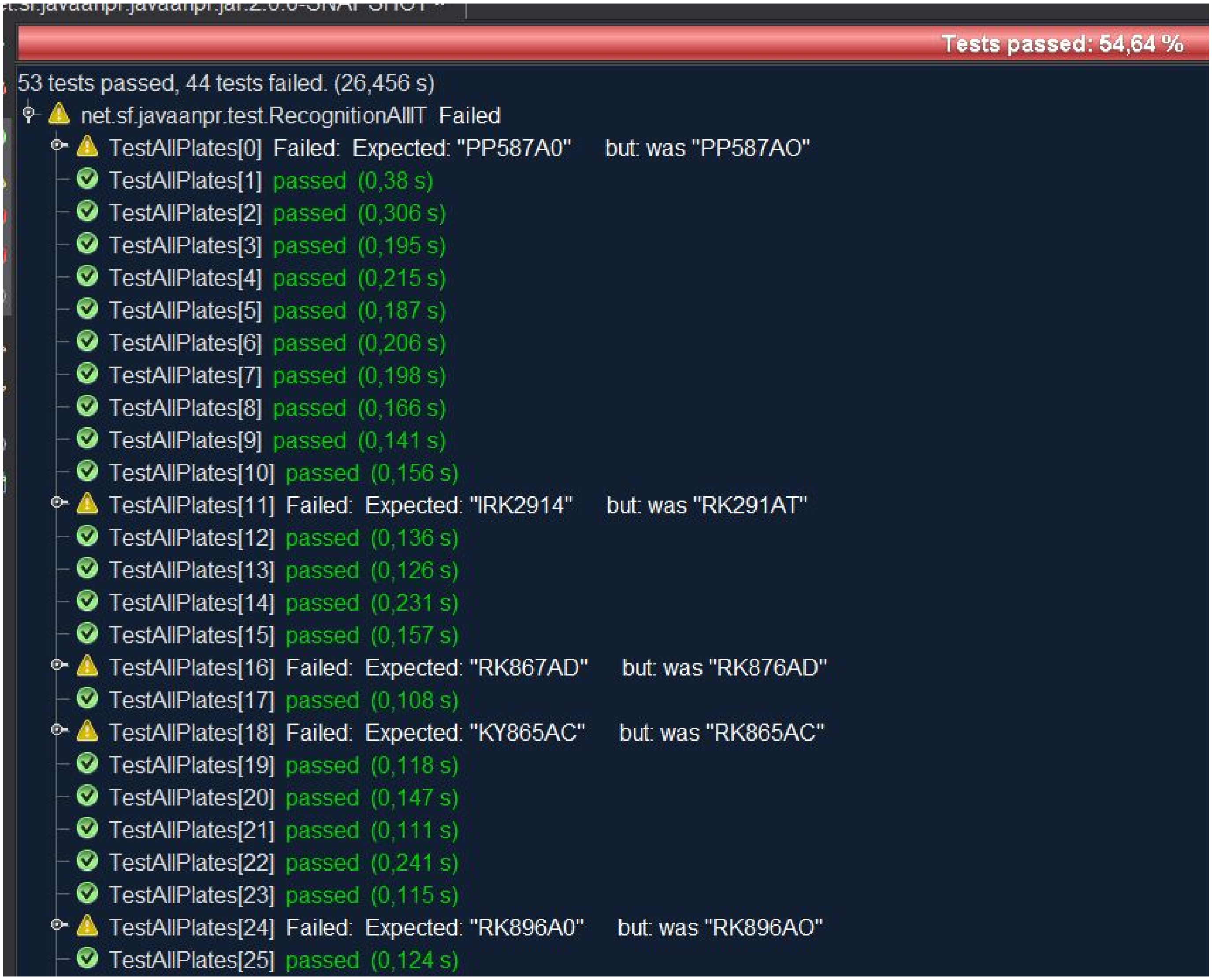
- *Explain the purpose of the Test (what the original test exposed, and what your test expose)*
  The purpose of test i to check whether or not what we have done is correct to what we expect. The original text exposed the result of the combined 2 tests, allowing us to see that the overall test of all snapshots passed but not telling us which one of them actually did passed in a user friendly way.
  The single test fails as not all of the snapshots passes.
  Our test test all the snapshots and shows exactly which on the snapshots that actually passed and which who failed in a easy to read manner, by giving each snapshot its own little test.

- *Explain about Parameterized Tests in JUnit and how you have used it in this exercise.*
  Parameterized Tests is a way to make the same test a 100 times without having to write 100 different tests that are almost identical except for the test value. We simply write a test that takes in a parameter with a value that changes for every time we runs the test, this saves much coding as well as it can save a lot of headache for writing the same stuff over and over. The test is invoked once for each item in the parameter list.
  In this assignment I used it to run my test, TestAllPlates, on every picture in the array filled with the pictures from the src/test/resources/snapshots path.

- *Explain the topic Data Driven Testing, and why it often makes a lot of sense to read test-data from a file.*
  Using files to contain data for Data Driven Testing make sense in the way that we may use a single script to test how our code handle information from different kinds of file formats as well as we do not need to hard code the test data directly into the program, if we do this the test data can only be changed from within the program itself, but if we use a file we may change the data from outside the testing program, we may also have the data changed by a second person hile we are modifying the test code.

- *Your answers to the question; whether what you implemented was a Unit Test or a JUnit Test, the problems you might have discovered with the test and, your suggestions for ways this could have been fixed.*
  I did not encounter any particular problems that was not caused by lacking imports, naming errors, or simple closing mistakes.



```
et.sf.javaanpr.javaanpr.jar:2.0.0-SNAPSHOT »
                                                            Tests passed: 54,64 %
53 tests passed, 44 tests failed. (26,456 s)
  net.sf.javaanpr.test.RecognitionAllIT  Failed
    TestAllPlates[0]  Failed:  Expected: "PP587A0"     but: was "PP587AO"
    TestAllPlates[1]  passed  (0,38 s)
    TestAllPlates[2]  passed  (0,306 s)
    TestAllPlates[3]  passed  (0,195 s)
    TestAllPlates[4]  passed  (0,215 s)
    TestAllPlates[5]  passed  (0,187 s)
    TestAllPlates[6]  passed  (0,206 s)
    TestAllPlates[7]  passed  (0,198 s)
    TestAllPlates[8]  passed  (0,166 s)
    TestAllPlates[9]  passed  (0,141 s)
    TestAllPlates[10]  passed  (0,156 s)
    TestAllPlates[11]  Failed:  Expected: "IRK2914"     but: was "RK291AT"
    TestAllPlates[12]  passed  (0,136 s)
    TestAllPlates[13]  passed  (0,126 s)
    TestAllPlates[14]  passed  (0,231 s)
    TestAllPlates[15]  passed  (0,157 s)
    TestAllPlates[16]  Failed:  Expected: "RK867AD"     but: was "RK876AD"
    TestAllPlates[17]  passed  (0,108 s)
    TestAllPlates[18]  Failed:  Expected: "KY865AC"     but: was "RK865AC"
    TestAllPlates[19]  passed  (0,118 s)
    TestAllPlates[20]  passed  (0,147 s)
    TestAllPlates[21]  passed  (0,111 s)
    TestAllPlates[22]  passed  (0,241 s)
    TestAllPlates[23]  passed  (0,115 s)
    TestAllPlates[24]  Failed:  Expected: "RK896A0"     but: was "RK896AO"
    TestAllPlates[25]  passed  (0,124 s)
```