Report

By- Muskan vig (19983530)

DetectEdges.java: Assignment functionality-

A program to detect horizontal and vertical lines in a PNG (Portable Network Graphic).

• Be able to open both a .png and .csv file and retrieve a 2D array of greyscale values from it.

• Write the values from that array to a file, in .csv or .png format.

• Prompt the user to use an Image using values from User Input.

• Prompt the user to use a Kernel using values from User Input.

• Perform Convolutions based on the values within the array to detect lines.

• Perform Smoothing on a s×s area of pixels based on x,y coordinates.

➔ **Justify your approach to data validation**

⇨ I approached data validation in a way that when values are initialized, they must fall in valid range as given in this assignment specification. I used my UserInterface static class for validation as well as to handle input and output String message, is. In UserInterface.java, certain values will be checked in the valid range and will be looped unless correct value is inputted. For instance, let's see that image elements for manual input must have been in the range 0 and 255 including both 0 and 255. So, to approach this, I called a submodule i.e. userInput() in UserInterface class by exporting string message and 0, 255 as range value. This applies to other validations too.

Also, if a user inputs wrong data type then my program must not crash and handle exception by giving meaningful message. For that also I used UserInterface class. Other validations such as Image size, kernel size etc. are handled in model class Image. Validations for smoothing inputs is handled in its own submodule smoothing() in Menu class. Validations for date inputted in DDMMYYYY is handled in Date model class as it will store the splitted date components and will give the desired formatted date string.

→ Justify your design decisions in regards to what functionality was placed in the classes you decided to write.

- I decided to design my program in such a way where in I can make maximum use of submodules that is, modular approach and use different static and model class.
  My main entry point is DetectEdges.java as I thought about the main purpose of the program and came up with it as we have to detect horizontal and vertical lines and also perform other operations on the image.
  My DetectEdges.java uses another static class, Menu.java which has main functionality.
  Menu.java has other submenus so that it will go ahead if previous menu options work well. In my Menu.java I have been using the submodules which have one task to perform. For example, importImage() imports the image by calling another menu() to choose import option as from file or manually, calling userInputImage() further. Same functionality is performed by importKernel() and further goes to userInputernel(). I chose to have these submodules so that other menus appear only after previous required tasks are done. All these sub modules are being called from entry point DetetEdges.java. So in case few changes are asked to done, we will not make modifications to all the files, instead we will change at one place.
  Convolution() method is chosen to perform convolution operation by calling it from Image model class. This is decided on the basis of image validation and importing kernel in it for the task. This will help if operation can not be performed, it will give the message as image will be an object created after validation.

  Further down, FileIO static class performs reading and writing .csv and .png files. It is easier to call those methods from static class instead of writing them again and again.
  PDIMath class has my mathematical submodules performing mathematical operation. I made it's use in smoothing for ceil'd value.

  My model Date class is chosen with regards of saving file with date. I made it so that it will come handy wihile storing a string of formatted date. As we will input in DDMMYYYY format and split it. Model class will be useful in validatingdate as in days, month and year.

→ Explain why we separate our program into different static classes and model objects.

- Our program need static and model classes for improved modularity and functionality. It makes our work easier for future changes as well. We can make model classes so that we can create object and can make use as data types apart from primitives.
  Static classes have static submodules which can be called from other static method by class name itself. We don't need objects to call those methods.
  A very good examply of static class can be Math(). We use it very often and just pass the paramters to the methods of it. Like, Math.ceil(0.868).

➔ **Discuss the real world implications of using this algorithm. You don't have to go into full depth about this, just explain why we may need to detect/view edges in an image**.

⇨ Well, Images have noises and thus we sometimes want to reduce it. To reduce those noises, we perform smoothing algorithm to smooth it.
Convolution, detecting edges in images algorithm is handy to extract the structure of the objects in an image. If we are interested in the number, size, shape, or relative location of objects in an image, edge detection allows us to focus on the parts of the image most helpful, while ignoring parts of the image that will not help us.
It allows us to detect the transition from a "light" side. Hence, showing us a line or edge in the resulting image.

→ Discuss any challenges you had in your implementation/design.

⇨ Well, Initially the biggest challenge I faced was putting all the tasks together. I did tasks individually in worksheet. I had to give a thought that how will I tie knots together. So gradually I worked on individual tasks and moved to others.
In implementation, I did face a minor challenge of saving file with date but later on I realized that I made a model class for it in practicals which will be quite useful now. I worked my way around and then came across another challenge of implementing smoothing as it was a new task in this assignment. I figured out the algorithm as I had to take pen and paper to understand its implementation and there I go.
After facing these problems, I found my way around and learnt from those mistakes on the way.
My program compiled and ran as I expected and gave magical results after smoothing and convolution operations on both csv and png files.