A REPORT
ON

AUTOMATIC LAND COVER CLASSIFICATION OF MULTI-TEMPORAL SATELLITE IMAGES

BY

Guntaas Singh	2018A7PS0269P
Nisarg Vora	2018A7PS0254P

AT

Indian Institute of Remote Sensing, Dehradun

A Practice School - I station of

Birla Institute of Technology and Science, Pilani

June, 2020

A REPORT

ON

AUTOMATIC LAND COVER CLASSIFICATION OF MULTI-TEMPORAL SATELLITE IMAGES

BY

ID Number	Name	Branch
2018A7PS0269P	Guntaas Singh	B.E. (Hons.) Computer Science
2018A7PS0254P	Nisarg Vora	B.E. (Hons.) Computer Science

Prepared in the partial fulfillment of the

Practice School - I course

AT

Indian Institute of Remote Sensing, Dehradun

A Practice School - I station of

Birla Institute of Technology and Science, Pilani

June, 2020

Acknowledgments

We would sincerely like to thank the Director of Indian Institute of Remote Sensing, Dr. Prakash Chauhan, for giving us an opportunity to work in this organization and gain a significant amount of exposure to corporate work culture, ethics, and etiquette to be followed while working in a professional environment.

We would like to extend our most sincere gratitude to our project in-charge, Dr. Hari Shanker Srivastava, Head of the Programme Planning and Evaluation Group (PPEG) at IIRS, for providing us with the opportunity to work with him on this project, and his guidance and mentorship during the same.

We wish to extend our gratitude to the faculty in charge of the PS-I program at IIRS, Dr. Rekha A., Assistant Professor at BITS Pilani - Bangalore Center, for her guidance and advice during the PS-I program, and her helpfulness and responsiveness while addressing all the concerns we raised during the same.

In addition, we would also like to thank the members of the Practice School Division, who have worked very hard for operating the PS-I programme remotely to ensure that we have a seamless learning experience.

BIRLA INSTITUTE OF SCIENCE AND TECHNOLOGY
PILANI (RAJASTHAN)
Practice School Division

Station: Indian Institute of Remote Sensing
Centre: Dehradun
Duration: From 18th May, 2020 to 27th June, 2020
Date of start: 18th May, 2020
Date of submission: 4th June, 2020
Title of project: Automatic Land Cover Classification of Multi-temporal Satellite Images

ID Number	Name	Branch
2018A7PS0269P	Guntaas Singh	B.E. (Hons.) Computer Science
2018A7PS0254P	Nisarg Vora	B.E. (Hons.) Computer Science

Name of guide: Dr. Hari Shanker Srivastava
Designation: Scientist/Engineer - SG. Group Head, Programme Planning and Evaluation Group (PPEG).
Name of PS faculty: Dr. Rekha A.

Abstract

Blah

BIRLA INSTITUTE OF SCIENCE AND TECHNOLOGY PILANI (RAJASTHAN)

Practice School Division

Response Option Sheet

Station: Indian Institute of Remote Sensing
Centre: Dehradun

ID Number	Name	Branch
2018A7PS0269P	Guntaas Singh	B.E. (Hons.) Computer Science
2018A7PS0254P	Nisarg Vora	B.E. (Hons.) Computer Science

Title of project: Automatic Land Cover Classification of Multi-temporal Satellite Images

Code No.	Response Option	Course No.(s) and Name
1	A new course can be designed out of this project.	
2	The project can help modification of the course content of some of the existing Courses	
3	The project can be used directly in some of the existing Compulsory Discipline Courses (CDC)/ Discipline Courses Other than Compulsory (DCOC)/ Emerging Area (EA), etc. Courses	
4	The project can be used in preparatory courses like Analysis and Application Oriented Courses (AAOC)/ Engineering Science (ES)/ Technical Art (TA) and Core Courses.	
5	This project cannot come under any of the above mentioned options as it relates to the professional work of the host organization.	

Signature
Date:

Contents

Acknowledgments . . . iii

Abstract . . . iv

1 Introduction 1

1.1 About IIRS . . . 1

1.2 Remote Sensing . . . 1

1.3 Land Cover Classification . . . 2

1.4 Normalized Difference Vegetation Index(NDVI) . . . 3

What is NDVI? . . . 4

Formula . . . 4

Illustration . . . 5

2 Methodology 7

2.1 Image Classification . . . 7

Pixel Based Approach . . . 7

Object Based Approach . . . 7

Semantic Approach . . . 7

2.2 Supervised Machine Learning . . . 8

2.3 Support Vector Machines . . . 8

2.4 Deep Learning and Neural Networks . . . 9

2.5 Convolutional Neural Networks . . . 11

Convolutional Layer . . . 11

Pooling Layer . . . 12

Fully Connected Layer . . . 13

2.6 Semantic Segmentation . . . 13

2.7 FCN Architecture . . . 14

Transpose Convolution . . . 15

Skip Connections . . . 15

2.8 U-Net Architecture . . . 16

Components . . . 16

Advantages . . . 18

3 Implementation 20

3.1 Data Collection . . . 20

Google Earth Engine . . . 20

Landsat 8 . . . 20

Copernicus Global Land Cover dataset 20
3.2 Intersection over Union (IoU) 20

4 Conclusion **21**
4.1 21

References **22**

List of Figures

1.1 Without NDVI 6

1.2 With NDVI 6

2.1 Support Vector Machine 9

2.2 A single neuron. 10

2.3 A two layer neural network with fully connected layers. 10

2.4 The convolution operation performed using a 3x3 filter on a 5x5x1 image. 12

2.5 (1) A typical max pooling layer. (2) The max pooling operation. 13

2.6 AlexNet - an influential NN architecture that popularized the use of CNNs and GPUs to accelerate deep learning. 14

2.7 A 3 × 3 transpose convolution with stride 2 and padding 1. The output gets up-sampled by a factor of 2. 15

2.8 The different variants of the FCN architecture defined in [9], with different skip connections added to the base network. 16

2.9 The addition of skip connections produced finer and more detailed segmentation map. Addition of more such connections gave diminishing returns. 16

