**Advance Software Engineering Project**

**Recognizing Hand Written Digits and Characters**

Software Requirement Specification

**Group 5394\_SM\_2**

Ravula, Balachandar

Kongara, Jyothesh

Pavuluri, Nitindra Chowdary

Sabah, Ayesha Fatima

Singh, Anurag

The CharacterRecognition is a maven project. The code is residing in github. You can clone or download the project from <https://github.com/Jyothesh/CharacterRecognition.git> .

Firstly, you should have few things to be installed

1. Java development environment (1.8)
2. Java runtime environment (1.8)
3. Maven
4. Tomcat web server
5. Eclipse IDE

Once the above software is installed, import the project to eclipse by using the import > maven > existing maven project> source.

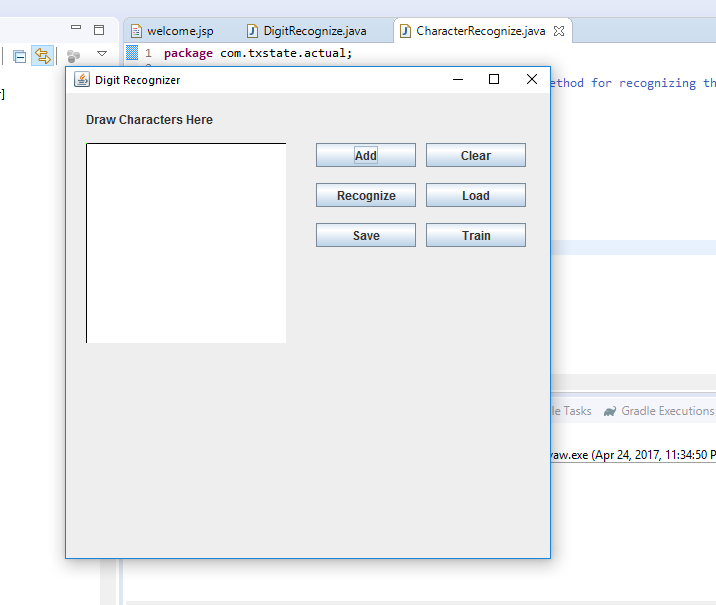
This project has two different things

1. Character Recognition
2. Digit Recognition

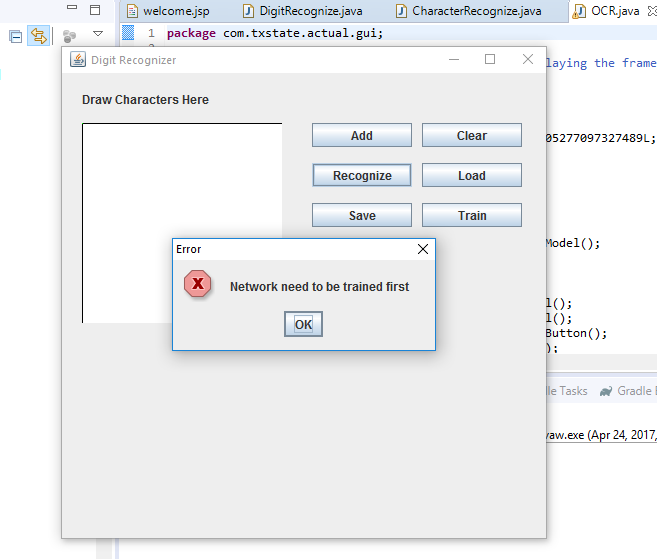
Character Recognition:

It is a Java Swing application. CharacterRecognize.java is main class for running it. You will find the above class under com.txstate.actual.

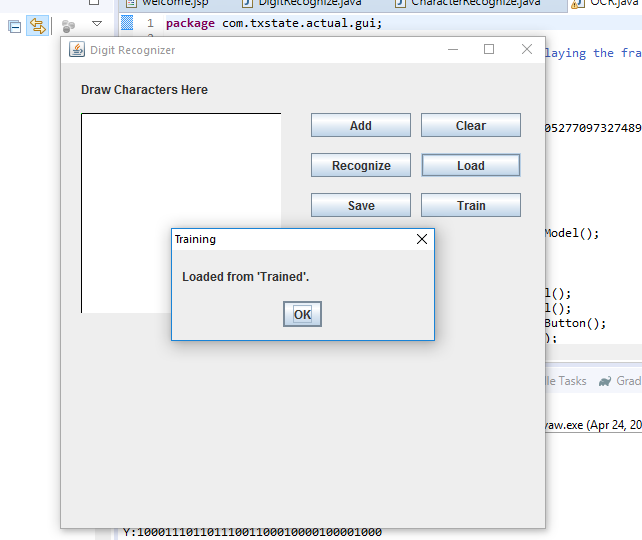
Run that class by using ctrl+f11. The project will start executing and you will see the below gui.



