Question 1

Every pixel of the image has a channel of numbers and we make use of the texture as well as the spectral characteristics of the images to separate before using DL. The specifics of the information in the remote frame segmentation cannot be adequately detailed, and the fulfilled segmentation cannot be provided for challenging tasks. DL in the natural scene supports DL in remote sensing images. As the ambient noise interacts, an irregular texture distribution, uneven light transitions as well as other parameters, complex remote sensor images must be processed. Thus the implementation of DL of remote sensing images needs greater specifications.

In the working model the encoder-decoder dependent on SegNet the CNN framework could be used to segment remote sensing images with the Pooling Index for multi-target semantic. The Pooling Index will store the special pixel information that may be returned during the decoding process. As the pixel location can be restored the specifics of the segmentation of the edges can be given. We apply enhanced batch normalization to CNN for multi-objective semantic segmentation for remote sensing images to be made acceptable. The normalization of the load applied will adjust the distribution of the input data to the next stage, such that the current distribution cannot be learned, on the other hand, the parameter can be updated more effectively.

We first train the model for all labels for image segmentation using the basic approach. Then we tweak the model to deal with the problem at hand and lean toward the new segmentation approach.

- Our work consists of many operations like the image is being cropped into patches and provided with a label.
- In image classification issue, the computer would produce a discrete label as we assume that the image contains only one object and not several objects. The purpose of Semantic Image Segmentation (SIS) is to label every pixel in an image with its respective class. This function is widely referred to as dense prediction since we are forecasting for each pixel in an image.
- U-Net is a network,functionally related to SegNet, where the encoding blocks use the layer before maximum pooling and link them to their corresponding decoder layer. Up-sampling is done on the blocks at the expense of more memory by transmission convolutional. U-Net has shown its efficiency in data sets without training data in which it is necessary to use maximum users. While satellite images can easily be accessed by various methods, data with accuracy is scarce mainly due to the lack of resources.

The main framework of FCN is used to execute the studied analysis by using U Net as a pre-trained framework & python language. Both performance and outcomes are seen with the training data set at end of the performance.

Question 2

After loading all the images in the model, first we pre process the date where we scale all the pixel values of the images between 0 and 1 by dividing the pixel values (which were between 0 and 255) by 255. It is important that we preprocess the training set and test set int he same way. First we create the convolutional base. We can use several different layers involving Convolution, Max pool, Normalisation, to create the convolutional base and get the output tensor. Then we apply some more layers on to this output tensor. First, it is flattened (this is like unstacking rows of pixels in the image and lining them up). This layer has no parameters to learn; it only reformats the data. Then we can apply one or more dense layers to this flattened array. The final layer of the these dense layers returns an array of logits of the image for each category. Thus, if the number of categories are ‘x’ then the last dense layer should be such that it gives ‘x’ scores. We can use the softmax function after this to convert this array of logits to probability of the imaage being an element of each of the categories. Before training the model, we need to compile the model. This includes creating:

- Loss Function - This measures how accurate the model is during training.
- Optimiser - This updates the model based on the data it sees and the loss function.
- Metrics - It monitors the training and testing steps. (We cana use several different parameter like accuracy, precision, recall to monitor depending upon the model.)

We now train the model. We feed the training data to the model. The model learns to associate images and labels. As the model trains, the loss decreases and accuracy increases and at some point it dosen’t change. Our model in now ready. We can calculate the accuracy of this model on the “Validation set” or “Test Set” and use it to make predictions about the class of an image.