2.10 The U-Net architecture. 17

1. Introduction

1.1 About IIRS

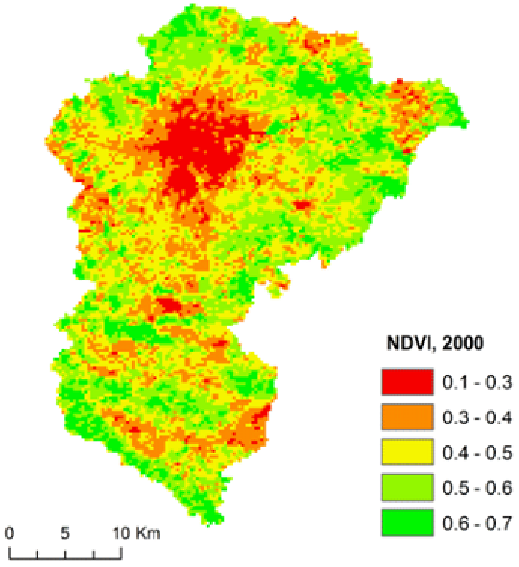
Formerly known as Indian Photo-interpretation Institute (IPI), the Institute was founded on 21st April 1966 under the aegis of Survey of India (SOI). It was established with the collaboration of the Government of The Netherlands on the pattern of Faculty of Geo-Information Science and Earth Observation (ITC) of the University of Twente, The Netherlands. The original idea of setting the Institute came from India's first Prime Minister Pandit Jawahar Lal Nehru during his visit to The Netherlands in 1957. Since its establishment in 1966, IIRS is a key player for training and capacity building in geospatial technology and its applications through training, education and research in Southeast Asia. The training, education and capacity building programmes of the Institute are designed to meet the requirements of Professionals at working levels, fresh graduates, researchers, academia, and decision makers. IIRS is also one of the most sought after Institute for conducting specially designed courses for the officers from Central and State Government Ministries and stakeholder departments for the effective utilization of Earth Observation (EO) data. Keeping pace with the technological advances, the Institute has enhanced its capability with time, to fulfill the increased responsibility and demand from Indian and international community. Today, it has programmes for all levels of users, i.e. mid-career professionals, researchers, academia, fresh graduates and policy makers. The sustained efforts by its dedicated faculty and the management have made the institute remain in the forefront throughout its journey of about four and a half decades from a photo-interpretation institute to an institute of an international stature in the field of remote sensing and geo-information science.[4][5]

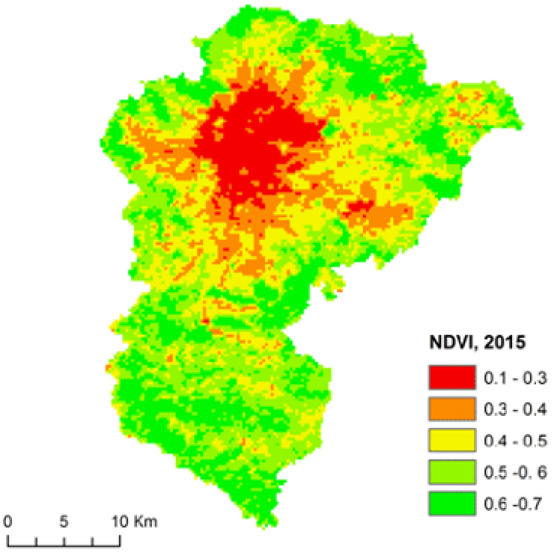
1.2 Remote Sensing

Remote sensing is the acquisition of information about an object or phenomenon without making physical contact with the object and thus in contrast to on-site observation, especially the Earth. Remote sensing is used in numerous fields, including geography, land surveying and most Earth science disciplines (for example, hydrology, ecology, meteorology, oceanography, glaciology, geology); it also has military, intelligence, commercial, economic, planning, and humanitarian applications. 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). Passive sensors gather radiation that is emitted or reflected by the object or surrounding areas. Reflected sunlight is the most common source of radiation measured by passive sensors. Examples of passive remote sensors include film photography, infrared, charge-coupled devices, and radiometers. Active collection, on the other hand, emits energy in order to scan objects and areas whereupon a sensor then detects and measures the radiation that is reflected or backscattered from the target. RADAR and LiDAR are examples of active remote sensing where the time delay between emission and return is measured, establishing the location, speed and direction of an object. [14]

1.3 Land Cover Classification

Land Cover classification generally refers to the categorization or classification of human activities and natural elements on the landscape within a specific time frame based on established scientific and statistical methods of analysis of appropriate source materials.Land cover refers to the surface cover on the ground like vegetation, urban infrastructure, water, bare soil etc. identification of land cover establishes the baseline information for activities like thematic mapping and change detection analysis’https://www.satpalda.com/blogs/sign
of-land-use-land-cover-lulc-maps”





Land cover classification data is of great importance in the following applications: Natural Resource Management: Land cover classification can help policy makers decide how to best manage all the available resources by looking at different terrains. For example government can decide which area requires irrigation facilities the most using land

cover classification.
Wildlife habitat protection: With shrinking habitats of wild animals land cover classification can provide important data for helping in wildlife preservation and protection.
Urban Expansion/ encroachment: Proper planning is required for expansion of urban area and different land cover features need to be studied to minimize the time and economic cost required for tackling natural obstacles to urban expansion.
Damage delineation.(Tornadoes, flooding, fire, volcanic): Live land cover data can provide us information about where a certain natural disaster like flood has occurred/can occur and where has it cause damages.
Target Detection - identification of land strips for use: Using land cover data, identification of ideal locations for different urban construction becomes easy. For example one can decide where exactly it is possible to build an airport or bridge etc.

1.4 Normalized Difference Vegetation Index(NDVI)

Satellite data has become an integral part of modern agricultural management to keep track of crop progress. Some satellite data, however, also can foretell how a crop will perform in the future, offering valuable insights to a wide range of industry players. NDVI is probably the most important of the satellite data. By reading infrared light waves reflected from plants, NDVI can signal stresses to plant health, such as oncoming drought, as much as two weeks before problems are visible to the naked eye. Many agricultural industry participants can benefit from such advance alerts. Farmers can increase irrigation or add crop protection to forestall a pest infestation. And physical traders, processors, and food and beverage companies could seek out alternative supplies, or hedge their positions.[**ndvione**]

NDVI also can reveal positive indications about a crop, providing a heads-up to market participants, logistics companies, and others to prepare for a big harvest. Another type of satellite data, called evapotranspiration (ET), also sends early-warning signals about plants, based on measures of moisture evaporation and transpiration. But ET is available only on a monthly basis, much less frequent than the eight-day reports on NDVIhttps://gro-intelligence.com/insights/articles/ndvi-the-indispensable-data-to-forecast-crop-yields.[**ndvione**]

What is NDVI?