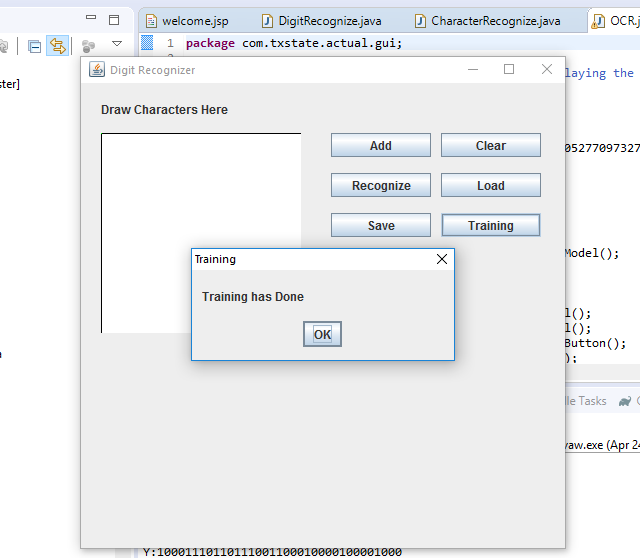
Now you need to train the network, before recognizing the character. If you click recognize button without training you will get error message as shown below.



Since it takes a lot of time to train a network, we already did some training for alphabets and saved it to the disk under resources. When you click it will be loaded from that file and you need to click the train button. Follow the below images.

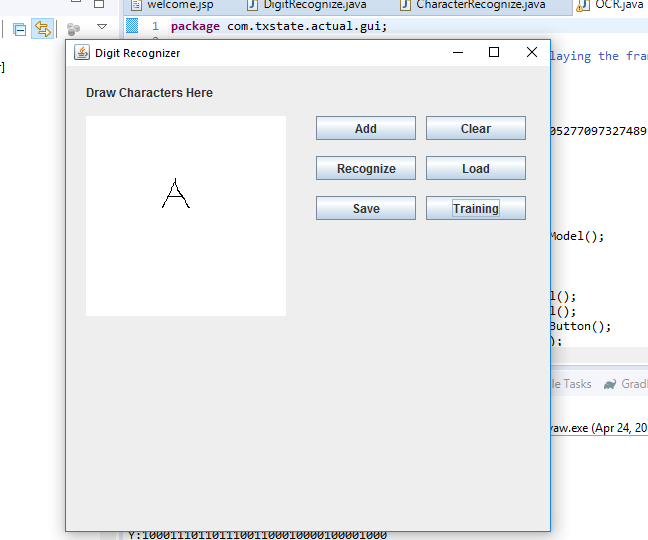


Click train button

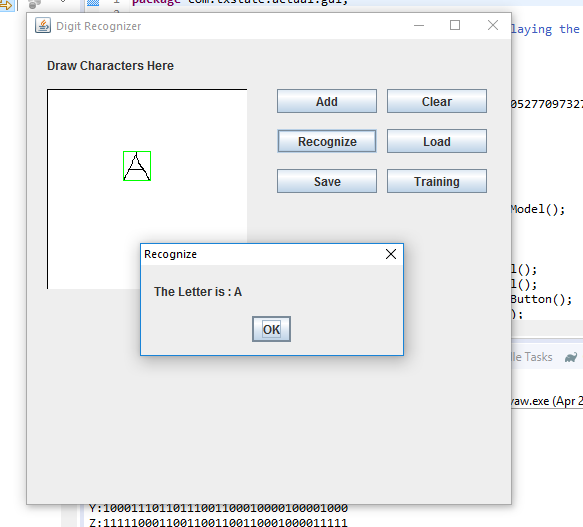


Once the training is done.

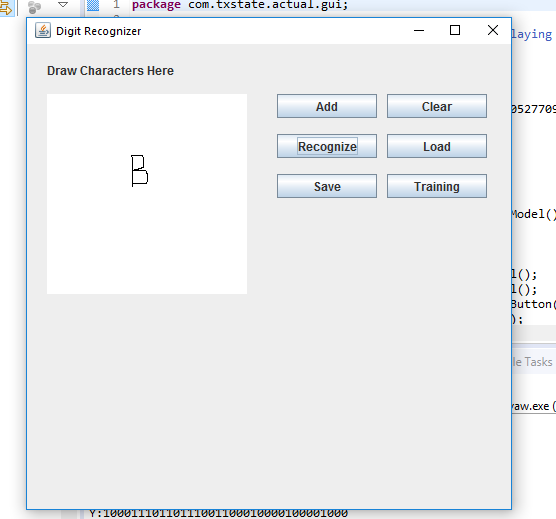
You can write some character in the image panel.

