Land Cover Classification on Hyperspectral Imagery using CNN-SVM Hybrid

Abhishek Dangayach[1], Aman Kumar[2], Garima Sharma(Assistant Professor)[3]

[1-3]Dept. of Computer Science & Engineering, GIT, Jaipur-India

---

Abstract: Accurate land cover classification is very important for environmental sustainability. Manual classification of land cover is a very difficult task because it is not possible to classify every place on earth. Most of the existing land cover approaches are pixel-based single-date multi-spectral images using conventional techniques like random forests (RFs), neural networks (NNs), and support vector machines (SVMs). They all provide classification with an unacceptable accuracy

Nowadays, there is a significant improvement in satellite systems and sensors that acquire data with improved spectral, spatial, radiometric, and temporal characteristics. Recently, several major satellites remotely sensed datasets become more affordable and feasible with much higher spatial resolutions acquired by the Landsat systems, ASTER (Advanced Space-borne Thermal Emission and Reflection Radiometer), and Sentinels, exploiting multi-temporal information for land cover classification. However, working with such higher-resolution multi-temporal, multi-spectral imagery datasets is facing some crucial challenges mainly caused by the frequent occurrences of pixels contaminated by clouds or shadows.

In the last decade, there is rapid development in deep learning algorithms like CNN. Compared to traditional classifiers like random forest and support vector machine, DCNN does not need extraction and selection of hand-crafted features. This motivated the researchers in the remote sensing community to investigate its usefulness for remote sensing image analysis. Moreover, convolutional neural networks (CNN) are not considered to be fully connected neural nets. CNN's have convolution and pooling layers and fully connected layers, whereas ANN has only fully connected layers, which is a key difference.

Therefore, in this project, the focus is to exploit multispectral remote sensing data by using a hybrid model based on convolutional neural networks and support vector machine to have land cover classification with improved accuracy. A system is designed that classifies the image using SVM based on the features extracted from the last layer of CNN. This hybrid model results in greater accuracy than CNN or SVM.

1. Introduction

The Physical material present on the surface of the earth is known as Land Cover. Land covers include grass, trees, water, crops, etc. The process to capture information is important to identify the region each land cover holds. The process of capturing information on the land cover can be done in two ways Remote Sensing, Field Survey. The methodology used in this paper to classify land cover is based on remote sensing. Remote sensing is the science of obtaining information about objects or areas from a distance, typically from aircraft or satellites. The information collected is the amount of data reflected by the earth's surface.

The collection of data is done by remote sensors. Several types of sensors are used to collect the data some are LANDSAT, MODIS, etc. These Sensors transmit the electromagnetic waves to the surface of the earth and then collect the number of waves that are reflected. These sensors transmit waves in different continuous ranges of the electromagnetic spectrum known as bands. Each band is represented by its wavelength and bandwidth. The scale of each band is grey-scaled. All these bands make a satellite image. The information of these bands can be used for the identification of materials on the surface of the earth. In this project, these images are used for land cover classification. These bands are used to calculate indices such as ENVI, NDWI, NDVI, etc. These indices are used to define the land cover by using machine learning algorithms. These indices are applied on pixel-wise or on a certain area of the satellite image. These indices are widely used in the field of crop type classification.

Many machine learning algorithms are used for hyperspectral image classification. Large dimensions/bands can be a problem for the classification problem. Therefore, Dimensionality Reduction (DR) is used to reduce the dimension of HSI. In this project, the Principal Component Analysis (PCA) algorithm is used to extract the best features. Several kinds of research were conducted on the Hyperspectral images using SVM for classification purposes. But, currently, there are so many algorithms available that can classify HSI with better accuracy than SVM. Newly researched deep learning models (Ex:-CNN) also results in better accuracy than SVM. CNN uses feature maps and applies Neural Networks on them.

In this project, a CNN-SVM based hybrid model is used which uses the feature maps extracted by CNN layers and then classifies them using SVM. The features extracted by CNN results better because CNN generates all possible maps by applying different filters and select only better features. This results in better accuracy than SVM or even CNN. This model is applied to the INDIAN-PINES dataset.

The result with increasing accuracy can be seen as the accuracy obtained by SVM is 83.67(%), the accuracy obtained by only CNN is 85.48(%) and the accuracy obtained by the hybrid model of CNN-SVM is 92.31(%).