Early space exploration quickly led to atmospheric and meteorological studies. NASA in 1972 launched the Earth Resources Technology Satellite, the forerunner to Landsat, the world's longest-running satellite imagery program. This first satellite was able to distinguish between visible red and near-infrared reflectance bands, which allowed it to identify vegetation, soil, water, and other features.[**ndvione**]

Light from the sun is present as visible light (reds, greens, and blues) and light not visible to our eyes (infrared and ultraviolet). These can be absorbed, transmitted, or reflected by an object. In healthy plants, most of the visible light is absorbed for use in photosynthesis, and much of the near-infrared radiation (NIR) is reflected. However, if the plant is stressed, because of dehydration, for instance, it reflects less NIR and absorbs less light in the visible spectrum, specifically in the red portion, since the plant is not using photosynthesis as efficiently.[**ndvione**]

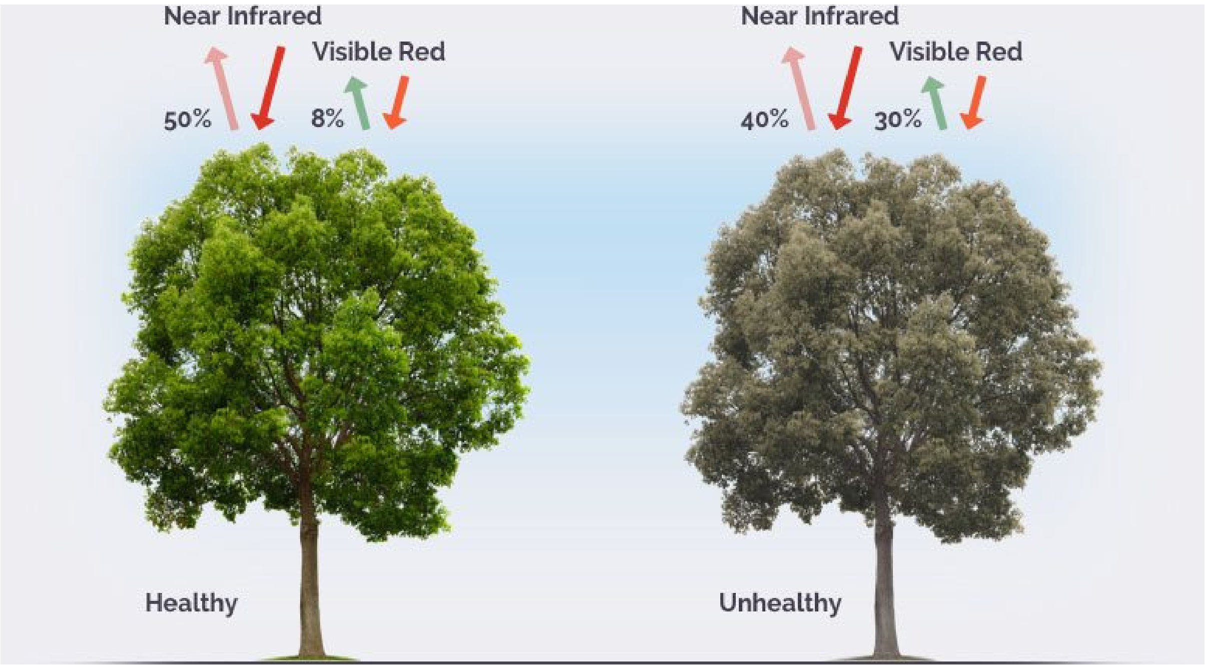
Formula

NDVI is calculated in accordance with the formula:

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

where,
NIR = reflection in the near-infrared spectrum
RED = reflection in the red range of the spectrum.

According to this formula, the density of vegetation (NDVI) at a certain point of the image is equal to the difference in the intensities of reflected light in the red and infrared range divided by the sum of these intensities.
This index defines values from -1.0 to 1.0, basically, representing greens, where negative values are mainly formed from clouds, water and snow, and values close to zero are primarily formed from rocks and bare soil. Very small values (0.1 or less) of the NDVI function correspond to empty areas of rocks, sand or snow. Moderate values (from 0.2 to 0.3) represents shrubs and meadows, while large values (from 0.6 to 0.8) indicate temperate and tropical forests. Crop Monitoring successfully utilizes this scale to show farmers which parts of their fields have dense, moderate, or sparse vegetation at any given moment. Put simply, NDVI is a measure of the state of plant health based on how the plant reflects light at certain frequencies (some waves are absorbed and others are reflected).



Illustration

Chlorophyll (a health indicator) strongly absorbs visible light, and the cellular structure of the leaves strongly reflect near-infrared light. When the plant becomes dehydrated, sick, afflicted with disease, etc., the spongy layer deteriorates, and the plant absorbs more of the near-infrared light, rather than reflecting it. Thus, observing how NIR changes compared to red light provides an accurate indication of the presence of chlorophyll, which correlates with plant health.https://eos.com/ndvi/

