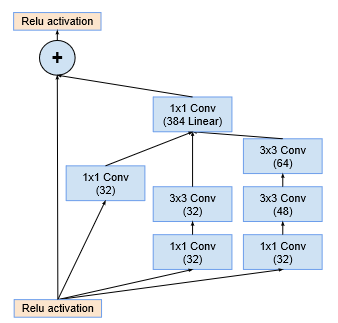
Inception Resnet V2

Why we do classification (exist ship/no ship) when the purpose is to do detection and segmentation? Since we may process huge number of pictures at the same time, it is a must to consider the speed of the algorithm. Obviously doing segmentation costs a lot more time than classification. Then one idea is to do classification and filter the pictures without ships and do segmentation only on pictures with ships.

We are both familiar with the net Inception and Resnet. Simply speaking, Inception Resnet V2 is just a combine of Inception V2 and Resnet.

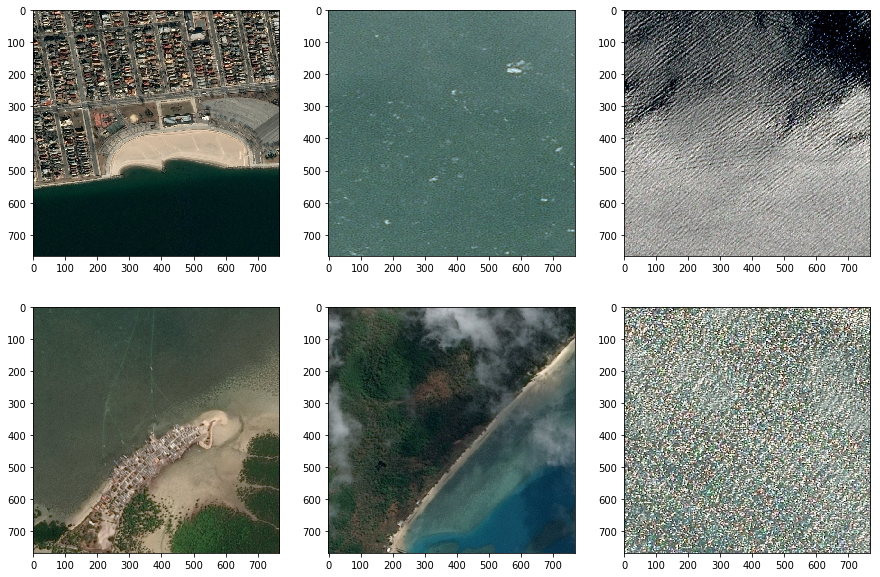
Inception V2 is a transitional model in the developing of Inception net, but it introduces two important features which could make the net deeper and faster. Since these two features (batch normalization and replacing big kernel size with two small ones) have been taught in class, we do not need to go deep to this.

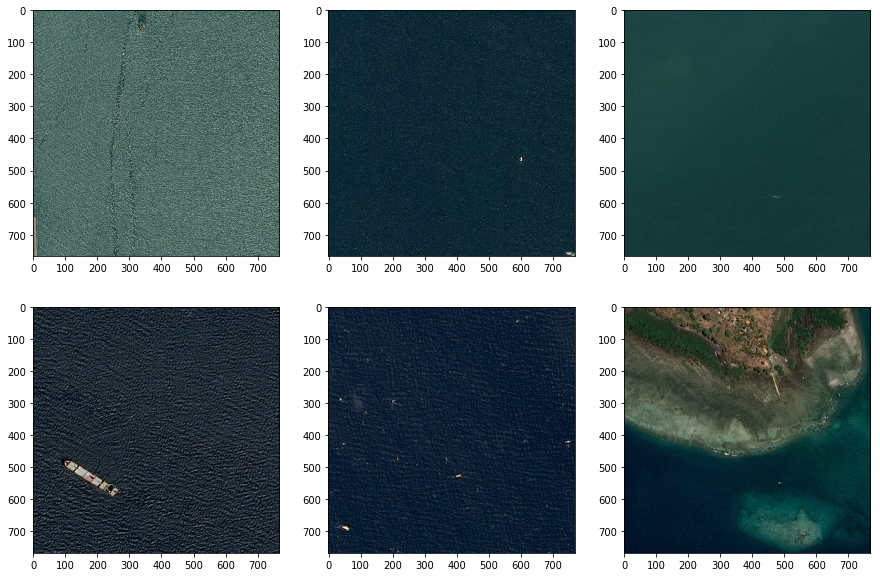
Pic.1 The schema for 35 × 35 grid module

of the Inception-ResNet-v2 network1

After V2, engineers have also developed V3 and V4 of Inception, however, these models are developed under the idea that without Resnet, we can get a similar effect just by modifying the Inception net. Compare to Inception V4, Inception Resnet V2 performs a little better just by adding some Resnet (under a near complexity). Furthermore, due to the import of Resnet, Inception Resnet V2 maintains a much faster speed of convergence in training.

After getting the trained model, we could see that on the training set, we get an accuracy of 12.76% on the training set. Then we look for the FP and FN for this model. The FP rate is about 32.9% and the FN rate is about 2.7%. However, this result shows that when there are no ships in the picture, we almost make no error and otherwise. It could not be an ideal result because when we put a picture with no ships into the segmentation, we generally get nothing (no bounding box). However, if we judge that a picture (with ships) have no ships, we would never put it into the segmentation. Then the performance of the preprocessing is dominated by that FP rate. We could have a look at the pictures that the model makes a wrong decision:



Pic.1 the wrong choice in FN

Pic.2 the wrong choice in FP

In Pic1. We could see that most FN pictures contain unsmooth special waves or lands. On the other hand, the situation of FP(Pic.2) is mostly that the ship is too small to define. This could be explained by two reasons:

1. When turn the 768x768 picture to 299x299 one, we just do scale change which means that small ships with only a few pixels are further compressed. Then it is harder to find the pixels of ships. One solution is to do convolution and max pooling on the raw picture(768x768) and turn it to 299x299. This process could be hard to control since the we could have quite different methods to do this and the gradient descent before Inception Resnet V2 is hard to control.
2. When we learn segmentation, we have the label of the exact location of the ships (the bounding box), then we could focus on the items we want to learn. However, in this situation, we only have the label of have/no ship. The model does not have enough information on the what to learn. Little ships could be seen like noise. Then it is general that the binary accuracy in classification is not as good as the accuracy in segmentation.

Improvement:

Using more data augmentation methods to make model robust. (We now only have flip)

Combining with other preprocessing to compensate the loss of ships.

Reference:

1. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi