates the distance to every image in the training set and obtains the ‘k’ training images closest to the test image. It then outputs the class according to some voting procedure from the labels of these ‘k’ neighbours , generally a majority vote.​



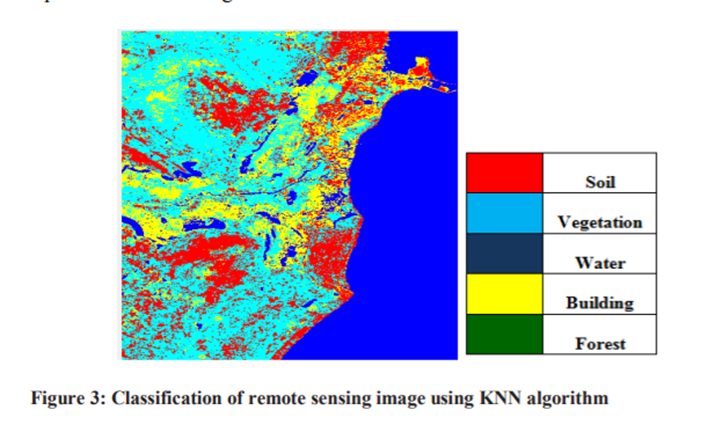
​

As k-NN is a very simple algorithm it doesn’t really have a lot of hyperparameters to tweak, just the two: the distance metric and the value of ‘k’. So what we can do is, run our model for various values of ‘k’ and get the model with the best validation accuracy, which will be used as our final model on the test set.​

​

The model only consists of the training data, that is, the model simply learns the entire training set and for prediction gives the output as the class with the majority in the ‘k’ nearest neighbours calculated according to some distance metric.​

Example :  If an unknown sample and known training data are taken and the distances between all training set samples and unknown samples will be calculated. The smallest distance value corresponds to their training set sample which are close to unknown sample. Thus the unknown sample can be classified on the basis of their nearest neighbour. It is highly required for the extraction of edges which is to be completely connected as there are several features values and the objects co-existing in different shapes and sizes in satellite images (LANSAT 8). The measurement of smoothness technique of an image by using Gradian operators and criterion of Euclidean is evaluated .​



​

# **SUPPORT VECTOR MACHINE**

# **Support Vector Machine Algorithm**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

## **Types of SVM**

**SVM can be of two types:**

* **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

**IMPORTANT TERMS:**

## Hyperplane and Support Vectors in the SVM algorithm:

**Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

**Support Vectors:**

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

**SVM Parameters**

SVM Parameters include the values, estimators, and various constraints used to implement ML algorithms. There are three types of SV parameters in a [Neural Network](https://intellipaat.com/blog/tutorial/machine-learning-tutorial/neural-network-tutorial/):

* **Kernel**

Kernel transforms the input data into any first as per the user requirements. The Kernels used in SVM could be linear, polynomial, radial basis functions (RBFs), and non-linear hyperplanes, created using the polynomial and RBF functions. You can obtain accurate classifiers by separating non-linear classes through an advanced kernel.

* **Regularization**

The C parameters in Scikit-learn denote the error or penalty representing any miscalculation. You can maintain regularization by understanding the miscalculation and changing the decision boundary through tweaking the C parameters.

* **Gamma**

Gamma parameters determine their influence over a single training example. There are two types of gamma parameters, low meaning ‘far’ and high meaning ‘close’ values. The low or far values define a Gaussian function with a large variance. Whereas, high or close values define it with small variance.

**IMAGE CLASSIFICATION USING SVM:**

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. ​

The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. ​

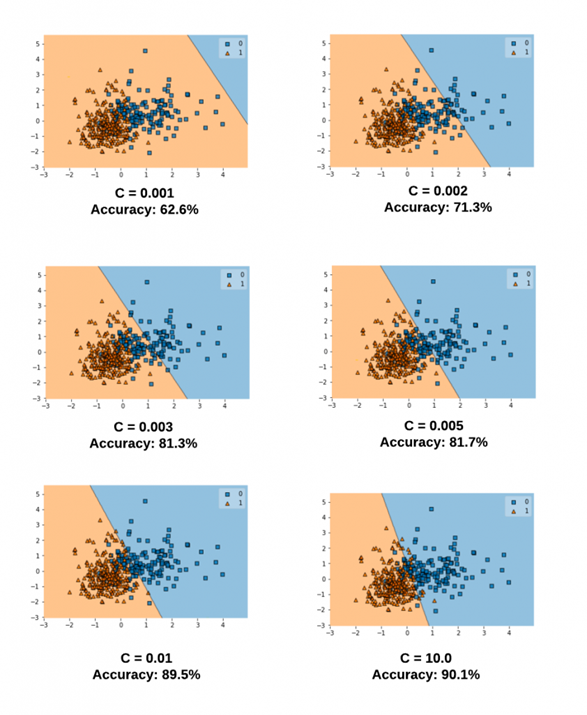
If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three. ​

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. The figure below shows the decision function for a linearly separable problem, with three samples on the margin boundaries, called “support vectors”​

​

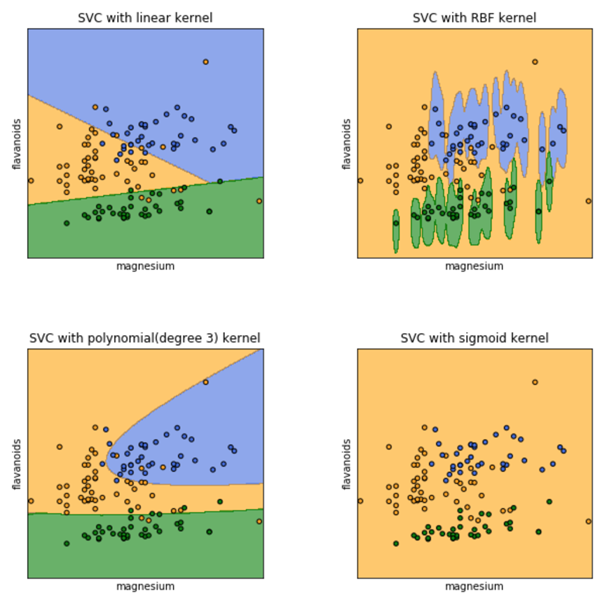
**SVM SEGREGATION**

The Support Vector Machine (SVM) classification process is based on finding an optimal separation hyperplane performing the minimum distance. ​



The optimal hyperplane is defined by a subset of feature vectors from the learning database named Support vectors denoted V. The classification problem is equivalent to a quadratic optimization with constraints  The optimization problem is parameterized by a penalty parameter C that describes the separation complexity and the classification error.​