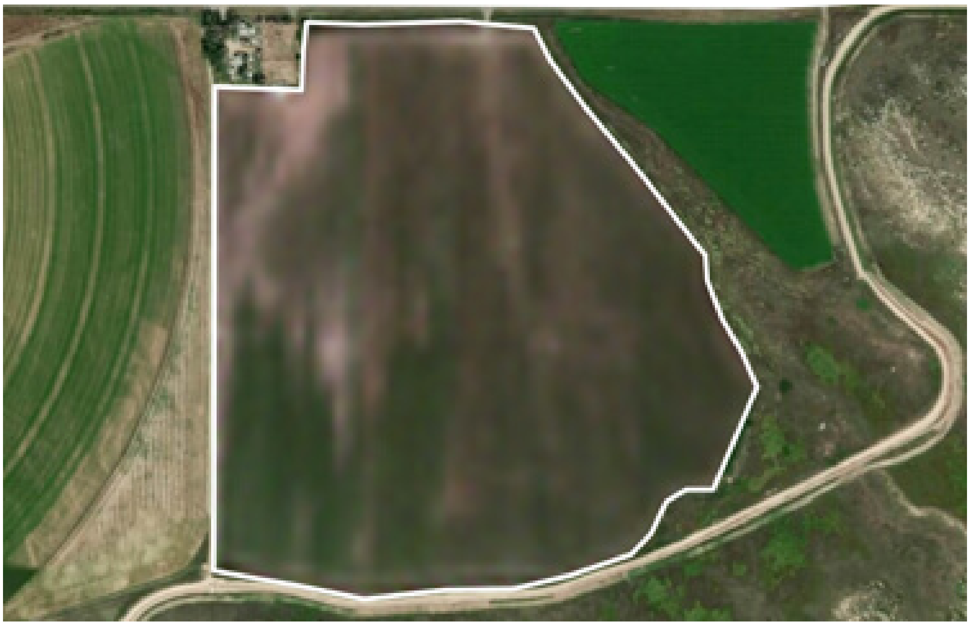


Figure 1.1: Without NDVI

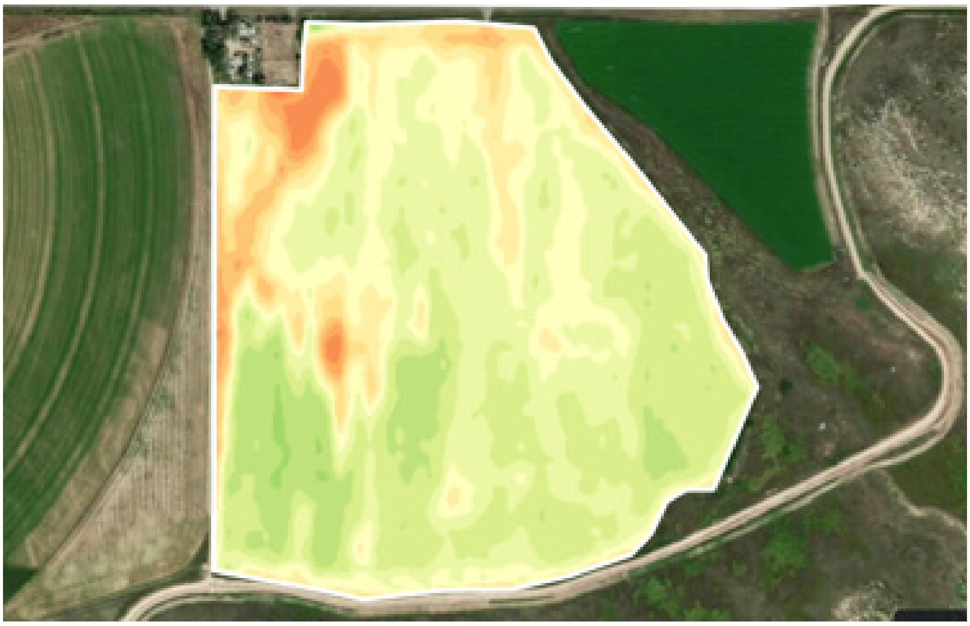


Figure 1.2: With NDVI

2. Methodology

2.1 Image Classification

Image classification is a standard task in computer vision. In general, the image classification problem involves assigning one label out of a given fixed set of discrete labels to the input image on the basis of its visual content. While this is a trivial task for humans, robust image classification is a big challenge for a machine. To the computer, the image is just a grid of numbers which entirely change in unreliable ways with variations in viewpoint, illumination, occlusion, etc. As a result, there is no obvious algorithm which solves this problem. However, a data driven approach of providing the machine with many examples of each class and use of machine learning techniques has shown to be useful.[8]

There are different ways in which these techniques can be applied for classification of satellite imagery.

Pixel Based Approach

In typical satellite images, pixel sizes are generally similar in size to the objects of interest. Most of the methods for image analysis using remote sensing data work on a per-pixel basis. However, with advances in remote sensing technology, the spatial resolution has become finer than the typical objects of interest, leading to an increase in within-class variability.[6]

Object Based Approach

The term "objects" represents meaningful semantic entities or scene components that are distinguishable in an image.[6] This approach involves the partition of the image into meaningful geographical objects that share relatively homogeneous spectral, color, etc.

Semantic Approach

This aims to label each scene image with a specific semantic class. Here, a scene image usually refers to a local image patch manually extracted from large scale remote sensing images that contain explicit semantic classes.[6]

2.2 Supervised Machine Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.[17] Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.[10] If enough data is available then the trained model usually identifies the underlying data patterns and is then able to make predictions on previously unknown input. The training data i.e. the set of input-output example pairs for the purpose of this project is the labelled region of Agra District, Uttar Pradesh, India obtained from Copernicus Global Landcover dataset. The model trained from these input-out put pairs is then tested on the region of Gandhinagar District, Gujarat, India. Supervise learning, however, faces the problem of overfitting when model is made very flexible in order to fit the underlying data and underfitting when the model is to rigid to fit the underlying data.

2.3 Support Vector Machines

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.[18]

There are two kinds of support vector machines: Hard Margin Support Vector Machines and Soft Margin Support Vector Machines. Hard Margin Support Vector machines do not allow misclassification of data points, tend to overfit the data and are thus sensitive to outliers. Soft margin support vector machines allow slight misclassification of data points and penalize for each misclassification. For this project, soft margin support vector machines have been used to classify the data points.

The advantages of using support vector machines are:

Support vectors can deal with high dimensional data easily with the help of kernel transformations.

They are only dependent on some part of the training data known as the support vectors which makes them very memory efficient.

Can be used for a variety of tasks due to availability of a large number of kernel functions for decision making.

Fast and memory efficient implementations in various languages is available via different libraries.

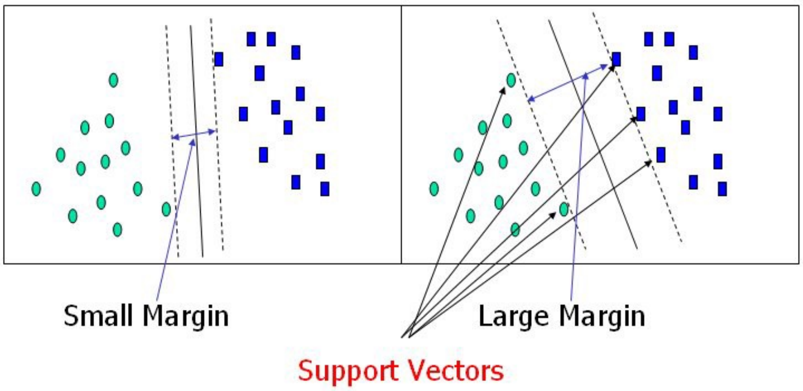


Figure 2.1: Support Vector Machine

The disadvantages of using support vector machines are:
They do not take the spatial orientation of the pixels into account while learning.
With wrong choice of kernel, support vector machines tend to to overfit the data.
Hard margin support vector machines are very sensitive to outliers and hence proper care should be taken while using them.

2.4 Deep Learning and Neural Networks

Application of traditional machine learning techniques requires handcrafted features, developing which demands a considerable amount of engineering skill and domain expertise. This, however, is not true for neural networks, which automatically learn these features from data using a general-purpose learning procedure.[6, 8] Despite having been around for decades, neural networks have garnered much attention only in the last few years on account of the availability of increased computational power and large amounts of data.

A standard neural network consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons. [16] Each neuron can be seen as a single unit applying a non-linear activation function (such as sigmoid, tanh, ReLU) to a linear combination of the input activations to the neuron.[12]

These single neurons can be stacked so that one neuron passes its output as input into the next neuron. The resulting network of neurons can, hence, consist of several

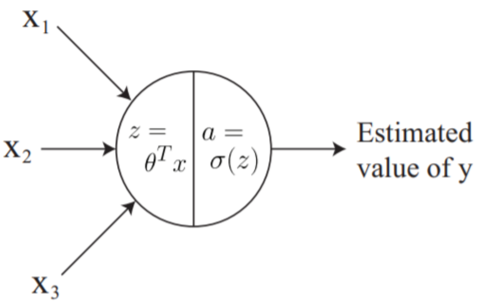


Figure 2.2: A single neuron.
[12]

layers of neurons, each with their own learnable weights and biases. Used in conjunction with an appropriate loss function and optimization algorithm, such a network can be used to learn any complex function, if sufficient data is available for training. Forward propagation through the network yields its prediction for a given input. This prediction is compared with the actual class label, and the loss is computed. Backward Propagation is used to compute the gradients of the loss function with respect to the parameters, which are then used by the optimization algorithm to adjust the parameters and minimize the loss over a number of iterations.

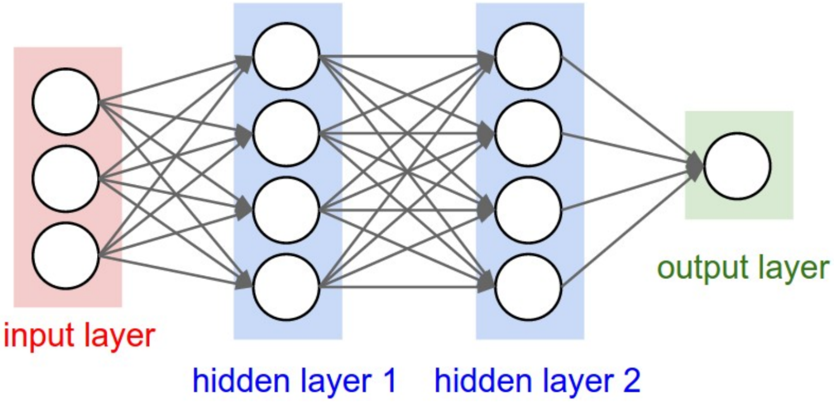


Figure 2.3: A two layer neural network with fully connected layers.
[8]

2.5 Convolutional Neural Networks

Regular neural networks do not scale well to full images. If the input to the neural network is a 200x200 RGB image, the number of weights for each neuron will be 200*200*3 = 120,000 weights. For large networks, the total number of learnable parameters become very large and lead the model to potentially overfit the training data, unless the training set is adequately large.

A convolutional neural network (CNN) is a sequence of layers. Each layers transforms an input volume (images are represented as a three dimensional matrix) of activations to another with some differentiable function which may or may not have parameters.[8, 13] These layers are of three main types:

Convolutional Layer

This is the core building block for convolutional networks. It is based on the convolution operation on images.

Each convolutional layer of a CNN consists of N kernels or filters of a certain volume of neurons sized $f \times f \times d$, with f being the spatial dimension and d being the number of feature channels of the kernel, which is same as the number of channels in the image at its input (D_i). Every one of these filters is convolved with a corresponding volume of the input image, and is slid through the entire image of size $H_i \times W_i \times D_i$. Convolution refers to the summation of the element-wise dot product of the neurons in each filter with the corresponding values in the input, for each position in which the filter is aligned with the image. Based on this notion, a convolution with a single filter at each layer results in a two dimensional output of a certain size. [8, 11, 13]

The intervals with which the filter moves in each spatial dimension is decided by the **stride** s. In order to prevent undue shrinkage of the volume along the spatial dimension, the image can be padded with pixels along the outer edges. The width of this **padding**, in pixels, is given by another hyper-parameter, p.[13, 11] The convolution operation is repeated for each of the N filters, and the resulting N activation maps are stacked together across the third dimension giving an output volume of dimensions:

$$H_o = \frac{H_i - f + 2p}{s} + 1$$

$$W_o = \frac{W_i - f + 2p}{s} + 1$$

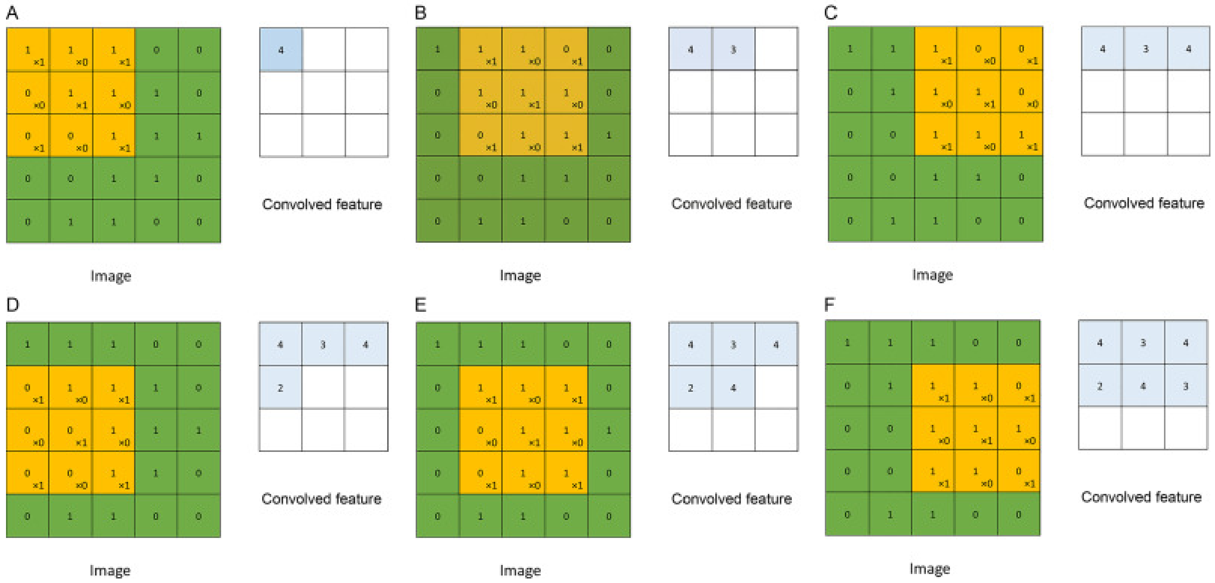


Figure 2.4: The convolution operation performed using a 3x3 filter on a 5x5x1 image. [1]

$$D_o = N$$

In a convolutional layer, each neuron is connected to only a local region of the input volume, called the receptive field of that neuron. The extent of connectivity is limited to the filter size along the spatial dimensions, but is full along the depth axis.[8] It should also be noted that all activations belonging to a particular channel in the output volume correspond to a single filter applied on the input volume, and hence depend on the same shared parameters. Local connectivity and parameter sharing not only help reduce the number of learnable parameters, but also make the CNN good at capturing **translation invariance**. This makes them an ideal choice for the image classification problem.

Pooling Layer

It is common to periodically insert a Pooling layer in-between successive convolutional layers in a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting.[8] The most common form of pooling layer in CNN architectures employs filters of size 2 × 2 with a stride of 2, taking a max over 4 cells of the input image. It is worth noting that while this halves the width and height of the image, the depth remains unaffected as the max operation is applied independently to each channel of the image. This down-sampling process effectively discards 75% of the activations.

