Eclipse Vision System Plugin

Gilles Ardourel, Dalila Goudia, Sagar Sen

The objective of this project is to build a simple eclipse vision system plug-in that can identify interesting features in a video and publish a summary on a social networking site such as Facebook. The videos can come from various scenarios. For instance, in a home care scenario the plugin must be able to detect that a patient has fallen down/anxiety. Publishing a message or an image of the patient will alert his/her relatives on a social network so that they can required action. We can imagine similar applications in other images/video.

One possible way to design the project is as follows:

1) Middleware Vision System Plug-in: This base plugin will provide extension points to a number of other plugins. This is like that orchestrator that captures images at regular intervals, processes, them and posts on a social networking site using a set of plug-in extensions.

2) Image Acquisition Plugin: This is a plugin that can extract an image from a video using a pull protocol. The video maybe from a file (avi, mpg) or from a video camera such as a webcam. This plugin must bind to the extension point provided by the middleware plugin. Typicall, the image extracted is of type BufferedImage. One may use Xuggle or v4l4j. The programmer will have to test both libraries and chose the one appropriate for the task.

3) Image Analysis Plugin: The BufferedImage captured must be processed by a library for image processing. There are many options. BoofCV is a pure Java library that has functionality to detect lines, SURF features, etc. The other options are OpenCV, IVT which are far more advanced C++ libraries with many features. Therefore, the programmer may have to use JNAerator or SWIG to call these libraries (that are locally installed first so there is a platform-dependent dependency).

4) Reasoning Plugin: The image analysis plug-in determines interesting features in the sequence of images. However, these features must be mapped to semantics or they should be given meaning. The features and their properties can be analyzed using an imperative program in Java to create a summary with an image and some text. One may also use a declarative constraint satisfaction engine such as Alloy that can map features to a summary.

5) Social Network Plugin/Website Plugin: Finally, the student has to publish the summary of important images or an important image with perhaps some text to a website, blog or a social network. The student can choose for instance the Facebook RestAPI or Java Blogger API.

The programmer has liberty provide variation in video source, type of image analysis, different useful reasoning strategies type of posts generated for a blog.