**SVM KERNELS**



**DIFFERENT KERNELS:**​

​

1. Linear Kernel​

2. Polynomial Kernel​

3. Sigmoid Kernel​

4. Gaussian Kernel​

5. Bessel function kernel​

6. ANOVA kernel​

7. Radial Basis Function kernel​

​

**Steps Involved In SVM :**

 It is a non-probabilistic model in which the dataset is trained using a trainer. The number of regions are mapped together and mainly used in image segmentation and classification purpose to identify the features. It helps in finding accurate and better results in classification purpose. It includes following stages: ​

​

► First of images are preprocessed ​

►Dividing the database into test set and training set​

► choose data for input  ​

►then training method is selected that includes: ​

 • Multi class training ​

 • Choose Kernel​

►Train the dataset​

►Test and evaluate the performance​

Advantages of SVM:

1. Effective in high dimensional cases
2. Its memory efficient as it uses a subset of training points in the decision function called support vectors
3. Different kernel functions can be specified for the decision functions and its possible to specify custom kernels

​

**DEEP NEURAL NETWORK**

**NEURAL NETWORKS**

Neural Networks are multi-input, single-output systems made up of artificial neurons. A Neural Network's principal function is to convert input into meaningful output. A Neural Network usually has an input and output layer, as well as one or more hidden layers. All of the neurons in a Neural Network influence each other, thus they are all connected.

**DEEP NEURAL NETWORKS**

We will use the models and the hidden layers to combine them and create non-linear models which best classify our data. Sometimes our data is too complex and to classify that we will have to combine non-linear models to create even more non-linear model.

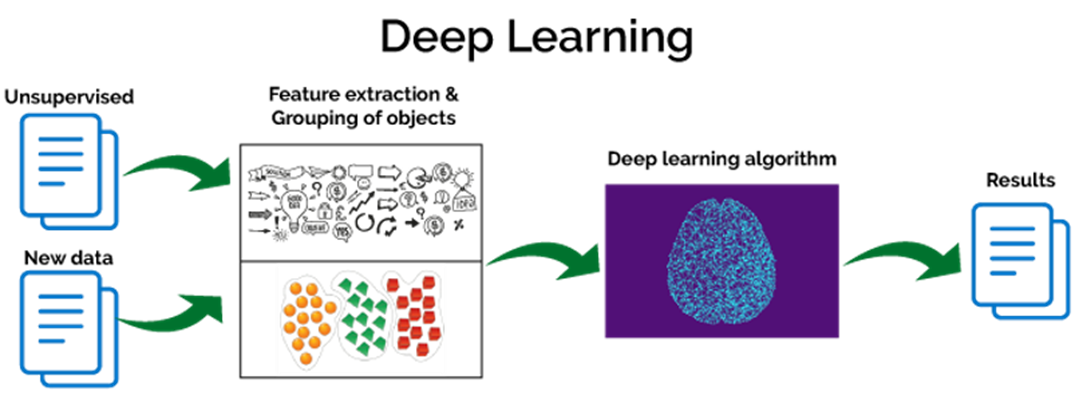
We can do this many times with even more hidden layers and obtain highly complex models as

To classify this type of data is more complex. It requires many hidden layers of models combining into one another with some set of weight to obtain a model that perfectly classify this data.

After that, we can produce some output through a feed-forward operation. The input would have to go through the entire depth of the neural network before producing an output. It is just a multilayered perceptron. In a deep neural network, our data's trend is not straight forward, so this non-linear boundary is only an accurate model that correctly classifies a very complex set of data.

Many hidden layers are required to obtain this non-linear boundary and each layer containing models which are combined into one another to produce this very complex boundary which classifies our data.

The deep neural networks can be trained with more complex function to classify even more complex data.



**FORWARD PROPOGATION**

**Feedforward Networks:** In this model, the signals only travel in one direction, towards the output layer. Feedforward Networks have an input layer and a single output layer with zero or multiple hidden layers. They are widely used in pattern recognition.

**Feedback Networks:** In this model, the recurrent or interactive networks use their internal state (memory) to process the sequence of inputs. In them, signals can travel in both directions through the loops (hidden layer/s) in the network. They are typically used in time-series and sequential tasks.

**Hidden layer**

The layer or layers hidden between the input and output layer is known as the hidden layer. It is called the hidden layer since it is always hidden from the external world. The main computation of a Neural Network takes place in the hidden layers. So, the hidden layer takes all the inputs from the input layer and performs the necessary calculation to generate a result. This result is then forwarded to the output layer so that the user can view the result of the computation.

In a Neural Network, the learning (or training) process is initiated by dividing the data into three different sets:

* **Training dataset –** This dataset allows the Neural Network to understand the weights between nodes.
* **Validation dataset –** This dataset is used for fine-tuning the performance of the Neural Network.
* **Test dataset –** This dataset is used to determine the accuracy and margin of error of the Neural Network.

Once the data is segmented into these three parts, Neural Network algorithms are applied to them for training the Neural Network. The procedure used for facilitating the training process in a Neural Network is known as the optimization, and the algorithm used is called the optimizer.

**Need for back propagation**

We are still making use of a gradient descent optimization algorithm which acts to minimize the error of our model by iteratively moving in the direction with the steepest descent, the direction which updates the parameters of our model while ensuring the minimal error. It updates the weight of every model in every single layer. We will talk more about optimization algorithms and backpropagation later.

**Back propagation**

For a single training example, **Backpropagation** algorithm calculates the gradient of the **error function**. Backpropagation can be written as a function of the neural network. Backpropagation algorithms are a set of methods used to efficiently train artificial neural networks following a gradient descent approach which exploits the chain rule.

The main features of Backpropagation are the iterative, recursive and efficient method through which it calculates the updated weight to improve the network until it is not able to perform the task for which it is being trained. Derivatives of the activation function to be known at network design time is required to Backpropagation.

To update the weight, we calculate the error correspond to each weight with the help of a total error. The error on weight w is calculated by differentiating total error with respect to w.

Now, we will backpropagate this error to update the weights using a backward pass.