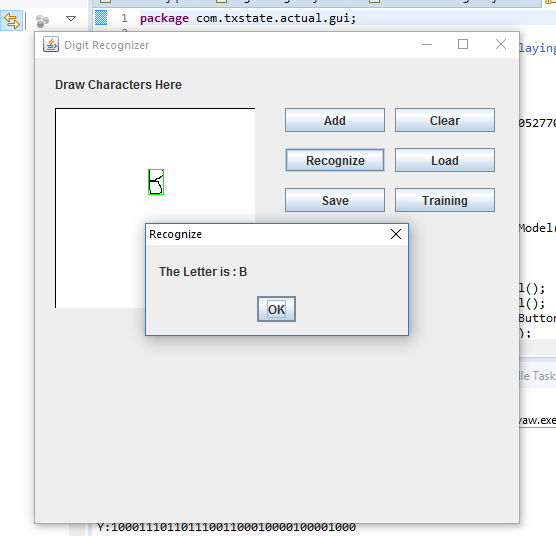


Now click the recognize button.



You can see that the we are cropping and bounding the image and passing it to trained network. This is how we can recognize a character. Few more examples are

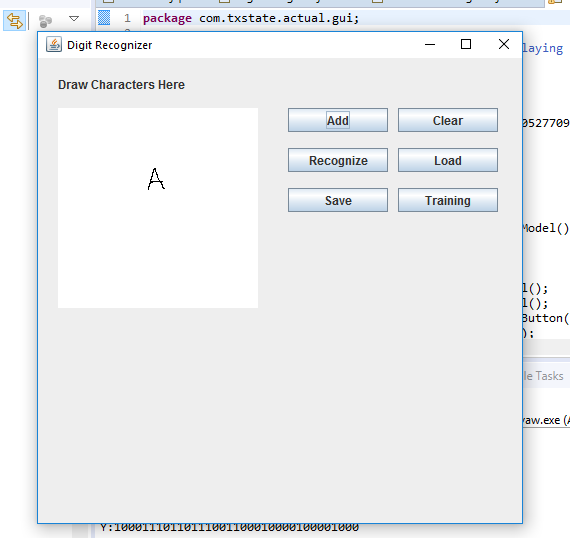




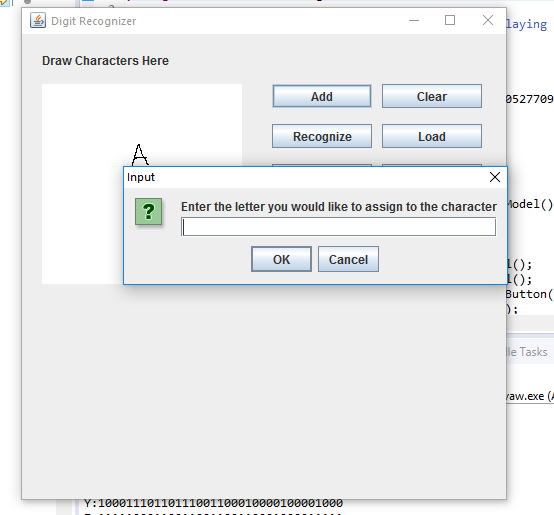
Now, let’s see how to train a character.

We need to use the add button for training.

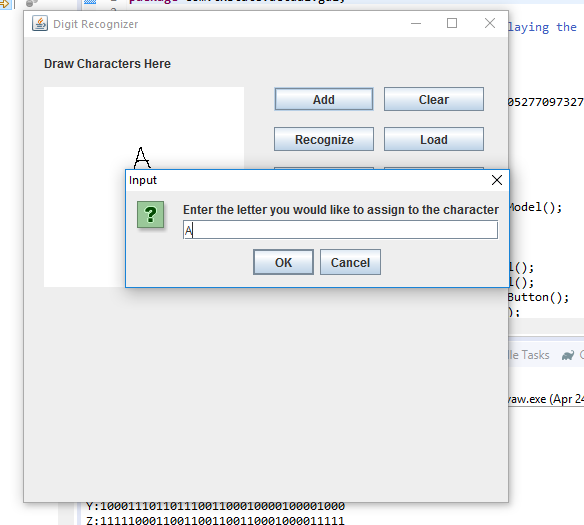
Let’s suppose we need to trained the character A. First write the character and click add.



