Training the neural network

1. Generate data:

In order to make neural network works accurately, we need large amount of data. The following section guides you to generate data from small amount of images taken.

Collect positive and negative images:

Positive image:

- Taken from the machine with tool in the middle

Negative image:

- A image without tool taken from the machine

Crop the image:

- Crop out the tool section from positive image with aspect ratio of 1:2 and scale it to 15by30 dimension.

- Generate negative image using negativeSample.m Matlab code from Data Compression folder. You should provide two folders. The program will generate more than enough image to the first folder and the program randomly select 4000 of them into second folder.

Generate positive train image:

We use python library call Augmentor to generate 4000 positive training image from the original positive images. Use sudo pip install Augmentor to install Augmentor. Link to Augmentor https://github.com/mdbloice/Augmentor

After installing augmentor, we use the program generator.py. Please specify the directory where all positive images are located. The generated picture will be generated in the source folder and inside folder call output.

Convert image into matrices:

We use imageToData.m from Data Compression folder to convert positive images and negative images into matrices data. We specify the folder with 4000 positive images, and temporary folder to rename positive image into ascending order(Because the image in the output folder is name randomly) and the folder holding 4000 negative images. We save the data into trainingData.mat which contains X and y matrices.
X is image data 8000 images and 450 pixels per images, so it is 8000by450 matrix.
Y is a column vector tells if the correspondent image is a tool or not. So the first 4000 elements is 1(is tool) and last 4000 is 2(not a tool).

Compress the data:

By compressing the data, the neural network can work 5 times faster. Use the program dataCompression.m to compress the data. It generate two files: compressed data and PCA parameter file. Compressed data will be used in training and PCA file will be used in detection.

2. Training the neural network:
Move the compressed data to Training folder. Call tool training function. You need to specify number of neuron in the network and number of iteration you want your network to go through. One iteration is one forward propagation and one backward propagation.

The function will give you an estimation on precision of the neural network. The output of the function is parameters(weights) of the neuron in hidden layer and output layer.

3. Detection:
Move the PCA parameter file and neural network parameter to the detection folder.
In the detection folder, detectTool.m can be call to detect the location of the tool. It return bounding box size and location of the top left corner of the bounding box.
Once you are satisfied with the result, you need to save the PCA and neural network parameter into .yml file which can be read from C++. To save the parameters into .yml file, you need to run Convert2YML.m.