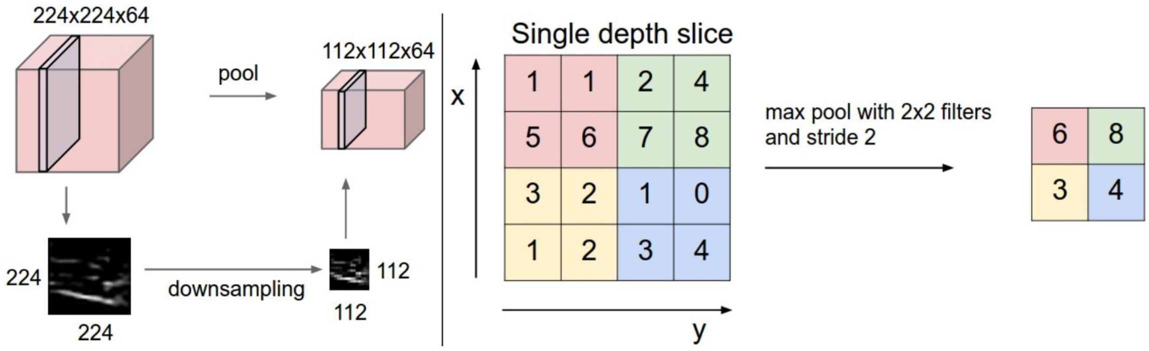


Figure 2.5: (1) A typical max pooling layer. (2) The max pooling operation. [8]

Fully Connected Layer

Once higher level features are detected from the preceding convolution and pooling layers, a fully connected layer is usually attached at the end of the network. This layer is fully connected to all activations in the previous layer, as in regular neural networks, allowing all the features learned by the network to be taken into account by the output layer.

In practice, fully connected layers have an equivalent representation as a convolutional layer having N filters with dimensions equal to those of the input image. The output of this layer will thus be a volume of dimensions $1 \times 1 \times N$. This simple change allows the same CNN to be applied on images with arbitrary spatial dimensions and classify them in a single pass of forward propagation, instead of iterating on different crops of adequate size. This is the basic intuition behind **Fully Convolutional Networks**.[9]

All of these different types of layers can be stacked together in various ways to form a CNN.

2.6 Semantic Segmentation

Typical CNNs used for image classification take fixed-size inputs and produce non-spatial outputs (predicted class labels). However, in case of land cover classification, the objective is not to assign a single class label to a given satellite image, but to divide the given image into segments each corresponding to one class out of a given set of land cover classes. This problem is known as semantic segmentation. There are different ways in which the existing image classification CNNs can be extended to solve this problem.

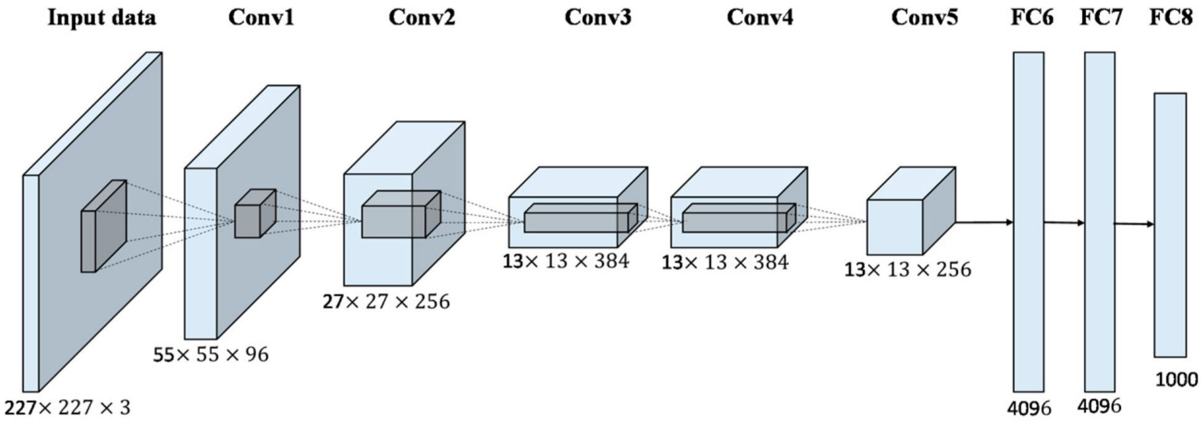


Figure 2.6: AlexNet - an influential NN architecture that popularized the use of CNNs and GPUs to accelerate deep learning.
[7, 3]

An obvious approach is to use a sliding window to extract small local patches or crops from the input image. These crops are then passed to an image classifier which assigns a class to the central pixel for each crop. In this way, we can classify all pixels of the original image. However, this approach is computationally very expensive as a separate forward pass (through the CNN) will be required to classify every pixel, and so it is not practically feasible. Moreover, as the local patches overlap, much of the computation is redundant.
Another approach is to use fully convolutional networks with only convolutional layers. In these networks, zero padding can be used to preserve the spatial dimensions of the input image. So, the network can be trained by comparing the spatial output to the ground truth land cover data. However, the large spatial dimensions throughout the network make the model computationally infeasible. [9, 8, 15]

While the problem requires the CNN to produce a spatial output, down-sampling is essential for reducing the need for computational resources to a feasible level. As a result, the most popular approach is to design the network as a combination of an encoder(down-sampling path) and a decoder(up-sampling path).

2.7 FCN Architecture

As discussed earlier, the re-interpretation of fully connected layers as convolutional layers yields Fully Convolutional Networks. These networks can take input of any dimensions and output classification maps. For up-sampling these classification maps, *Long et al.*[9]

propose the use of transpose convolution.

Transpose Convolution

Ordinary convolution involves taking a dot product between the filter and the input for every receptive field at a stride of s in the input, and storing the result at a stride of 1 in the output. The resulting output image gets down-sampled by a factor of s. In transpose convolution, we use the values at a stride of 1 in the input and take their scalar product with the filter, storing the resulting matrix at a stride of s in the output. As a result the image gets up-sampled by a factor of s. Any overlapping values in the output are summed together.[9, 8] A stack of layers performing transpose convolution with an appropriate activation function can be used to learn a non-linear up-sampling.[9]

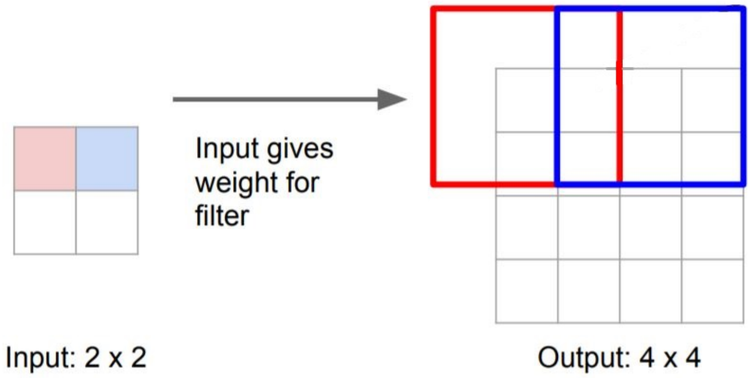


Figure 2.7: A 3 × 3 transpose convolution with stride 2 and padding 1. The output gets up-sampled by a factor of 2.
[8]

