VisRec JSR #381 Review

May 15, 2019
Agenda

• Goals
• JSR Process
• Implementation Notes
• Issues
• Questions, Discussion, Next steps
Agenda

• Goals
• JSR Process
• Implementation Notes
• Questions, Discussion, Next steps
Overall Goal

Promoting Java as a first-class citizen in AI/ML

Create a high-level standard API for object recognition using machine learning that is familiar to and useful for Java Developers
Why It’s Important for Java SE and Devs

- Machine Learning - a huge industry trend
- Visual Recognition (VisRec) - important subset of ML
- Java ML APIs need to be “Java Developer Friendly”
- Standard APIs offer portability and maintenance benefits

- High-level abstractions for sustainable development of products
- Protect developers from lower-level changes (and provide hooks allowing lower-level access)
- Building custom ML models/Image Classifiers (not just using pre-trained Classifiers)
Issues with Existing Offerings

- Disparate OSS/proprietary ML engines and toolkits
- Different image classes, algorithms and implementations, often with native dependencies
- Each has different set of APIs
- Reduced Portability
  - Image Recognition Apps
  - Lower-level Bitmap, Image, etc, pixel-level manipulation
- Some Toolkits very complex for Avg Java Developer
- Most Toolkits are not Java-friendly (C flavor)
Agenda

• Goals
• **JSR Process**
• Implementation Notes
• Questions, Discussion, Next steps
History (necessary?)

• List the significant dates in the history of the JSR.
  – Submittal: April 2017
  – EDR: June 2018
  – Public Review: Early June 2019
Technical scope and features

• Provide a high-level summary of technical features.
  – No more than 2 or 3 slides.
  – 1 slide: architecture
  – 2 slide: design - class diagram
  – 3 slide: example usage
  – 4 slide: comparison with existing libs

Zoran - TBD
The Expert Group

- Zoran - University of Belgrade
- Sandhya - (former) IBM, (current) Microsoft
- Frank - NYJavaSIG

- Status meetings (zoom) every Tuesday
  https://groups.io/g/visrec/topics
- Groups.io mailing list (visrec) and calendar
Contributors and Advisors

- Constantin Drabo
- Amit Nagesh
- Marissa Staller
- Eric Bruno
- Anakar Parida
- Jyoti Buddha
- Guillaume Laforge
- Ed Burns
- James Weaver
Other Docs, Presentations, etc

• Examples - 4-5 working examples
  https://github.com/JavaVisRec/jsr381-examples
• Getting Started Document
• JavaOne/CodeOne panel
  Heather/Sandy/Frank/EdBurns
• Intro to ML for Java Devs - Zoran/Frank - CodeOne
• Visual Recognition - Sandy/Frank - Devoxx US
Overview

The VisRec API JSR #381 is a software development standard recognized by the Java Community Process (JCP) that simplifies and standardizes a set of APIs familiar to Java developers for classifying and recognizing objects in images. There are two types of Java developers that may be interested in VisRec JSR #381: application developers interested in
Collaboration with Community Groups

• Kevin - In contact with NLJUG (Dutch JUG) to organize sessions nationwide to adopt the JSR once there are multiple visual recognition examples implemented using the API and RI.
• Frank - NYJavaSIG - waiting for 1.0 to actively engage
Agenda

- Goals
- JSR Process
- Implementation Notes
- Questions, Discussion, Next steps
Implementations

• How many implementations (apart from the RI) exist?
  – One more in progress: Neuroph educational neural network framework with support image recognition
Schedule

- June 1 - Beta release
- Dec - 1.0 release
RI and TCK development

• The TCK and RI are being developed simultaneously in a TDD (test-driven development) working environment as much as possible to keep the RI compliant with the TCK at any time.

• The API, RI, TCK are being developed by two active committers of which one is a Spec Lead of the JSR:
  – Zoran Sevarac (Spec Lead)
  – Kevin Berendsen (Contributor)
RI and TCK development

- TCK Runner (consists of the TCK and RI):
  - [https://github.com/JavaVisRec/jsr381-tck-ri](https://github.com/JavaVisRec/jsr381-tck-ri)

- Source-code repositories:
  - API: [https://github.com/JavaVisRec/visrec-api](https://github.com/JavaVisRec/visrec-api)
  - RI: [https://github.com/JavaVisRec/visrec-ri](https://github.com/JavaVisRec/visrec-ri)
  - TCK: [https://github.com/JavaVisRec/visrec-tck](https://github.com/JavaVisRec/visrec-tck)
  - Examples: [https://github.com/JavaVisRec/jsr381-examples](https://github.com/JavaVisRec/jsr381-examples)
RI and TCK development

- Snapshots published in Sonatype:
  - API: https://oss.sonatype.org/#nexus-search;quick~visrec-api
  - RI: https://oss.sonatype.org/#nexus-search;quick~visrec-ri
Participation and Transparency

• JSR page on JCP.org

• JSR project website
  https://github.com/JavaVisRec
Agenda

• Goals
• JSR Process
• Implementation Notes
• Questions, Discussion, Next Steps