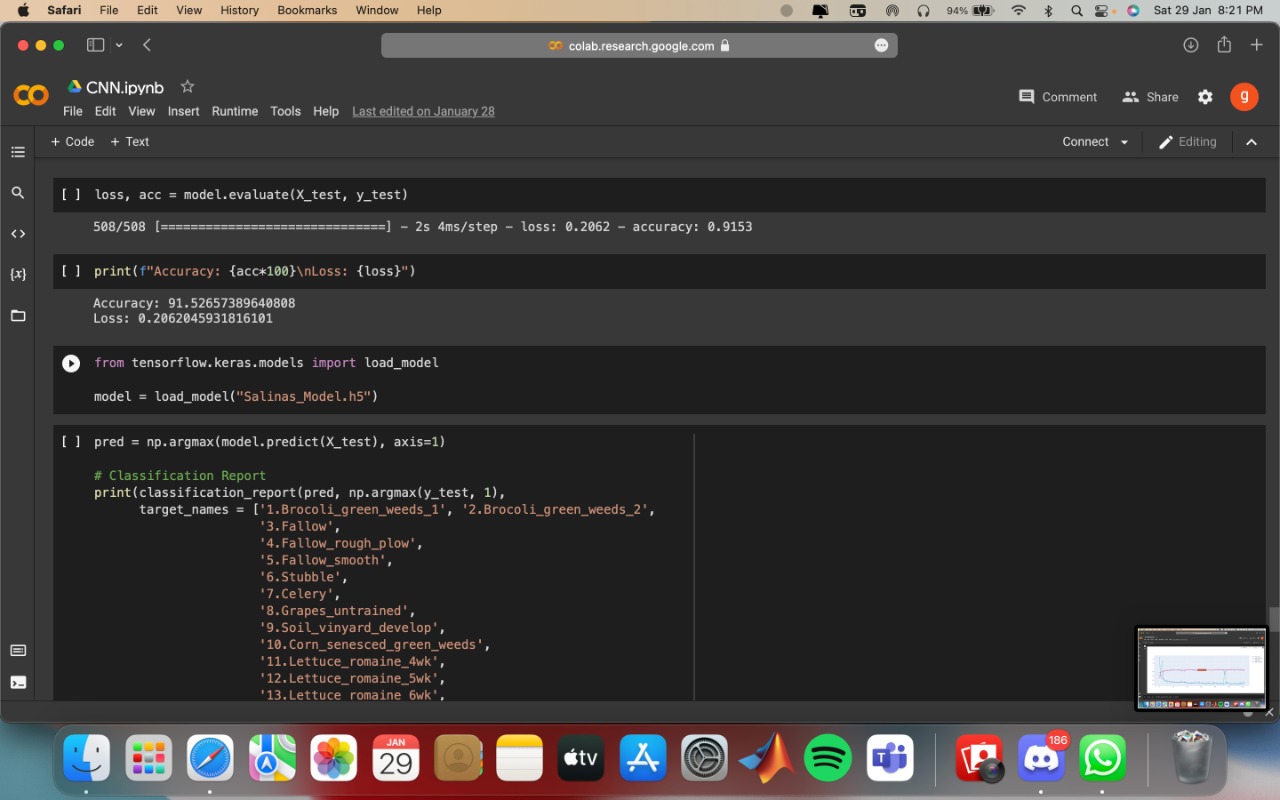
**What is depth of neural network?**

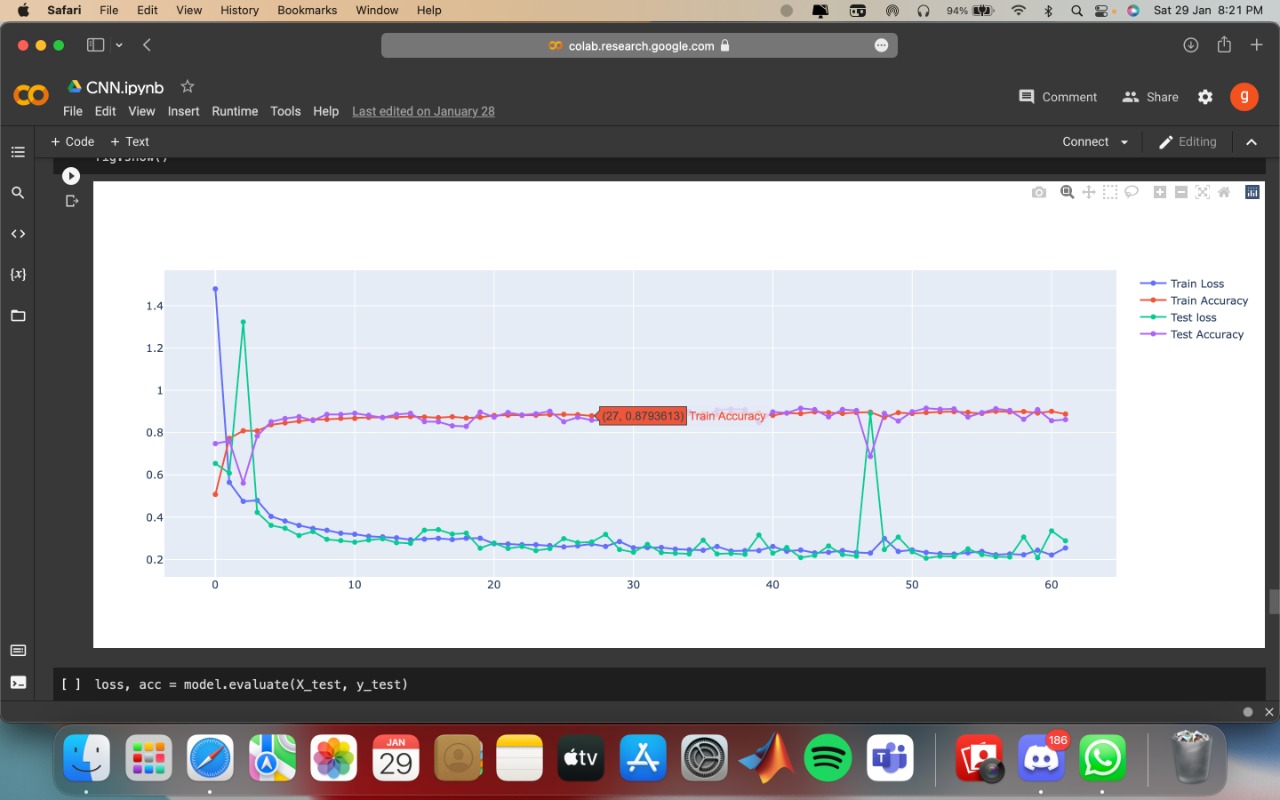
There can be multiple hidden layers which depend on what kind of data you are dealing with. The number of hidden layers is known as the depth of the neural network. The deep neural network can learn from more functions. Input layer first provides the neural network with data and the output layer then make predictions on that data which is based on a series of functions. ReLU Function is the most commonly used activation function in the deep neural network.