2. Dataset Description

The dataset used in this project is the **Indian Pines Hyper-Spectral Dataset.** Indian pines dataset contains images with mainly two resolutions.

- ***Spatial Resolution****:* - It shows how many areas a pixel in image occupies. A spatial resolution of 10m means the 1-pixel cover ground area of 10m x 10m.
- ***Spectral Resolution:*** **-** Spectral resolution describes the ability of a sensor to define fine wavelength intervals. The finer the spectral resolution, the narrower the wavelength ranges for a particular channel or band.

Hyperspectral imagery with a resolution of M x N x B can be seen as follows:

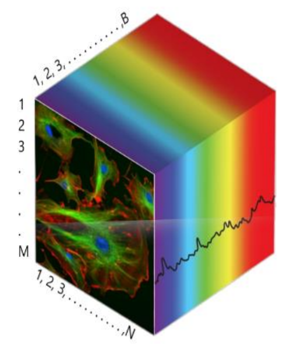


Fig: **-** Hyper-Spectral image with dimensions N x M x B

The Indian pines dataset contains 145x145 pixels with 200 bands and ground truth resolution of 20m. Each pixel of a band corresponds to the spectral characteristics of a particular area in the frequency range corresponding to the sensor range. This dataset contains a total of 16 classes. The ground truth contains a total of 10249 pixels. The pixels in the ground truth corresponding to these classes are not evenly distributed. So, we randomly select 75% of these pixels for training purposes and the remaining 25% for testing these pixels on the machine learning model. These pixels are firstly scaled in the range from [-1, 1]. After that patches are created using the ground truth corresponding to each pixel. The pixel distribution in Indian Pines is as follows:

#	Class	Samples	No. of Train Samples	No. of Test Samples
1	Alfalfa	46	35	11
2	Corn-notill	1428	1071	357
3	Corn-mintill	830	622	208
4	Corn	237	178	59
5	Grass-pasture	483	362	121
6	Grass-trees	730	547	183
7	Grass-pasture-m	28	21	7
8	Hay-windrowed	478	358	120
9	Oats	20	15	5
10	Soybean-notill	972	729	243
11	Soybean-mintill	2455	1841	614
12	Soybean-clean	593	445	148
13	Wheat	205	154	51
14	Woods	1265	949	316
15	Buildings-Grass-	386	289	97
16	Stone-Steel-Tow	93	70	23
	TOTAL	10249	7686	2563

In this dataset bands that are affected by atmospheric absorption are removed. So, the training is done only on 200 bands of the image.

3. Classification Algorithms

There are two classification algorithms used in this project. First, we use CNN (Convolutional Neural Network); this algorithm is used to extract the important features needed for classification. Then the features are extracted before the last layer. These features are then used to train the dataset on the second model. The second model used for the final classification is SVM (Support Vector Machine). SVM trains dataset based on features from CNN. Detailed explanations of these algorithms are given as follows:

- **CNN (Convolutional Neural Networks)**

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that takes an image as an input and then can classify them by assigning labels accordingly. It is a type of feed-forward artificial neural network. The pre-processing required in CNN is lower as compared to other classification algorithms. CNN has wide applications in the field of natural language processing, audio and video recognition, recommender systems, etc.

CNN is comprised of one or more layers. CNN is generally made of two types of layers: convolutional layers, pooling layers. These layers generate random patches corresponding to the input patch. This patch is passed to the convolution layer to generate feature maps. These feature maps can be pooled or can be used to generate even more feature maps. Generally, the activation function used in the convolution layer is **ReLU Activation Function.**

$$y = \max(0, x)$$

After the preprocessing phase (generation of feature maps and classification) is done, these features can be used to classify the patches. Generally, ANN (Artificial Neural Networks) is used as a classifier in the last layer of the CNN model for classification. The basic architecture of CNN can be seen as:

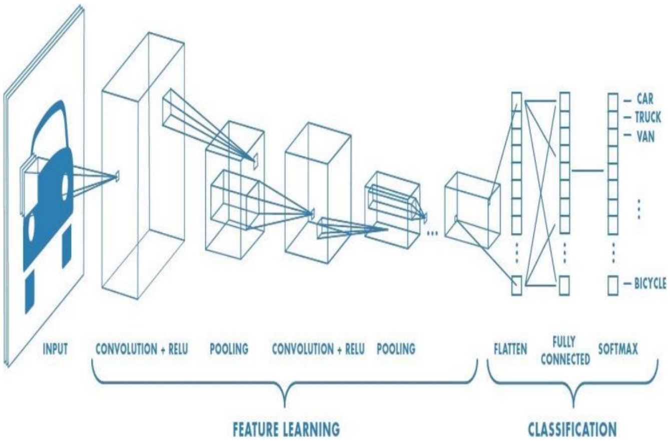


Fig: - The Convolutional Neural Network (CNN) architecture

Layers of CNN can be explained as:

1. **Input layer: -** This is the first layer of CNN which takes an image as an input. In this case of the colored image, the size of the input is M x N x 3, where 3 represents no of channels/bands (R, G, and B).

2. **Convolutional Layer: -** The element involves in carrying the convolution operation in this layer is known as kernel/filter. The main objective of this layer is to extract high-level features such as edges etc. This layer applies the kernel to generate the feature maps. These maps then again pooled to extract some high-level features.

3. **Pooling Layer: -** This layer works almost the same as the convolution layer to decrease the spatial size of the convolved feature. This reduction is done to reduce the computational power required to process the data. Furthermore, it is useful for extracting dominant features that are rotational and positional invariant, thus maintaining the process of effectively training of the model. The most common pooling done is max pooling. In this pooling, the max value is taken from the portion of the image on which filter is applied.

4. **Flatten Layer: -** This layer is used to flatten the resulted feature maps from the max-pooling layer to provide input to the fully connected layer.

5. **Fully Connected Layer: -** This layer is used to train the model on input data by extracting dominating features for that image. This layer provides the input flattened by the flattening layer to the feed-forward neural network and the backpropagation is applied to every image. After training of the model is done images taken by the input layer can be classified easily using the weights obtained after training. For dividing or classifying this image the **Softmax function** is used which divides the image according to the probability of which class it belongs.

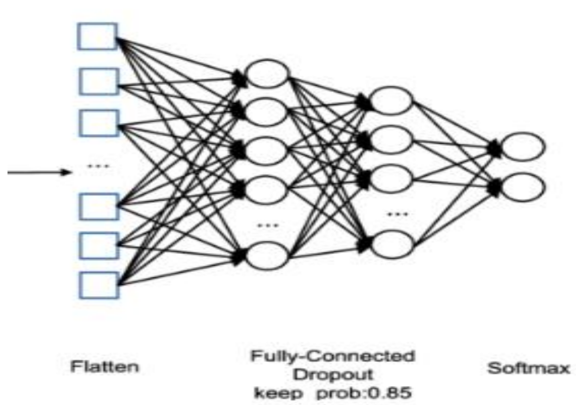
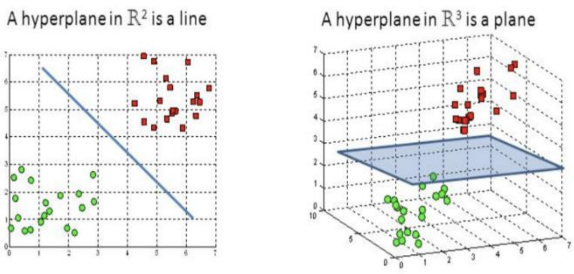


Fig: - Fully Connected layer with input from pooling layer

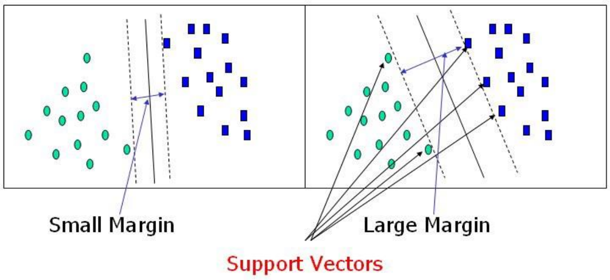
- **Support Vector Machine(SVM)**

SVM is a type of classifier that is used to classify an input in a certain class. The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space.

