Development Instructions V1.0

Zhe Sun, ymwdalex@gmail.com

June 9, 2012

Description

This file provides the development instructions of the web application called "Detection of vascular bifurcations in retinal fundus images using trainable COSFIRE filters" which can be accessed from http://matlabserver.cs.rug.nl:8080/RetinalVascularBifurcations/index.html. Should you have any queries please do not hesitate to contact me by email: ymwdalex@gmail.com.

Development environment

- Eclipse: any version which supports "dynamic web project" development.
- MATLAB: must contain MATLAB Builder JA toolbox
- SVN: any version

File structure

- All the source code is managed by SVN. The url of the svn repository is `https://subversion.assembla.com/svn/retina/`. All the files reside there. The permission to access the SVN repository is required. At present, Nicolai Petkov, George Azzopardi and myself are the owners of the repository. Please ask us to give you the permission to access the SVN repository.
- If you fetch the source code from SVN repository, you will see two folders:
  - MATLAB: the scripts of the COSFIRE filters, entry function and some postprocessing functions.
  - COSFIRE: All the files in the web application project, which can be opened by Eclipse.

Deployment

- MATLAB library: if any MATLAB code is updated, the MATLAB library must be rebuilt as per below instructions:
  1. There is a "retina.prj" file under "MATLAB" folder. Open it with MATLAB Builder JA toolbox.
  2. Add or delete the MATLAB files to this project
  3. Press "build" button
  4. Copy the library "./retina/distrib/retina.jar" to the web application folder "/WebContent/WEB-INF/lib"
- Web application: the web application is deployed on Apache Tomcat 7
  1. Export the project as a "war" file, name it as "RetinalVascularBifurcations.war".
  2. Run "cmd" and go to Apache Tomcat 7 directory

3. Shutdown the server by the command "./bin/shutdown"
4. Delete the old folder of web application "webapps/RetinalVascularBifurcations"
5. Copy "RetinalVascularBifurcations.war" under "webapps" directory
6. Startup the server by the command "./bin/startup"
7. The real url is `http://matlabserver.cs.rug.nl:8080/RetinalVascularBifurcations/index.html`. The other MATLAB server already occupied port 80 by Apache httpd, the access url of our project is `http://matlabserver.cs.rug.nl/RetinalVascularBifurcations`, then the page will is redirected to the real page.

High level design of the web application

- The fore-end job:
  - The foreground job is developed by Html + CSS + JavaScript + JQuery
  - Layout and CSS follow the design of other former MATLAB server application in "http://matlabserver.cs.rug.nl
  - Html + Javascript communicate with the user; If any server side action is called by the user, a Ajax call is evoked, the Java Servlet in the back-end executes and respond to the Ajax call
- The back-end job (server):
  - The background job is developed by Java Servlet
  - "InitParam" is a special Servlet and is called when the web application startup. It takes charge in the initialization of system configuration.
  - The MATLAB code is compiled as a "jar" library, and called as a Java class.

Files list

- MATLAB server
  - retina.prj: MATLAB Builder JA project file
  - retina.m: Entry of the COSFIRE filter. Generate and save the COSFIRE response in the sever.
  - SystemConfig.m:Configure the COSFIRE parameter. Pay attention to "params.dataset.groundtruth" until "params.dataset.trainingData". It tells the MATLAB code where the input/output/operator files are. pickBestCircle.m: The responses of the COSFIRE filters may be very close to each other. This script implements a condition that if the distance between two strong responses is less than a given distance threshold then the weaker one is suppressed.
  - posterior.m: A post analysis method based on some mathemtical morphology used to enahnce the localization obtained by theCOSFIRE filters. However, this is not being used.
- Fore-end:
  - "./" folder
    - index.html: The main page
    - explanation.html: the page which explains the parameters
    - conf_operators.json: Configure the file name and the default threshold of each COSFIRE operator
    - conf_path.json: configure the path of the image files
    - conf_period.json: configure the maximum survival time of the files that are uploaded by the user. Here, you configure the interval which checks and cleanup the user uploaded file
  - DevelopmentInstructioin folder: This document
  - "./css" folder: CSS files

        - hpci.css: The same style as the former MATLAB server in matlabserver.cs.rug.n
        - jquery-ui-1.8.16.custom.css: JQuery UI CSS, for slider
    - "./Dataset" folder: Save the default input images and operator images
        - groundtruth: The folder that contains the manually segmented retinal vessel images taken from DRIVE
        - operators: The folder where puts the COSFIRE operators
        - originalImg: The folder that contains the colored retinal vessel images taken from DRIVE
        - outputImg: Empty folder, the generated images will be stored here when the server is running in Tomcat 7.
    - "./img" folder: Some images sources used in the webpage
    - "./js" folder: JavaScript files
        - ajaxupload.js: The plugin used to upload image files
    - "./WEB-INF/lib" folder: the folder that contains the 3rd party library
        - retina.jar: The library generated by MATLAB Builder JA

- Back-end:
    - MainServlet: This servlet is executed when the user clicks "Detect the vascular bifurcations" button
    - DeleteUserImg: This Servlet deletes the temporary file after the user refreshes or closes the browser
    - DetectFolderExisted: Detects if the image has already been processed by the system
    - FetchImage: On a change event in the website this servlet is called and generate the image in the server
    - GetLocalImgFileList: This servlet returns the file list of the images in the database
    - InitParam: Initializes the parameters that will be used in the web application. This Servlet is called only
    - UploadBinaryServlet: Receives the file uploaded by a user, checks the format and size and then saves into the server disk.

Hint for future development

Some matters need attention when adding "segment the retina vessel" and "Upload the color vessel image":

- UI part
    - Ajax uploader plugin can be used to implement uploading function. The JavaScript code in Line 475 in index.html and UploadBinaryServlet.java are the examples.
    - Pay attention to the relationship with the two select box: if one changes, another one should be updated as well.
    - When uploading or do the segmentation, user input or other information text should be disabled. The JavaScript functions "disableAllInput" and "disableUserFilter" should be helpful.

- Servlet
    - If the segmentation function is written by MATLAB, use MATLAB builder JA to compile the library;
    - Add a Servlet descriptor in "./WEB-INF/web.xml"
    - Take care of the saving folder of the segmented image
    - Don't forget add the uploading file to the "UploadFileList", in that way, the user uploading file can be deleted periodically
    - If more global configuration parameters are needed, they can be initialed in InitParam.java and write the value in json files.