You will see a popup to enter the character as shown below

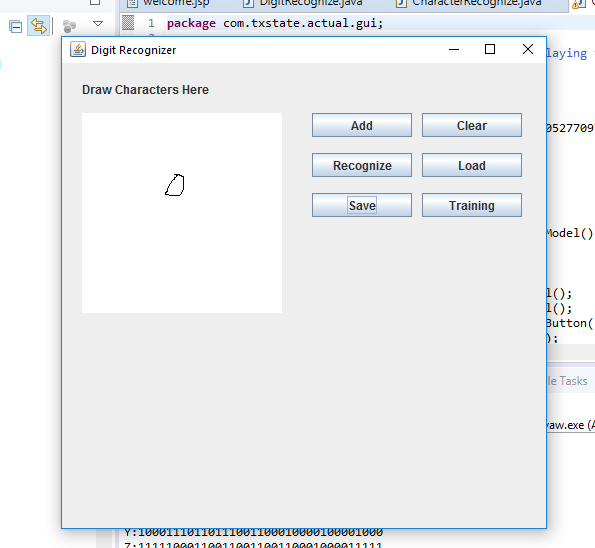


Now you will enter the alphabet A and the network will use it as one of the pattern for alphabet A.

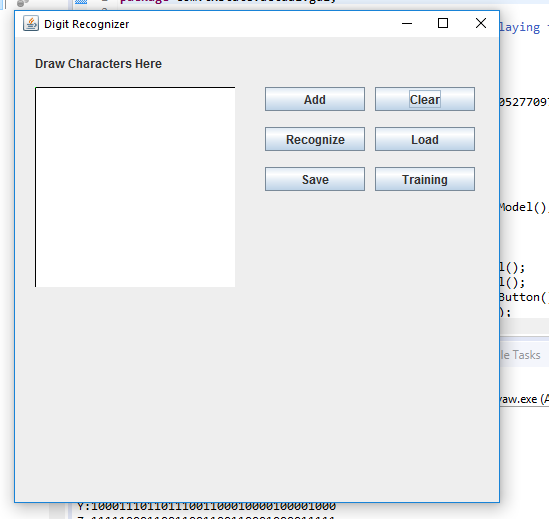


Click ok.

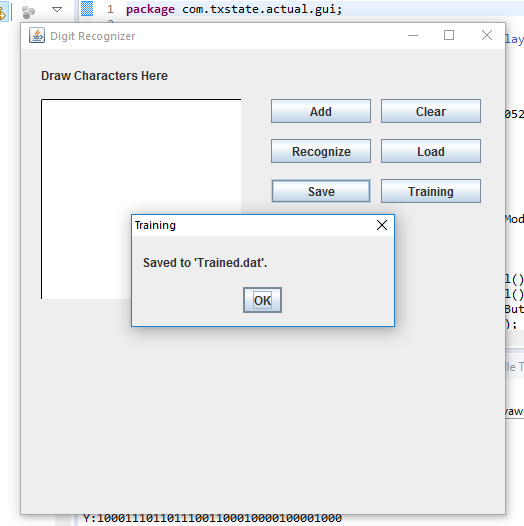
Clear button is used to clear the image panel.



After clicking clear button



You save the network by clicking the save button.



Digit Recognition:

For digit recognition, we used MNIST dataset which is freely available at

<http://yann.lecun.com/exdb/mnist/>

We downloaded the dataset and trained the network using it. Here, we are using the Back-propagation algorithm in neural network.

We did it as a web application, so that we can host it over the internet.

This is the url for accessing the application.

<http://ec2-52-35-157-253.us-west-2.compute.amazonaws.com:8080/CharacterRecognition/>

you can also deploy it locally using tomcat web server and eclipse.

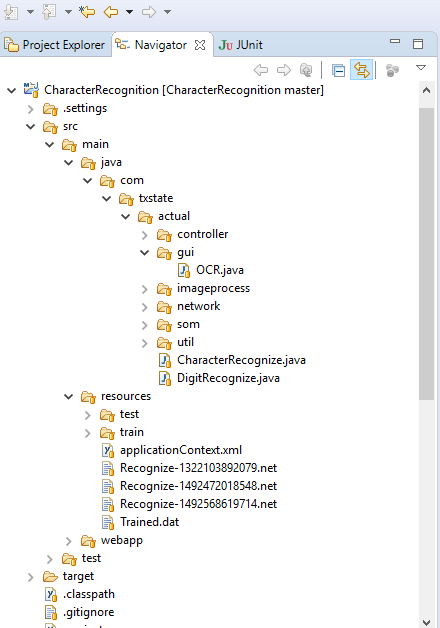
The java doc is also deployed on internet. You can access it using the below url

<http://ec2-52-35-157-253.us-west-2.compute.amazonaws.com:8080/CharacterRecognition/site/apidocs/index.html>

DigitRecognize.java under com.txstate.actual is the file that will train the network by loading the dataset which is present in the resources folder.

