Lisp in Summer Projects Submission

Submission Date	2013-10-01 15:38:03
Full Name	Nikita Beloglazov
Country	Belarus
Project Name	Nonojure Recognition
Type of software	gui app
General category	utility
LISP dialect	Clojure
GitHub URL	https://github.com/nbeloglazov/nonojure
Did you start this project?	Yes, all the code is written by me
Project Description	I want to describe my project in this form.
Purpose	Nonojure Recognition is a utility app that parses images of nonogram puzzles and converts them to digital form. It may be useful when you have a lot of nonogram in paper form (magazines, newspapers) and want to create nonogram app. With help of nonojure one can (ideally) avoid manual retyping all nonograms.
Function	Given photo of nonogram create digital representation,
Motivation	Ultimate goal is to create open online database of nonogram puzzles that can be easily downloaded from web. It may be useful for those who want to create their own nonogram apps but doesn't want to waste much time on creating his own nonograms. Nonojure Recognition is a part of this project that helps to fill database.
Audience	Audience is pretty narrow, basically it just me.
Methodology	I used OpenCV for image processing and Encog for digit recognition. Puzzle recognition consists of 2 main steps: 1. Given photo, determine size of nonogram, find squares

with numbers. Substeps:
a. load image
http://s5.postimg.org/ig6hm7xav/step1.png

b. binarization using adaptive thresholding- convert image to 2 color image
http://s5.postimg.org/mhnkrvplj/step2.png

c. skeletonization - thinning all lines to 1px width
http://s5.postimg.org/swmlojwbb/step3.png

d. finding grid nots
http://s5.postimg.org/6v0ws917b/step4.png

e. add missing knots
http://s5.postimg.org/9qdzz457b/step5.png

f. build squares based on knots
http://s5.postimg.org/oav2ty05z/step6.png

g. find squares that contain numbers
http://s5.postimg.org/w4vofc7yv/step7.png

2. Given squares with numbers parse them and recognize. Each square processed as follows:
a. given square extract subimage from original binarized image and convert it to 30x30 image

b. clear noise from image and center it

c. try parse image as if it was 2 digits that a not connected (2 separate componentes)

d. try parse image as if it was 2 connected digits (1 component, digits might have been merged due to bad quality of photo or binarization algorithm)

c. try parse image as if it was 1 digit

Digit recognition was done using neural network that was taught beforehand.

Conclusion

Project can parse some specific format of nonogram like in examples in "Methology" section. It can't parse nonograms when numbers are not in squares like here:
http://s5.postimg.org/a9nom4lzr/bad.jpg

The other problem is big puzzles. Lines in big photos may not be very straight because of soft paper and numbers are pretty small and recognition process gives bad results.

Example: app parses 15x15 and 30x30 puzzles correctly but 45x45 gives some errors in digit recognition.

In future I should definitely support more nonogram formats like one in the link given before. Currently all images are resized so they sizes do not exceed 1000px. It may be bad solution for big puzzles. It should work with bigger images but currently it slows parsing process.

Build Instructions

Build instructions given in README on project page. Here they are:

	Installation is pretty complex because it uses [OpenCV](http://opencv.org/) java bindings. Steps: 1. Install leiningen - http://leiningen.org/. 2. Install OpenCV for java: 1. Use http://docs.opencv.org/2.4.4-beta/doc/tutorials/introduction/desktop_java/java_dev_intro.html manual to download and compile OpenCV with java support. 2. Copy `build/bin/opencv-246.jar` to `nonojure/recognition` folder and run: `lein localrepo install opencv-246.jar org.opencv/opencv 2.4.6` 3. copy `build/lib/libopencv246.so` to `nonojure/recognition/resources/native/` folder. 3. Download network.eg - https://www.dropbox.com/s/6eqjofaog2xtii7/network.eg to `nonojure/recognition` folder. This file contains configuration for neural network so it can recognize digits.
Test Instructions	There is no tests except running app.
Execution Instructions	Use `lein run` to run app. You can try images from `nonojure/recognition/resources/examples`. Please don't use `nono1.png`, `nono2.png`, `nono3.png` because they're not real photos and my app couldn't parse them. Well, you can try but it throws exception.
Describe any bugs or caveats	App tested on Oracle JDK 1.7.0_21. It should work with 1.6. I tried it with OpenJDK but got exception saying it hasn't found libpng. May be Oracle JDK includes some libs that are not included to OpenJDK.
Screen shots	screen1.png screen2.png
Official	I have read rules and have abided by them. I am 18 years of age or older. I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.