We use convolution to compute the training data picture. The formulation of the convolution is : $z[n_1, n_2] = \sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1} w[k_1, k_2] \cdot x[n_1 + k_1, n_2 + k_2]$. The parameter n_1, n_2 is a position in the picture and we get its value by create a filter w and let it has convolution with part of the original picture. The meaning of this value $z[n_1, n_2]$ is that when the picture is conform the feature which this filter described, then we will get high value in that position. In the final dense layer, we will use the formulation $z[i, k] = \sum_{j=1}^{N_{in}} w[j, k] \cdot u[i, j] + b[k]$ .and k is the number of output channel, j is the number of input channel.

In this case, we are trying to deal with the classification problem. Then we label the y_i as 1 if there is a ship in that picture and label 0 if not. Then after the activation layer we will use binary cross entropy to evaluate the loss function: $\sum_i \ln\left(\sum_k e^{z_{oik}}\right) - \sum_k r_{ik} z_{oik}$ To get the accurate gradient, we use Adam optimizer to calculate the gradient. In this case we use SGD update our gradient as we has too many samples and we could not calculate the whole training data to get the value. Instead, we choose a random batch of images to calculate the gradient use the function: $\theta^{t+1} = \theta^t - a^t g^t, \theta^{t+1} = \alpha^t \nabla L(\theta^t) - \alpha^t \zeta^t$

.