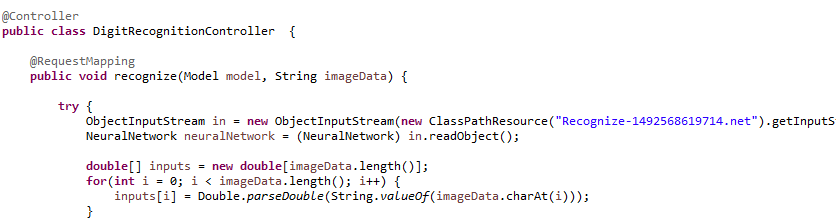
It takes a lot of time like 9-10 hours depending the hardware configuration to completely the train the network.

Once the training is done, it automatically saves the file into resource folder.

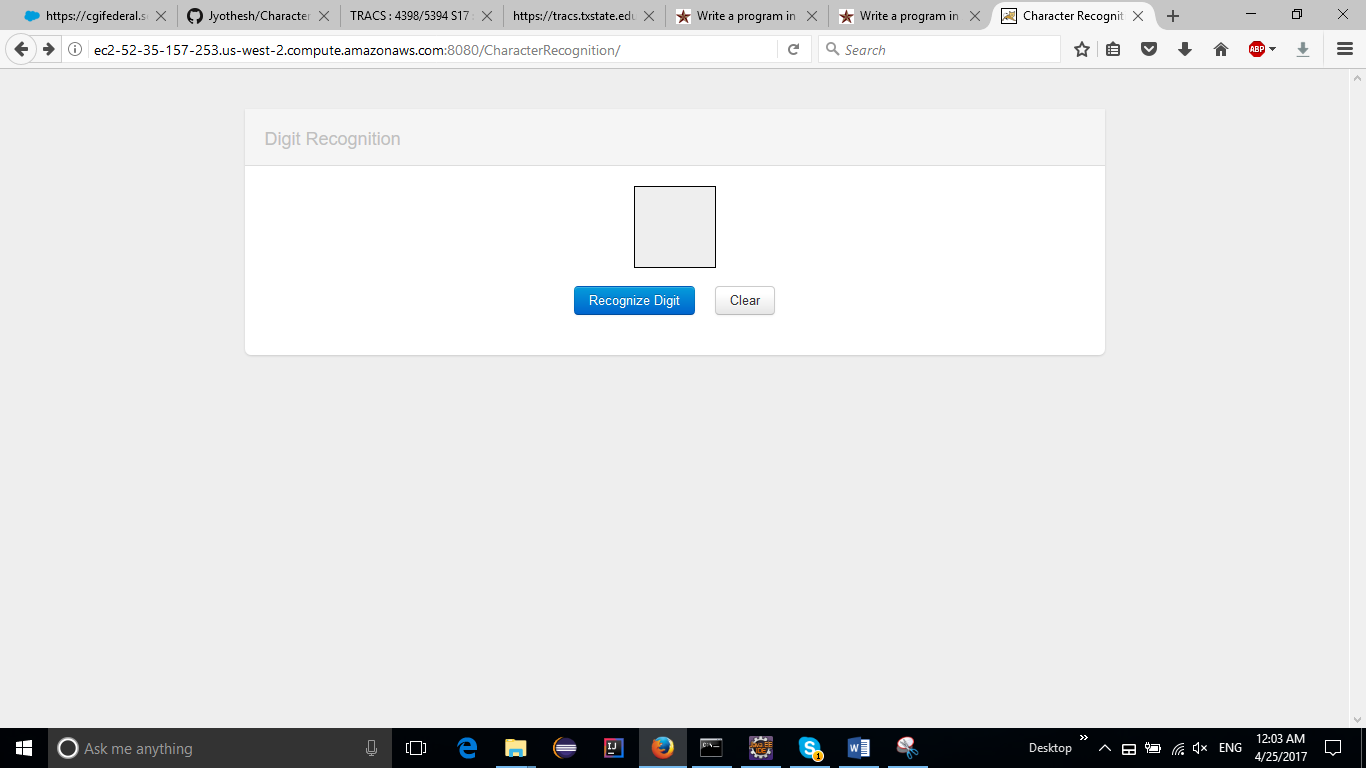


The Recognize-\*.net is the file. The filename is based on the system time. New file will be generated once the training is done.