(N - the number of features) that distinctly classifies the data points. Hyperplanes are decision boundaries that help in classify input. The area on either side of a hyperplane can be considered as a different class. Support vectors are the data points that are closer to other points. Support vectors help in defining the position and orientation of the hyperplane. A hyperplane can be seen as:



(a)



(b)

Fig: - (a) Hyperplane to divide two classes, (b) Support Vectors

4. Methodology

Classification of land cover is a very crucial task. But, recently many works are carrying out on this topic. So, hyperspectral images are used for classification. The classification can be done by applying machine learning models to images. Since hyperspectral images are large in dimensions and can have large no of bands, this can make the training of image a lot costly. Normal computers with no GPU or GPU with less computing power cannot work effectively for this purpose because training can take a lot more time. So, for training purposes, we need a GPU with at least 20GB RAM. The model used in this project is a hybrid of CNN and SVM. The methodology used in this project can be seen with the help of the chart as follows:

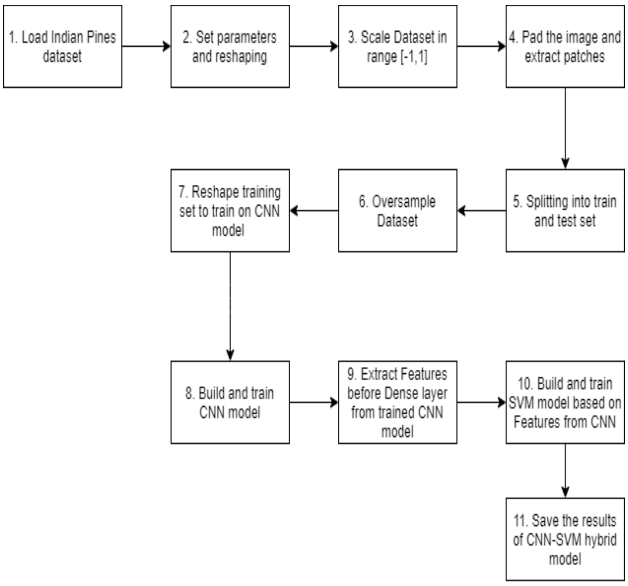


Fig: - Flow chart of CNN-SVM hybrid model

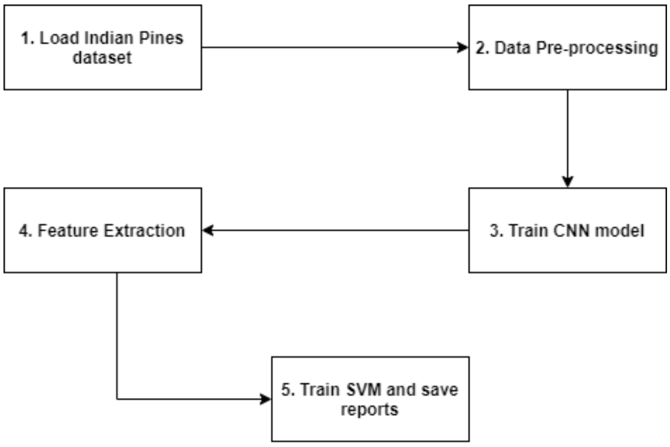


Fig: - Brief of CNN-SVM hybrid

In this project, a hybrid model of CNN and SVM is used. First, a CNN model is built and trains then features are extracted, and then the SVM model is trained. The steps can be seen in the above flowchart. These steps are divided into 5 parts:

1) **Import dataset:** In this step, the Indian pines dataset is loaded to train and built our model.
2) **Data pre-processing:**

    1. **Defining parameters:** In this step, parameters are decided to pre-process the dataset. Like height, width, no. of bands, oversampling, etc.
       Parameters used are: height=145, width=145, bands=200
    2. **Reshaping Dataset:** In this step, the dataset is reshaped and converted into a NumPy array. All other steps are performed on this array.
    3. **Scaling:** Above created NumPy array is scaled into a range of [-1, 1] because a hyperspectral image contains more than one band and each band may have a different range of values.
    4. **Pad the image:** In this step, the image is padded in all sides with values. These values can be zero or cornered values. This is done because the patch-size can be greater than 1. And to extract patch corresponding to each pixel padding must be done.
    5. **Extract Patches:** Patches are extracted and for each patch, its ground-truth value is matched with the center pixel of the patch.