Skip Connections

Semantic segmentation faces an inherent tension between semantics and location. Deep layers capture local information and resolve the semantic information that the image contains. Global information available to shallower layers can be used to capture the location. This motivates the idea of adding skip connections from shallow layers to the final prediction layer. Combining the information captured by shallow and deep layers at the time of up-sampling lets the model make local predictions that take into account global structure and semantics. To combine activations with different spatial dimensions, they are first up-sampled individually as required using transpose convolutions and then summed up. The result is then up-sampled by a final layer, so that the output spatial dimensions match those of the input image.[9]

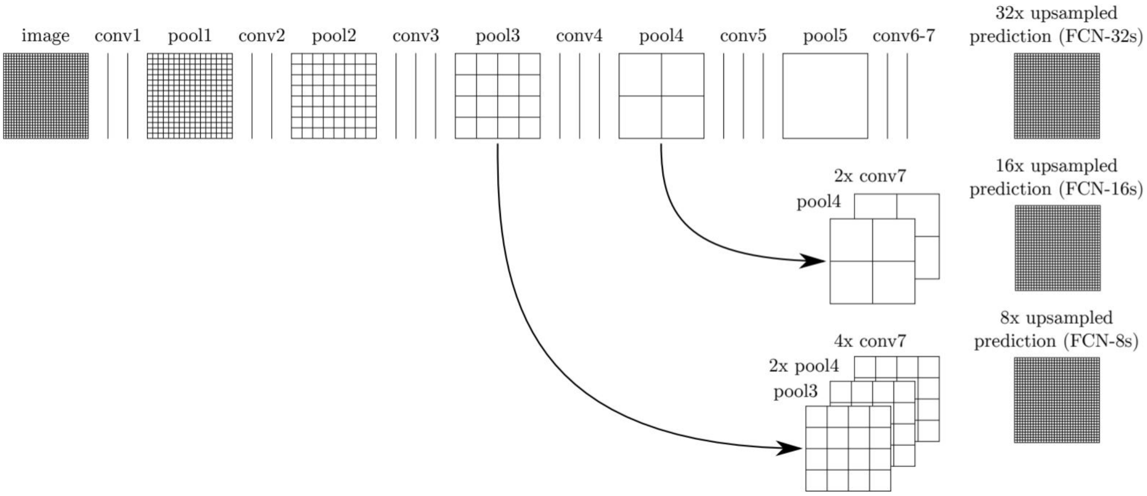


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

Figure 2.8: The different variants of the FCN architecture defined in [9], with different skip connections added to the base network.

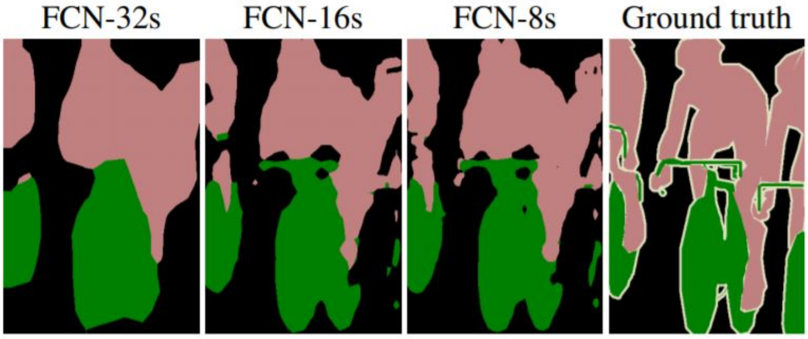


Figure 2.9: The addition of skip connections produced finer and more detailed segmentation map. Addition of more such connections gave diminishing returns.
[9]

2.8 U-Net Architecture

Proposed by *Ronneberger et al.* [15], the U-Net architecture builds upon and extends the FCN architecture for more precise segmentation using a smaller training dataset. While the architecture was designed particularly for bio-medical segmentation applications, it has shown to work well for other domains too.

Components

The U-Net architecture can be divided into three major parts:

1. The down-sampling path

2. The bottleneck

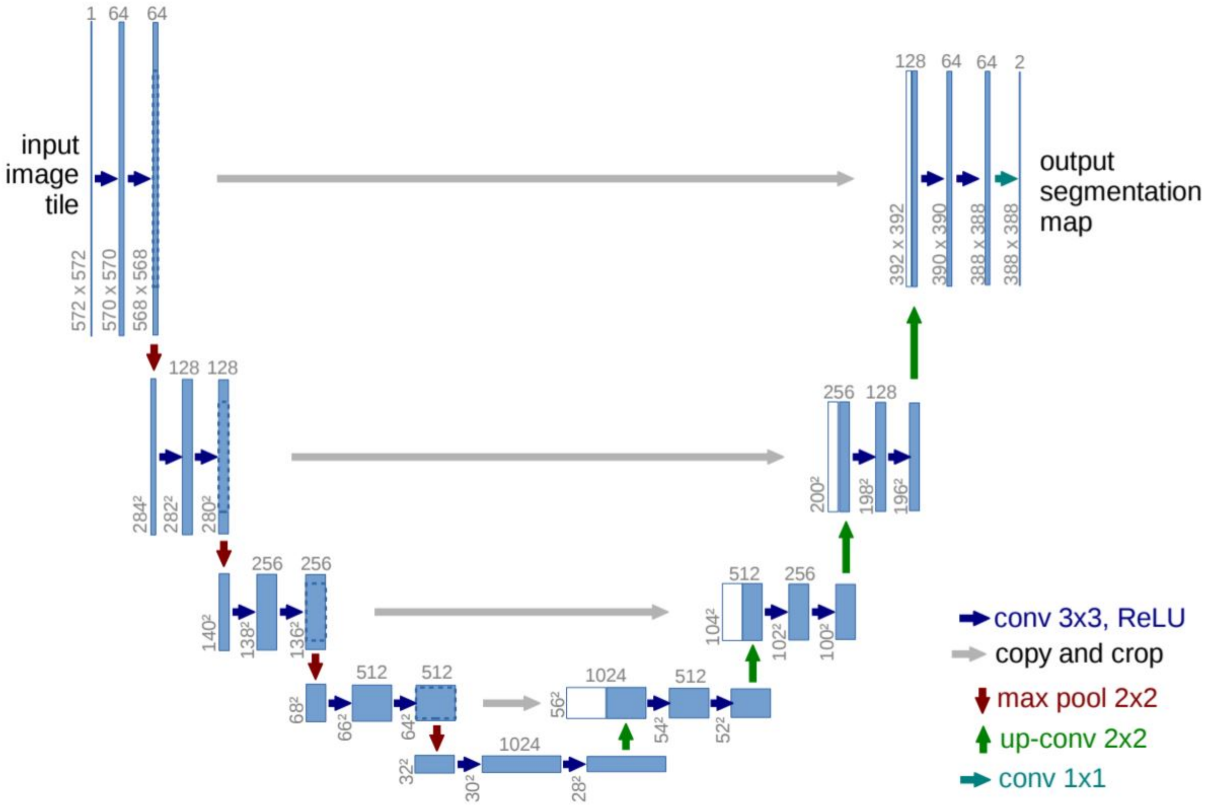


Figure 2.10: The U-Net architecture.
[15]