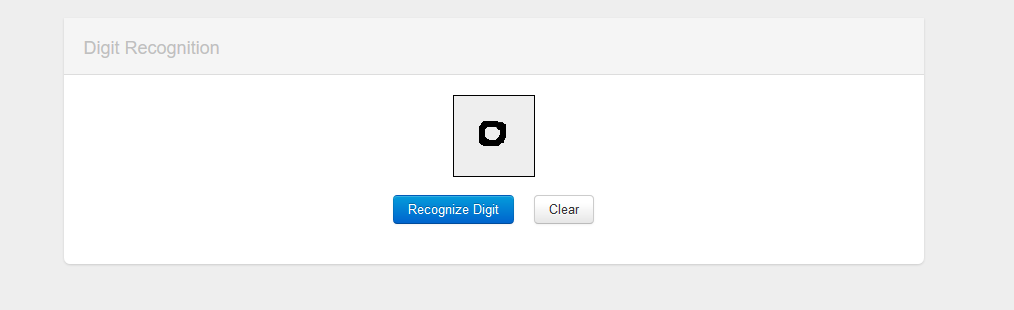
The web application uses this file for loading the network and recognize the character, which is done in the controller.



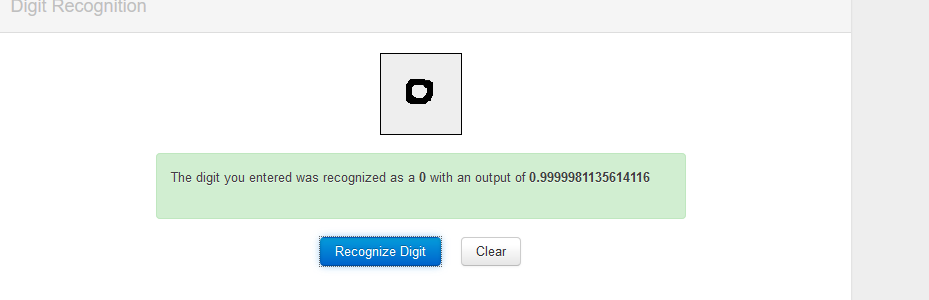
This is a sample, when you try it online



Draw a digit



Click the recognize Digit button



You will see the recognized character and its accuracy.

The class diagram is also added here



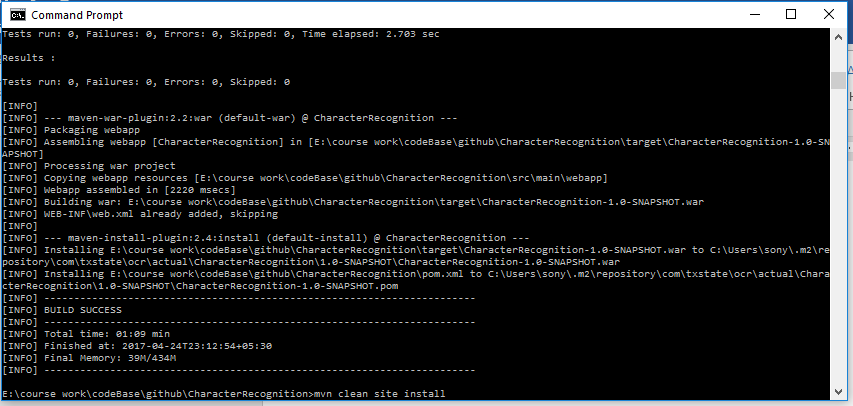
We have a sample test for digit recognition. The SampleTest.java will take the imageData as input and recognizes it by loading the network.

Improvements that can be done:

1. Move the character recognition to Web application
2. Junit test cases to improve the coverage. (As, we didn’t have much time, we were not able to write test cases)
3. Also, use the NIST dataset for training characters as well.

Commands to build the project:

Mvn clean install

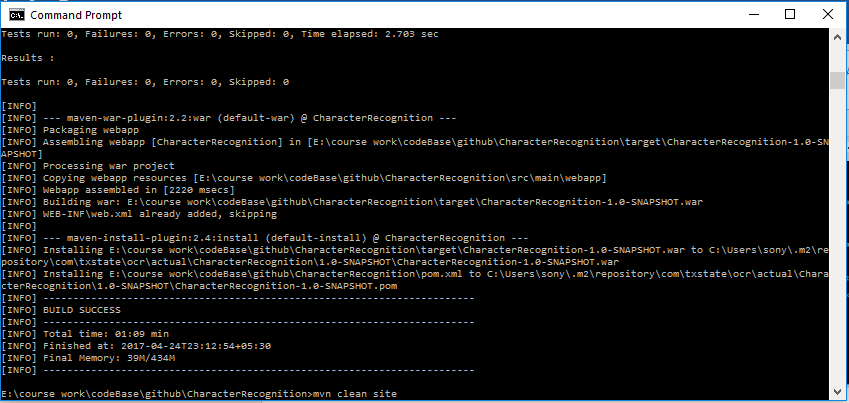


This command is for generating the war file.