       Ex: Let the patch-size=3 x 3
       Then after padding shape will be 146 x 146
       So, in $padded\_x[\mathrm{x}][\mathrm{y}]$; $0 \leq x < 3 \; and \; 0 \leq y < 3$
       $padded\_x[1][1]$ Corresponds to $ground\_truth[0][0]$

6. **Splitting into train and test set:** After extracting patches the dataset is divided into training and test set. The distributed dataset contains only those pixels for which ground-truth value is available. The total no of pixels with ground truth available is 10249. The no samples in the training set are 7686 and no. of samples in the test set is 2563 with test ratio is 0.25.
7. **Oversampling:** This is done if no of samples for a certain class is much lower than the max no of samples of any class. Then to reduce the difference between no. of samples of classes oversampling is done. This process creates new patches by making copies, rotate, etc. previous samples.

3) **Build and Train CNN Model**

   1. **Building CNN model:** In this step, the CNN model is built. This model is built because recent works prove that CNN results in a greater change in accuracy than SVM. In this project, CNN is used just only for feature extraction.
   2. **Training CNN model:** After the Successful building of the CNN model its training is done on the dataset. This training is done for feature extraction.

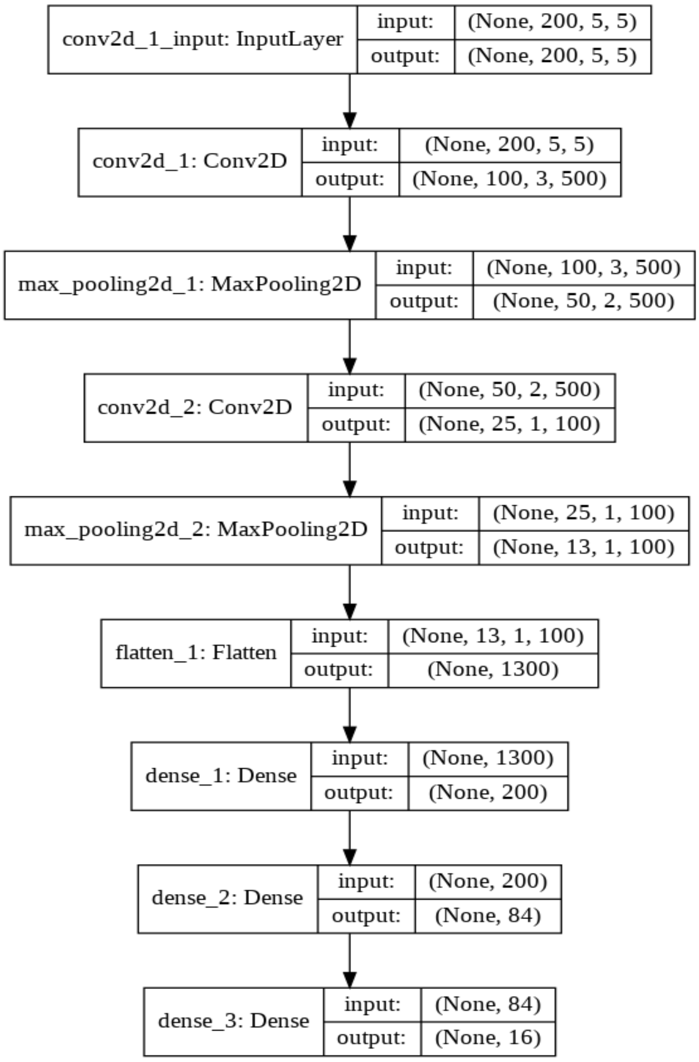
4) **Feature Extraction:** In this step features from the layer before fully-connected are extracted. This step is done because when our model trains on CNN it selected the best features by itself that can result in a better classification when passed to MLP (Multi-level Perceptron). So, the best features selected by CNN can also be passed to SVM rather than MLP.

5) **Build and train the SVM model:** Features from CNN can be passed to any classifier. In this project, SVM is used as a classifier while CNN is used as a feature extractor. Based on the features from CNN, the SVM model is built and trained.
6) **Save Results:** After Training is done various reports are saved.

   1. Accuracy of Only SVM on test samples
   2. Accuracy of Only CNN on test samples
   3. Accuracy of CNN-SVM hybrid model on test samples
   4. Overall Accuracy Comparison

After this step, it is found that the CNN-SVM hybrid results in greater accuracy than both of these models.