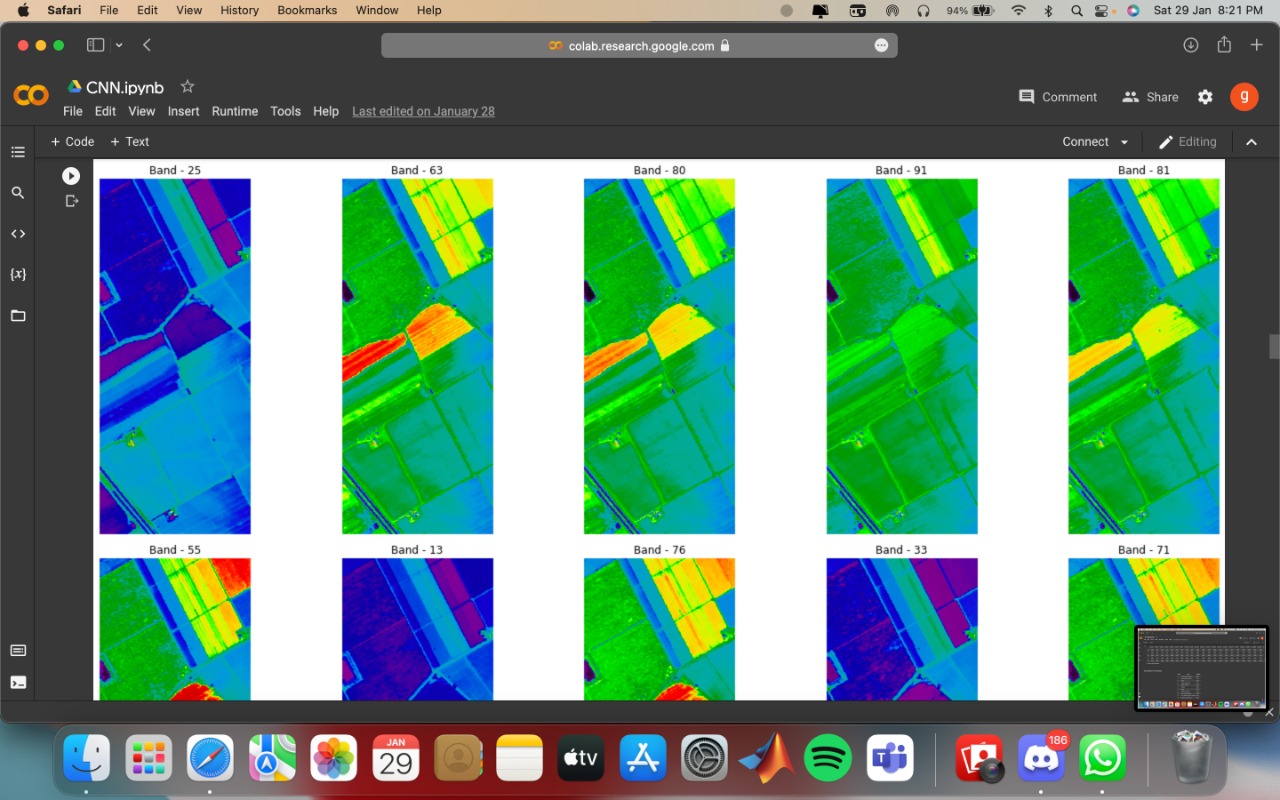
One of the main advantages of deep learning lies in being able to solve complex problems that require discovering hidden patterns in the data and/or a deep understanding of intricate relationships between a large number of interdependent variables.

**CODE EXECUTION**

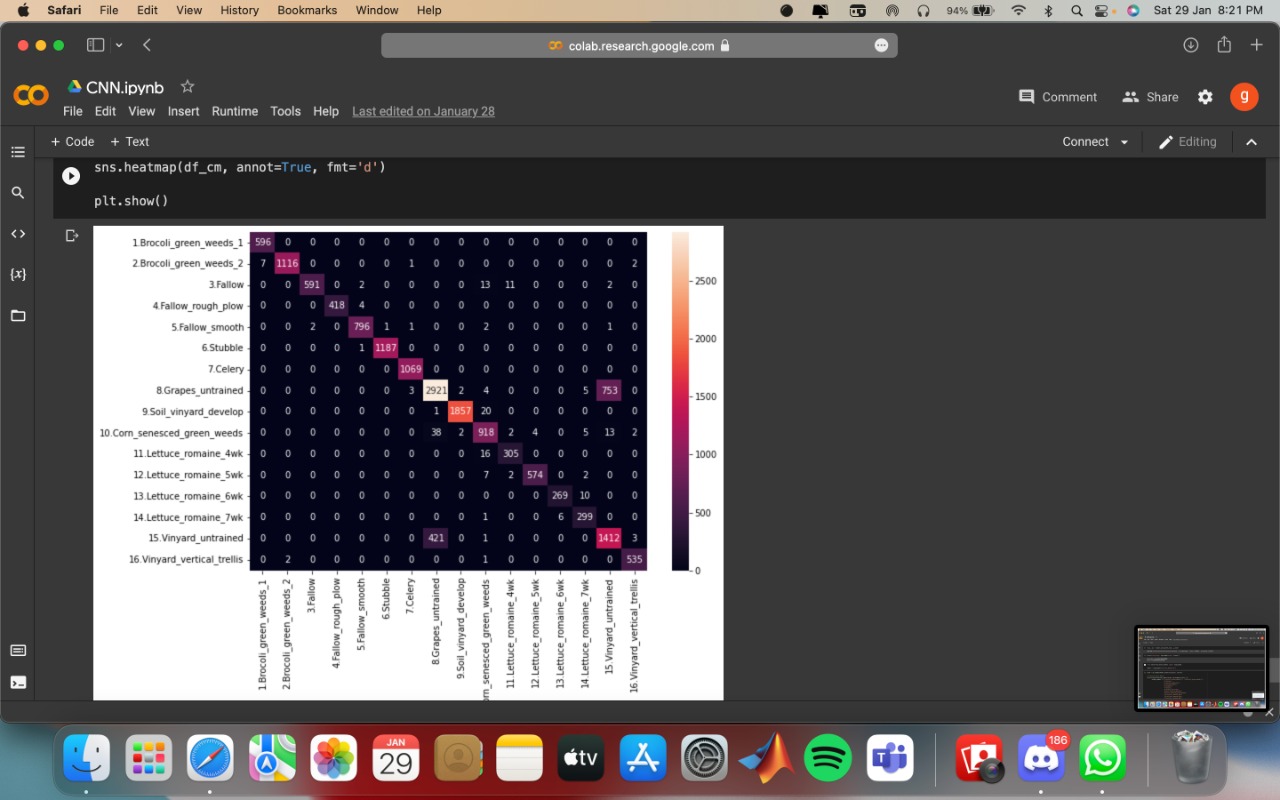
CNN:



Accuracy graph: 