We are using maven plugin for generating the java docs.

Command for that is

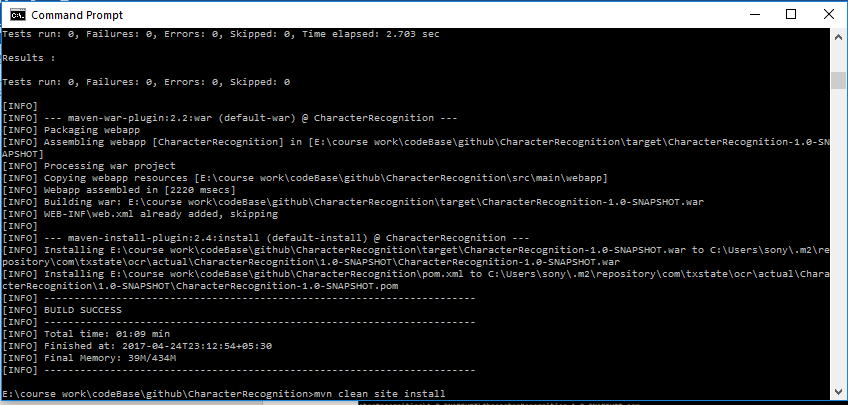
Mvn clean site



This will generate the folder called as site under target. Which is to be placed under character recognition folder after deploying.