The overall CNN model architecture is:



5. Results

The training of the INDIAN PINES dataset on the hybrid model of CNN and SVM results better as compared to either of them. All the models are trained over 7686 samples with available ground truth and tested over 2563 samples and the predicted results are compared with ground truth values corresponding to these pixels. The accuracy can be seen as:

No. of train samples: 7686

No. of test samples: 2563

Accuracy on Test Samples Using Only CNN: 85.48575639724731 (%)

Accuracy on Test Samples Using Only SVM: 83.67479674796748 (%)

Accuracy on Test Samples Using CNN-SVM: 92.31369488880219 (%)

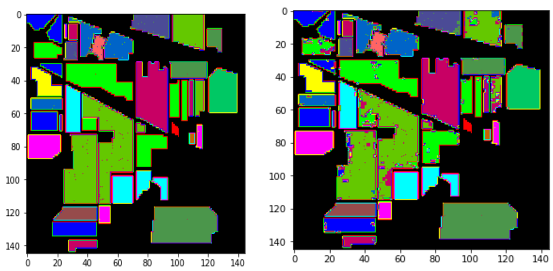
```
Classification result:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       0.88      0.88      0.88       357
           2       0.87      0.95      0.91       208
           3       0.95      1.00      0.98        59
           4       1.00      0.97      0.98       121
           5       0.99      0.99      0.99       183
           6       0.70      1.00      0.82         7
           7       0.99      1.00      1.00       120
           8       1.00      1.00      1.00         5
           9       0.84      0.95      0.89       243
          10       0.95      0.85      0.90       614
          11       0.89      0.93      0.91       148
          12       1.00      1.00      1.00        51
          13       0.98      0.94      0.96       316
          14       0.81      0.95      0.88        97
          15       1.00      1.00      1.00        23

    accuracy                           0.92      2563
   macro avg       0.93      0.96      0.94      2563
weighted avg       0.93      0.92      0.92      2563
```

(a)

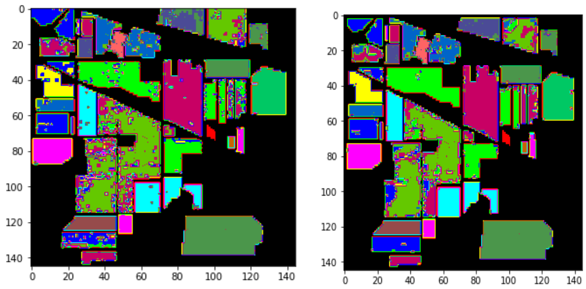
```
confusion matrix:
[[ 11   0   0   0   0   0   0   0   0   0
 [  0 313  11   1   0   0   0   0   0   8
 [  0   0 198   0   0   0   0   0   0   3
 [  0   0   0  59   0   0   0   0   0   0
 [  0   0   0   0 117   0   3   1   0   0
 [  0   0   0   0   0 181   0   0   0   0
 [  0   0   0   0   0   0   7   0   0   0
 [  0   0   0   0   0   0   0 120   0   0
 [  0   0   0   0   0   0   0   0   5   0
 [  0   3   1   0   0   0   0   0   0 232
 [  0  34  16   2   0   1   0   0   0  32 5
 [  0   5   1   0   0   0   0   0   0   2
 [  0   0   0   0   0   0   0   0   0   0
 [  0   0   0   0   0   0   0   0   0   0
 [  0   0   0   0   0   0   0   0   0   0
 [  0   0   0   0   0   0   0   0   0   0
```

(b)

Fig: - (a) Classification report, (b) Confusion Matrix of CNN-SVM on test samples.



(a) (b)



(c) (d)

Fig: - (a) Indian pines ground truth, (b) predicted CNN-SVM hybrid image, (c) Predicted SVM image, (d) Predicted CNN image

From the above figures, it can be easily seen that CNN-SVM hybrid results in better accuracy than both of them.

6. References

a. https://medium.com/nybles/a-brief-guide-to-convolutional-neural-network-cnn-642f4 7e88ed4
b. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72
c. https://ieeexplore.ieee.org/document/7729859
d. https://ieeexplore.ieee.org/abstract/document/1293752
e. https://ieeexplore.ieee.org/document/6626036
f. https://ieeexplore.ieee.org/document/8659654
g. https://medium.com/@openweathermap/satellite-imagery-landsat-8-and-its-band-com binations-5ce297440d1c