Output :-

Heat map :-

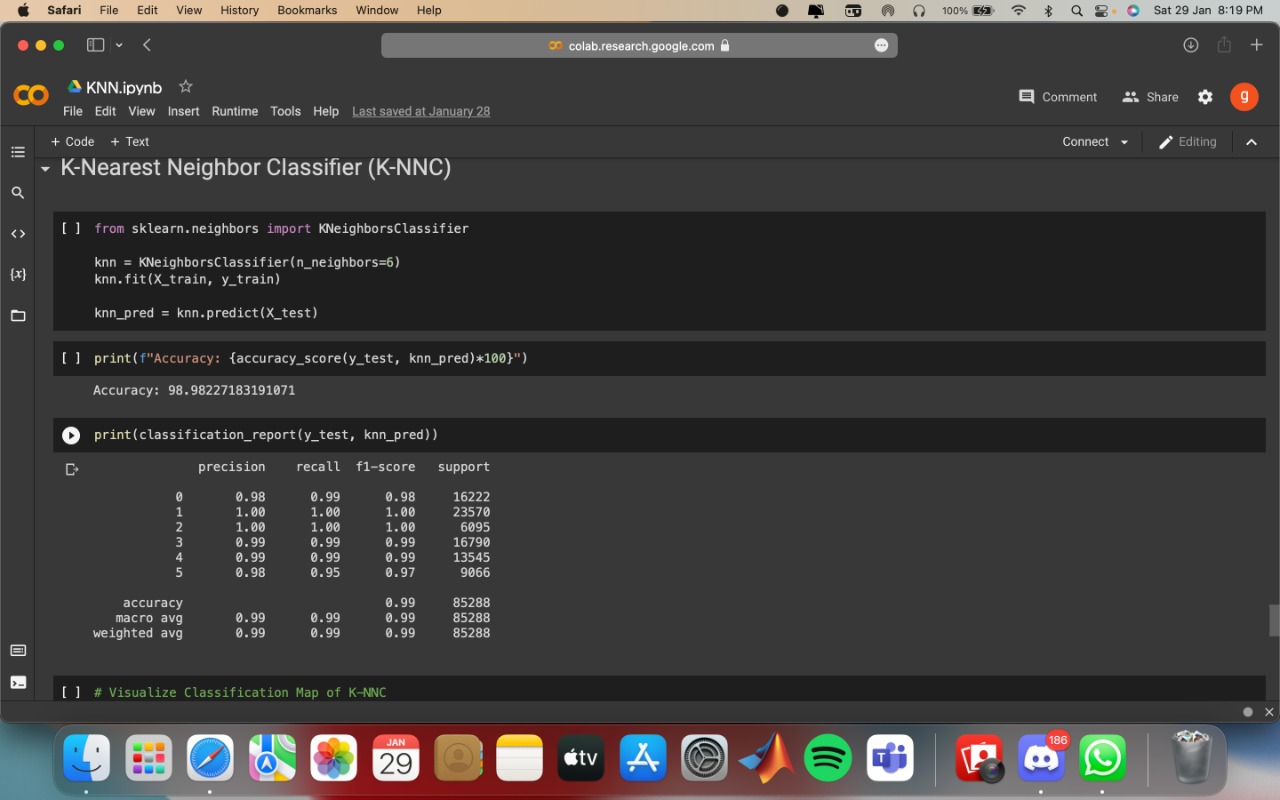


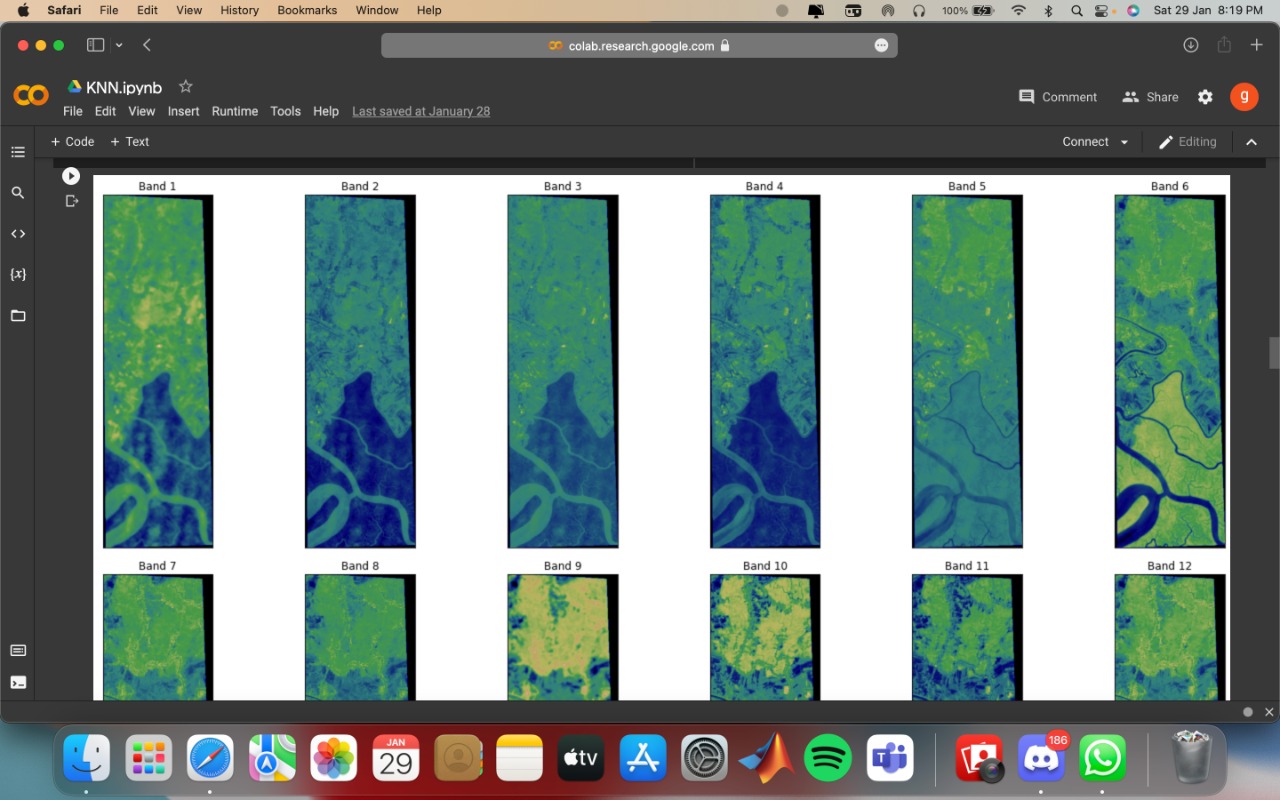
​

​

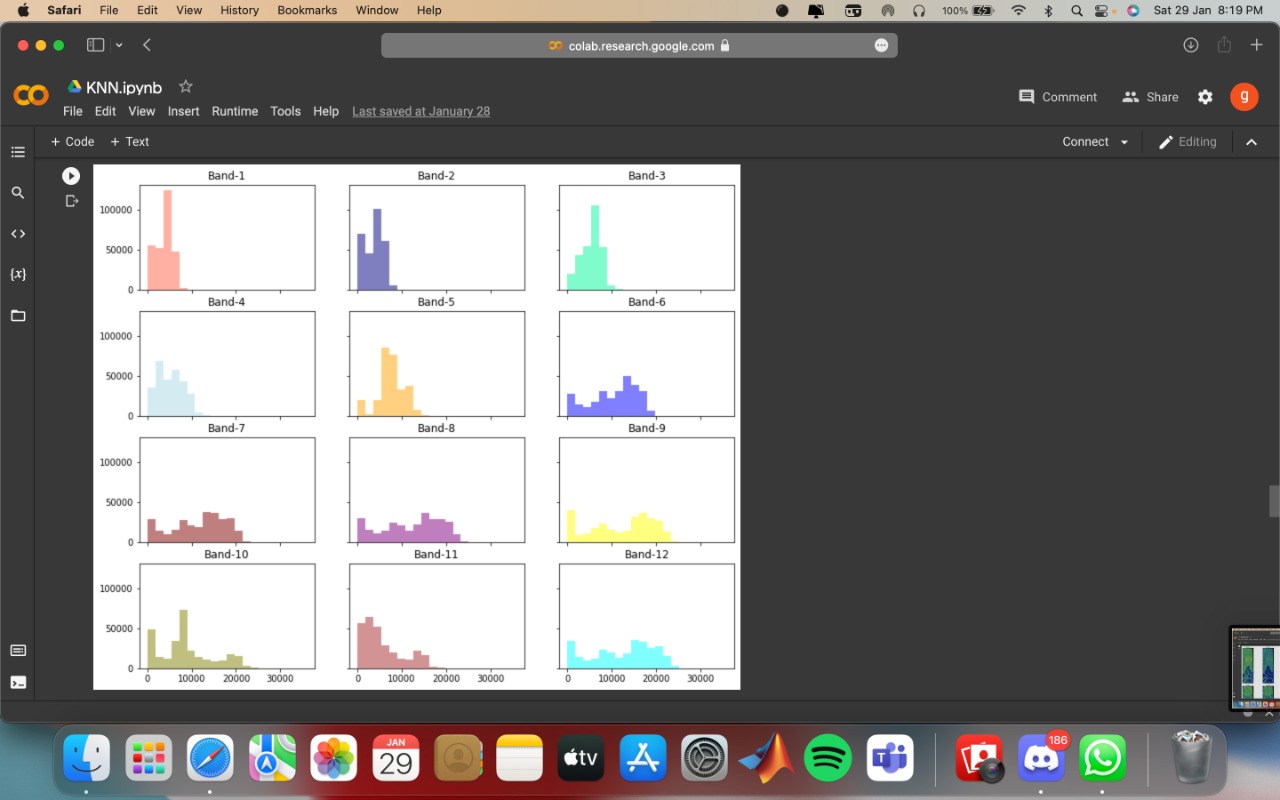
​

​KNN :

