Convolution Neural Network and Multisource imager for Multiclass Classification of Forest Cover using CNN

In our previous work, we have found that adding a Touzi phase *ϕαs*1 decomposition component to multispectral data improve the accuracy of image classification work. A Sentinel-2 image obtained on August 8, 2020 and fully polarimetric Alos-2 acquired on June 27, 2019 courtesy of Japanese Space Agency (JAXA) were used to derive land cover information. There are different ways of doing land cover classification with CNN, but in this case we opted for following approach.

At the first part of the project, we worked exclusively with Sentinel 2 data and training polygons. All Sentinel-2 bands including Touzi phase *ϕαs*1 component were resampled to 20 m.

We converted all our training polygons for the area to raster and stacked with multispectral image. The values of the raster represented different cover types matching Alberta Satellite Land Classification (ASLC) codes. For instance Black Spruce polygon has ASLC code of 51 while Aspen has 55. Our stack had in total 10 bands where nine are features (Sentinel2 bands) and one is labels (raster of training data).

Next, we generated regularly spaced point shapefile (figure 2) which we used to read values of all bands in the stack. We had 17 different cover types ( ASLC classes) including burns-41, different pure coniferous (Sb-51, Pl-52, Sw-53), mixed conifer-54, decidusous-55 and water 80, mixedwood and wetlands

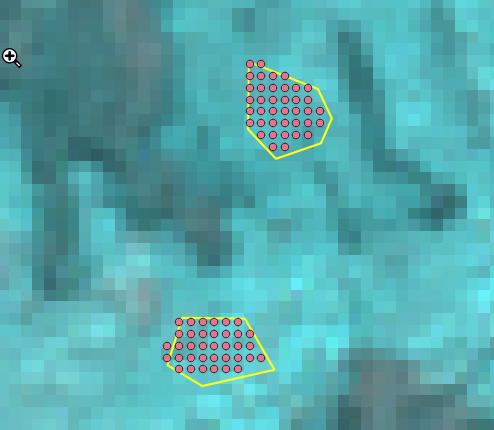


figure 4. Regularly spaced points 20 m

Then we converted that shapefile to CSV file. So, that we can use CSV file in python environment to easily manipulate input data. Our training sites contained 17 different ASLC classes, and instead of using regular ASLC codes (ex. Sb is 51), we added another field ASLC\_ to order them from 1 to 17. For instance, Burns with ASLC code 40 were assigned id 1 in “ASLC\_” field and last young Pine with ASLC code 552 received id 17. Our predictors (features) are bands B2, B3..B12 and our response variable (label) is “ASLC\_”.

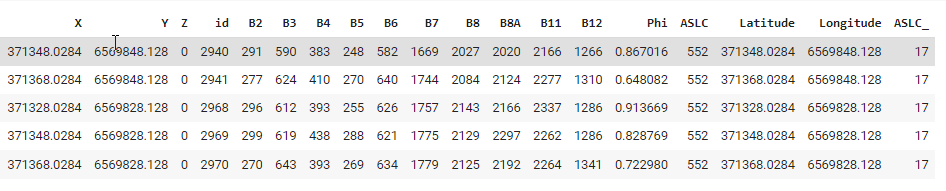


Figure 5. A CSV file

In the second approach we did the same, except we added Touzi scattering type phase *ϕαs*1 component to the input data layers and then fed the data to the model. This time there were 11 bands. Then we compared output of the model using Sentinel-2 data against the output of the model using Sentinel-2&Touzi phase.

### 2.2.1 DATA BALANCING

Balancing among classes is really important when it comes to accuracy assessment. Our training polygon samples range from 50 to 1000 regularly spaced points. When we look at the unique classes we see that close White Spruce (Sw) stands have in total 1085 samples, while graminoid wetland ,Black Spruce (Sb)- lichen and young Pine (82,87, 552) stands have barely 50 samples. To make them balanced, we down-sample those with high and up-sample those with low number of training samples. Balancing will result with each cover type having similar amount of training pixels(~ 500). Now we can split the data into training and validation.