You can also use a single command for generating the docs and war file at a time

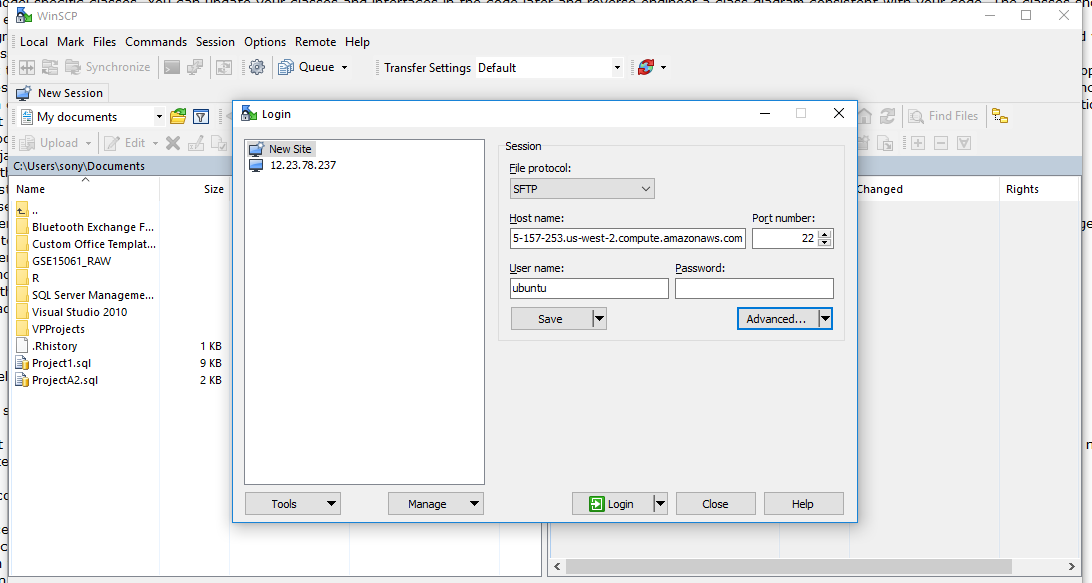
Mvn clean install site

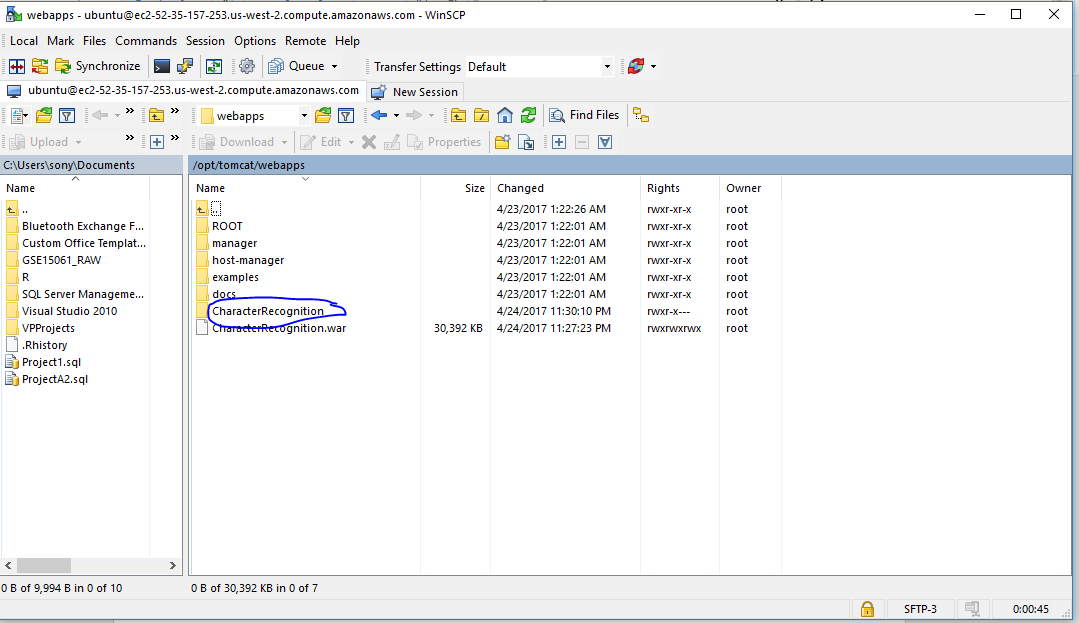


Folder structure of Character Recognition web app:

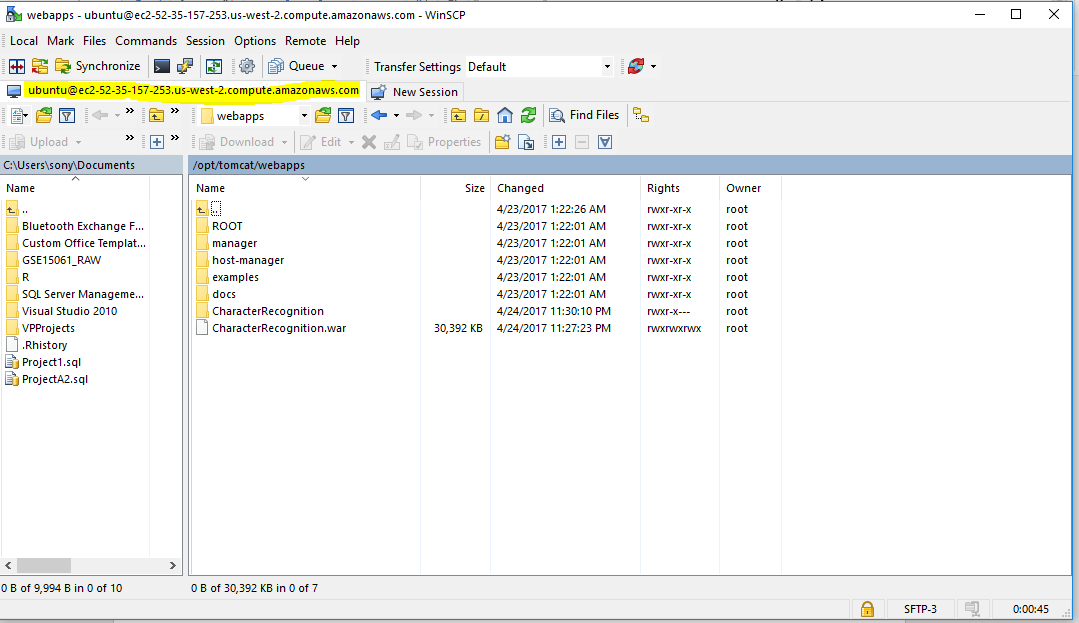
It is deployed in amazon ec2

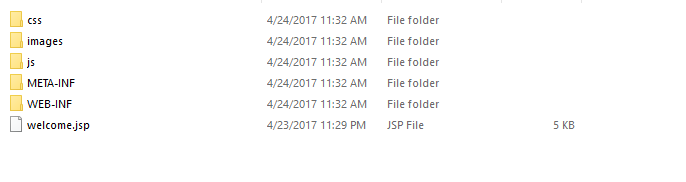
We will be using putty or winscp for accessing the server.





Which is deployed on amazon server “highlighted”





These are folders that are available under CharacterRecognition.