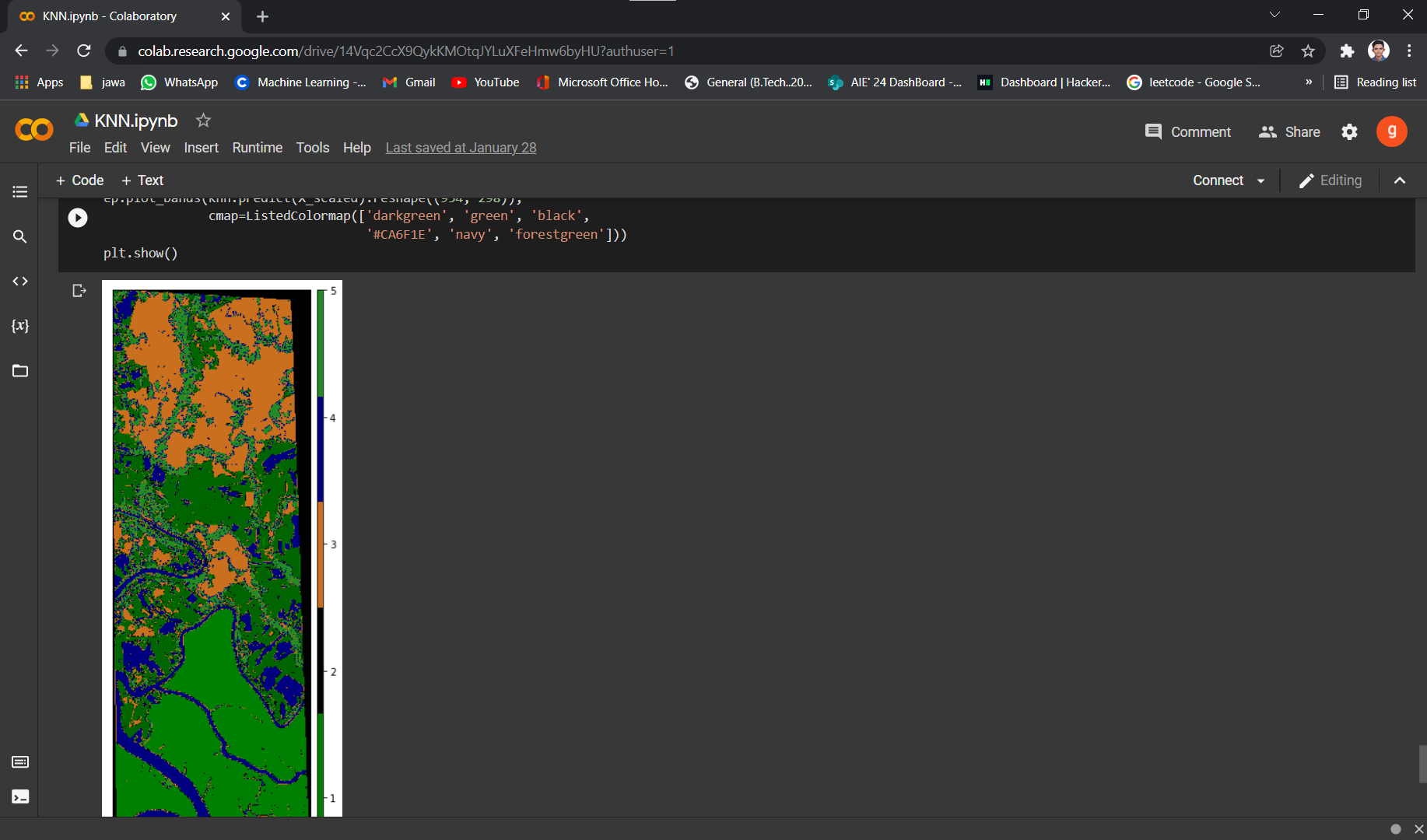
​

Input :- 

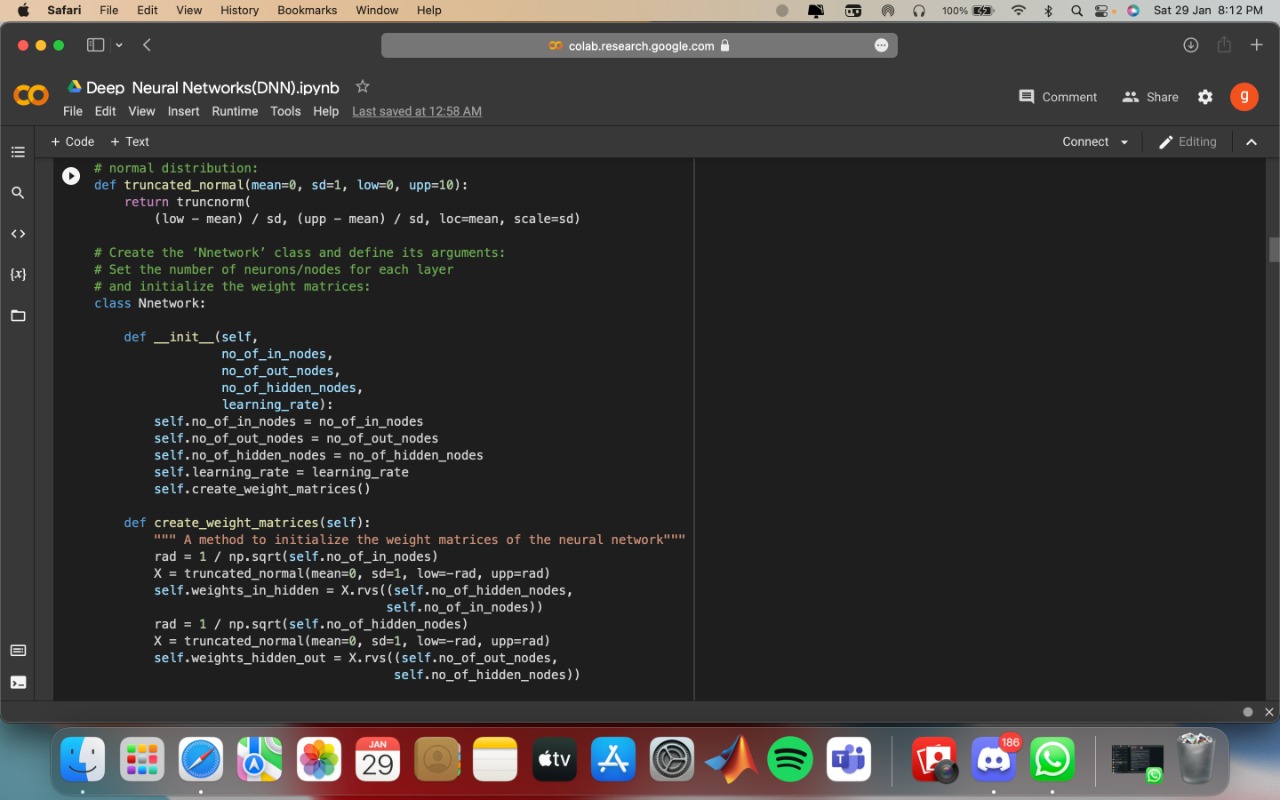
Different Band :-



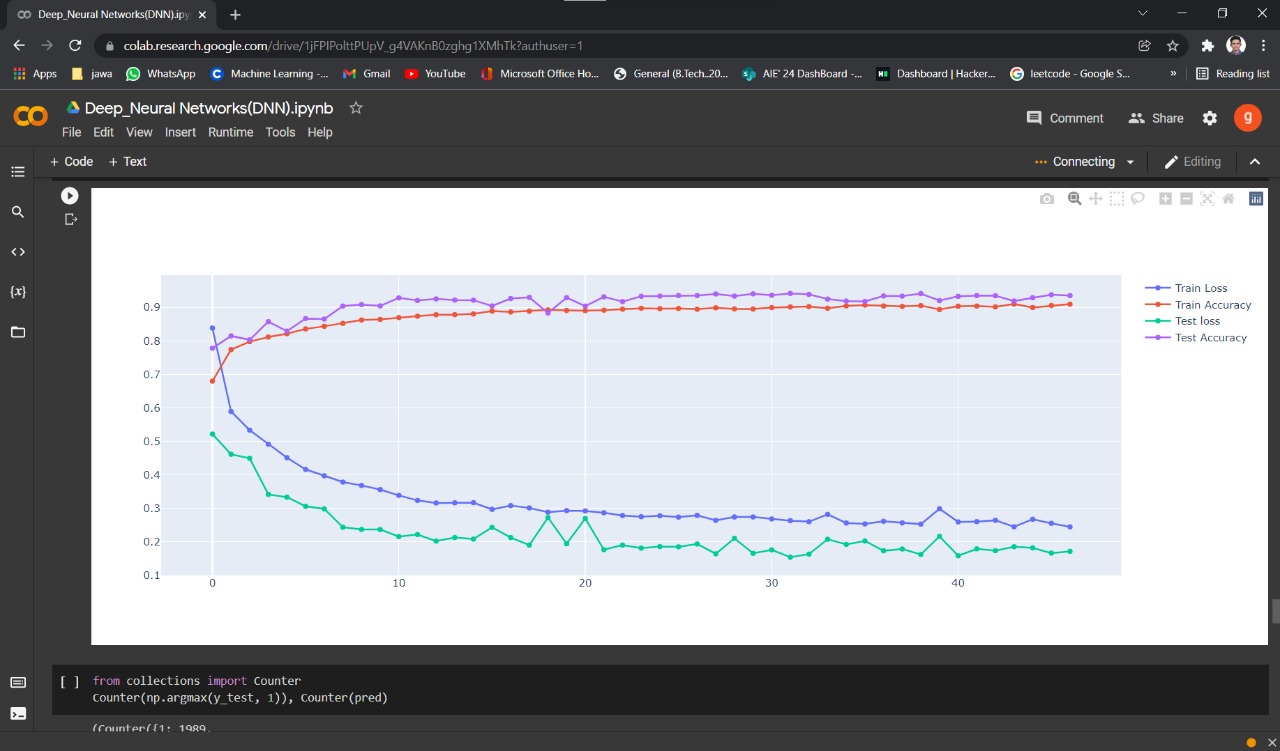
Output:-



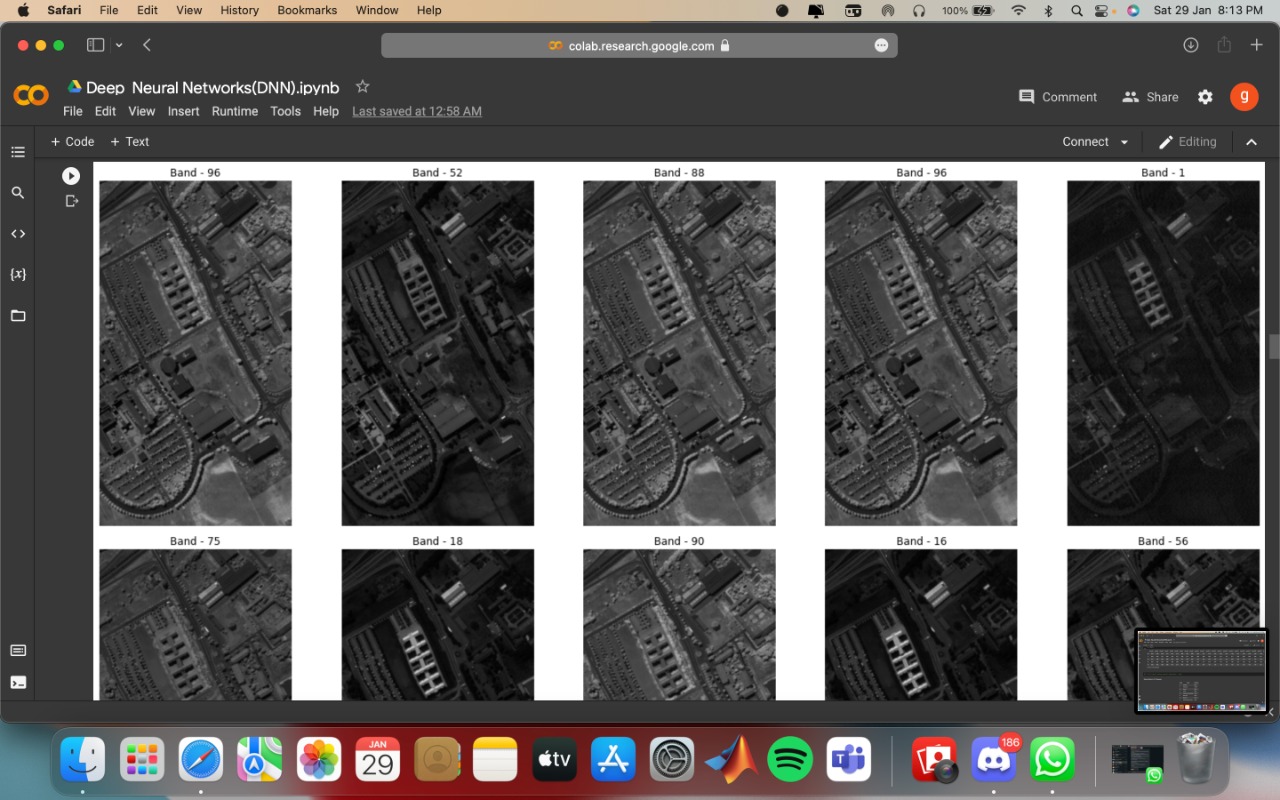
DNN :

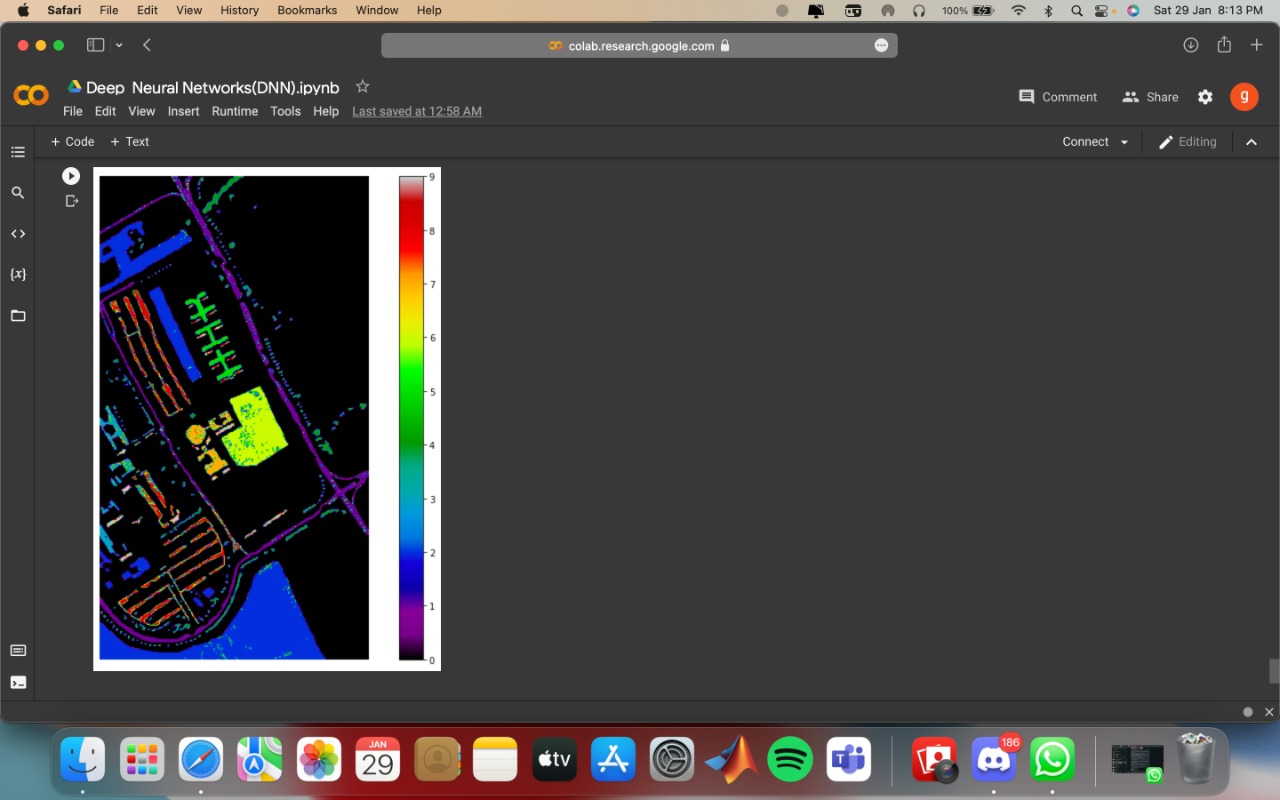


Accuracy graph:



Input :-



Output :- 

**CONCLUSION**

By taking this module, I learnt to read more and think more, and to draw information from a wide range of sources, such as lecture notes, literature, journals, and websites, to support my essential learning. Therefore, I can go beyond the superficial knowledge of remote sensing to understand how satellite images are acquired and used to different applications. Even though I still do not think I have a very comprehensive knowledge of remote sensing, I have truly learnt a lot from both lectures and independent reading, and of course by implementing them through MATLAB software.

**REFERENCE**

1. DNN :- [**https://academic.oup.com/nsr/article/6/6/1082/5484863**](https://academic.oup.com/nsr/article/6/6/1082/5484863)
2. CNN :-

<https://www.mdpi.com/journal/remotesensing/special_issues/CNN_RS>

1. SVM :-

<https://iopscience.iop.org/article/10.1088/1755-1315/20/1/012038/pdf>

1. KNN :-

<https://www.mdpi.com/2072-4292/13/7/1345>