## 2.2.2 SPLITTING THE DATA INTO TRAIN AND TEST SUBDATASET

In order to evaluate the performance of the model at a later stage and see how our model performs on unseen data , we need to create a validation set. This is done by partitioning the training set data using various ratios, but in this case we opted for 80:20. Before splitting the data we do shuffling so that different “ASLC\_” codes are randomly spread. Now, we will split the data for training and validation. This is done to make sure that the model has not seen the test data and it performs equally well on new data. Otherwise, the model will overfit and perform well, only on training data. The validation data set will be used to optimize the model parameters during training process.

## 2.2.3 NORMALIZING/ SCALING

## In the next step we normalize/scale the data which is also important so that all features (bands) are treated equally. Neural networks are sensitive to the distribution of data. Bands with reflectance e values (7,8 and 9) will play larger role in fitting the model unlike bands (B2 or B3) with lower reflectance values. Therefore we need to scale them in a range between 0-1 or -1 to 1. There are many ways to do it, but here we will normalize it under normal standardized distribution.

## 🡪 z = (value – mean)/standard\_deviation.

## 2.2.4 ENCODING

While doing multiclass classification,  hot encoding is necessary. We need to transform labels into an array, because labels can be strings or numbers and it is called 'hot encode' where label 1 and label 2 would look like [1. 0. 0. 0. 0. 0. 0.] [0. 1. 0. 0. 0. 0. 0.]

## 2.3. DEFINING Model architecture

The simplest deep learning neural network consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.A **hidden layer**is any layer between the input (first) layer and output (last) layer. There can be multiple hidden layers*.*  We will use the **sequential** model (figure 5) and add the layers one after another one which are fully connected. The *Dense* function in [Keras](https://keras.io/guides/sequential_model/) (python library) constructs a fully connected neural network layer, automatically initializing the weights and biases.

2.3.1. Train the model and make predictions

Once the network architecture is defined, and before model is compiled one still has to choose “*Loss function” or “error” -* the quantity that will be minimized during the training can generalize well to new data is a challenging problem.

## 2.5 PREDICTION

Once we have a model that we are happy with we save it into \*.hdf5 format, so that we can restore it and use it anytime for prediction. When we use a new image for prediction we have to make sure that we work with the same bands. If we used 9 or 10 bands to train a model, then we need to use the same bands for prediction as well. Also, if we trained model on normalized

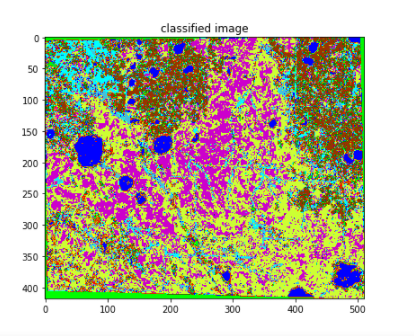


figure 10. Predicted output

data we have to use the same parameters as we normalize new unknown image. Therefore, our image for prediction needs to be scaled to the same standardized normal distribution parameters as well.

**3. RESULTS AND DISCUSSION**

### General conclusion was that initial model that did not take in consideration Normalizing and balancing data, produced lower accuracy for these 17 landcover classes. After closer inspection our product looks really good, it perfectly finds all these shruby wetlnads around water bodies, Black Spruce bogs and burned scars

### 

Figure 10. Confusion matrix for Sentinel-2 and Touzi phase *ϕαs*1 component

CONCLULSION

While observing classification report or confusion matrix we can find out which class the model performed bad out of the given 17 classes. We see that the classifier is underperforming for class 9, 2, 5. These are Black Spruce (ASLC- 51), Mixed Conifer (ASLC- 54) and Mixedwood (58). Classification report will help us in identifying the misclassified classes in more detail. The CNN model after training has 82.40% accuracy for multispectral and 87% for the multisource data.