3. The up-sampling path

Down-sampling Path

It is made up of four identical blocks. Each block encloses the following sequence of layers:

1. Convolutional layer ($f = 3, s = 1, p = 0$)
2. Convolutional layer ($f = 3, s = 1, p = 0$)
3. Max Pooling layer ($f = 2, s = 2$)

At each down-sampling step (max-pooling), the number of channels is doubled, starting with 64 channels in the first block. The down-sampling path captures not only features relevant to classification, but also contextual information about the location of different segments.

Bottleneck

It is made up of 2 layers:

1. Convolutional layer ($f = 3, s = 1, p = 0$)
2. Convolutional layer ($f = 3, s = 1, p = 0$)

Up-sampling Path

It is made up of four identical blocks. Each block encloses the following sequence of layers:

1. Up-sampling transpose convolution layer ($f = 2, s = 2$)
2. Concatenation with appropriately cropped feature map from down-sampling path
3. Convolutional layer ($f = 3, s = 1, p = 0$)
4. Convolutional layer ($f = 3, s = 1, p = 0$)

At each up-sampling step (transpose convolution), the number of channels is halved. The output volume after up-sampling is concatenated with the corresponding volume from the down-sampling path after adequate cropping. The introduction of these skip connections allows the propagation of context information about the localization from the down-sampling path to the up-sampling path which already carries semantic information. This helps the network to spread out activations correctly when up-sampling and lends the network its U-shape.

Advantages

The U-Net architecture presented above has several features which lend it better performance than the FCN architecture.

In the FCN architecture, the single up-sampling layer is added at the end of the CNN and has limited number of channels (equal to the number of class labels). The U-Net architecture makes use of multiple up-sampling layers (each with multiple feature channels) in the up-sampling path, interspersed with blocks of convolutional layers. This provides for the opportunity to add more skip connections and so, to fuse contextual and semantic information better, at the time of up-sampling.

Concatenation is used for combining information from skip connections, as opposed to the summing operation used in the FCN architecture. This helps to retain more information from both sets of activations, without polluting either.

The U-Net architecture is also particularly well suited to applications where labelled training data is not available in abundance. This makes it more attractive for our project on account of the limited computational resources and time available for training the network. We, hence, propose to use this CNN architecture for the problem of land cover classification.

3. Implementation

3.1 Data Collection

Google Earth Engine

Google Earth Engine is a computing platform that allows users to run geo-spatial analysis on Google's cloud infrastructure. It provides several ways to interact with the platform. The Code Editor is a web-based Integrated Development Environment (IDE) for writing and running scripts. The Explorer is a lightweight web app for exploring the data catalog and running simple analyses. The client libraries provide Python and JavaScript wrappers around the web Application Programming Interface (API). The platform provides easy to use tools for handling, visualizing, and extracting large amounts of remote sensing data.[2]

Landsat 8

Copernicus Global Land Cover dataset

3.2 Intersection over Union (IoU)

4. Conclusion

4.1

Information of land cover classification is very crucial for applications like natural resource management, wildlife habitat protection, urban expansion, damage delineation, target land detection etc. With developments in technology, conventional method of field survey which is both inaccurate and time consuming has been replaced by remotely sensed imagery analysis which is highly accurate and covers wider range. Indian Space Research Organisation (ISRO) and Indian Institute of Remote Sensing (IIRS), Dehradun play an important role in providing information for these applications. Modern statistical and computing techniques like machine learning and deep learning minimize the human intervention in the task of analysis and classification, thus making the outcome very accurate and the process very agile. Support vector machines efficiently classify the pixel data however they do not take the spatial orientation of the pixels into account. Convolutional Neural Network is another efficient algorithm that can be used for classification however it fails to account for the temporal nature of the data.

References

[1] Ehsan Fathi and Babak Maleki Shoja. “Deep neural networks for natural language processing”. In: *Handbook of statistics*. Vol. 38. Elsevier, 2018, pp. 229–316. (Visited on 06/02/2020).

[2] Google. *Google Earth Engine - Platform Overview*. URL: `https://earthengine.google.com/platform/` (visited on 06/04/2020).

[3] Xiaobing Han et al. “Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification”. In: *Remote Sensing* 9.8 (2017), p. 848.

[4] *History—Indian Institute of Remote Sensing*. URL: `https://www.iirs.gov.in/historyandobjectives` (visited on 06/01/2020).

[5] *Indian Institute of Remote Sensing*. URL: `https://www.iirs.gov.in/institute-profile` (visited on 06/01/2020).

[6] Manideep Kolla, Aniket Mandle, and Apoorva Kumar. *Eye in the sky*. GitHub Page. 2018. URL: `https://github.com/manideep2510/eye-in-the-sky` (visited on 06/01/2020).

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[8] Fei-Fei Li, Justin Johnson, and Serena Yeung. *cs231n - Lecture Slides*. 2017. URL: `http://cs231n.stanford.edu/slides/2017/` (visited on 06/01/2020).

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[10] Ameet Talwalkar Mehryar Mohri Afshin Rostamizadeh. In: *Foundations of Machine Learning,* 2012.

[11] Shivaprakash Muruganandham. *Semantic segmentation of satellite images using deep learning*. 2016.

[12] Andrew Ng and Kian Katanforoosh. *cs229 - Lecture Notes*. 2018. URL: `http://cs229.stanford.edu/notes/` (visited on 06/01/2020).

[13] Andrew Ng, Kian Katanforoosh, and Younes Bensouda Mourri. *Convolutional Neural Networks*. 2017. (Visited on 06/02/2020).

[14] *Remote Sensing - Wikipedia*. URL: https://en.wikipedia.org/wiki/Remote_sensing (visited on 06/01/2020).

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[16] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[17] Peter Norvig Stuart J. Russell. In: *Artificial Intelligence: A Modern Approach, Third Edition*. 2010.

[18] Savan patel. *Chapter 2: SVM(Support Vector Machines)*. 2017. (Visited on 06/05/2020).