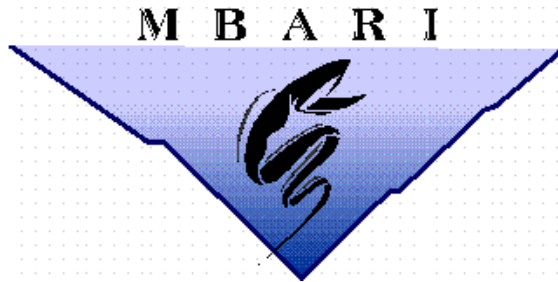


The MBARI Automated Video Event Detection Project



AVED Classifier User Interface

Software Requirements Specification

Version 0.3

Revision History

- 0.1 10-Aug-05 - DC initial draft
- 0.2 August 18, 2005 DC revision based on meeting with Brian and Andrew
- 0.3 September 1, 2005 DC minor revision to based on email between Brian and Andrew

Classifier Overview

The core classifier code is written in matlab. It is based on a test program given to us by student Marc'Aurelio Ranzato from Caltech. A JNI interface has been written to access the classifier library from Java.

There are three steps to using the AVED classifier: collection, validation, and training and testing.

The foundation of the classifier is a collection of images files. This collection of image files constitutes a "class" which typically represents a single animal in its various sizes and orientations, .e.g. a Rathbunaster class, Poeobius class, etc.

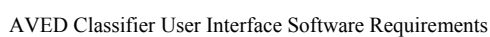
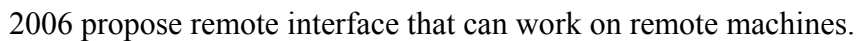
In the collection step parameters about each image are collected and stored in a matlab file for use in the validation and training phases. These stored parameters are used later in the validation or training phase to create a model of the animal. Out of all the phases, the collection phase takes the longest time to execute and should be run after adding or removing images from a given class. This step requires all the images of a given class be in the same directory on *Nanomia*.

To test how well a class performs as a discriminate for a given animal, validation of the class is done. This phase requires that the class be collected beforehand. The validation takes a 10% random sample from the class and runs the classifier against the class itself, and optionally one or more other discriminating classes, e.g. "Rathbunaster" and "Benthic" class. This should be done periodically to indicate how a class is performing.

Once validation is done, to use the classifier for quantitative work, a training set must be generated. A training set is two or more classes. A training set is first generated, then this training set is then tested against a collection of unknown images and an assignment of each the image is given.

The AVED classification metadata is stored on a database hosted by *Solstice*. Physical class image files and matlab-specific data are hosted on the 2TB RAID connected to the *Nanomia* server.

2005 propose stand-alone application only works on *Nanomia*.



2005 - Classifier standalone user interface requirements

Scope

This document covers only requirements for 2005 work.

General design

The software is to be designed and implemented in modern, maintainable technology, such as an object oriented programming language (OOP), which is supported over the useful life of the system. Software systems are to be well documented and to include written software test and maintenance procedures. There should also be maximum use of industry proven standards and technology and commercial off the shelf software (COTS).

Definitions

A class shall be defined as a collection of related images that represent a particular animal, collection of animals, or objects.

A training set shall be defined as more than one class.

Class preview

The interface shall include an alphabetical list that displays all available training classes. This list would include classes that may correspond to concepts in the VARS Knowledge Base (e.g. *Poecilius-merces*), or those that are more generic (e.g. “Benthic undefined” or “Benthic other”, which might represent a benthic animals that are not already defined by a class.)

The preview of a class shall include one representative image and a list of filenames or URL references for each image. (Would be nice to have thumbnail of all class images but not necessary).

Creating a new class

The interface shall allow the user to create a class that corresponds to a concept name in the VARS Knowledge Base, or information outside the knowledge base.

Rationale: Classes will typically come from the VARS KB, but are not limited to the Knowledge Base. For example, in our collaborations we use video from areas outside Monterey, with concepts that may not exist in our KB and our testing may require grouping animals in some way that does not correspond to a concept.

The user shall provide a general description of the class and an example image at the creation of a new class.

The name of the class shall be formatted with the concept name appended by “-” and a unique database ID and version number for the class.

The interface shall allow the user to select a local directory or files within a local directory to import images that make up the class.

Once a class is locked, the user cannot update the class collection and must create a new class.

Deleting classes and images

The interface shall allow a user to delete an entire class. This shall not be password protected, so anyone can delete a class.

The interface shall allow the user to tag individual images for deletion in a given class.

Note, even though images associated with a given class are deleted, they will still physically reside on nanomia and have a database reference if the image is cross-referenced by other classes.

Collecting a class

The interface shall allow one or more classes to be collected.

An indicator to display class collection is commencing shall be displayed.

While collection is commencing the interface shall tag the classes as not usable and not allow other operations that require the classes to begin.

Validating a class

The classification software shall include the ability to validate an individual class or collection of classes.

A minimum of TBD images shall be required to validate a class. For class with less than TBD images, the validation function shall be disabled. *Rationale: the current classifier crashes when testing a class with a small number of images. The minimum number of images seems to depend on the size of the animals and/or amount of feature detail in the animal. Classes that represent larger animals like the Rathbunaster require fewer images than the small Poeobius. This may be fixed in later versions of the classifier library, but for now is an outstanding problem.*

The results of the class validation shall display both percent correct and exact numbers of classification results, for example. 95% correct classification or 3812 out of 4013 correctly classified as Rathbunaster.

Note, in the classifier library, the validation function automatically generates a training set for the class. This operation is handled by the classifier matlab library and is hidden from the user. If changes are made to one or more classes in the training set (e.g. by adding/deleting), regeneration of the training set is required,

Generating a training set

To generate the training set, the interface shall allow the user to select from available class collections in an alphabetical drop-down or scrolling type list.

Testing a class against a training set

To test against a training set, the interface shall allow a user to first select an available training set.

Once a training set is selected, the interface shall allow the user to select a local directory with test images. The directory listing shall show images in PNG or JPG format.

Once testing is started, a progress bar shall display the status of the testing.

A summary of results of the testing class validation shall display both overall percent and exact numbers of classification results, for example. 15% or 150/1000 classified Rathbunaster, 10% or 100/100 as Unknown, and 75% 750/1000 as Other.

Details results shall be displayed in a grid format with the headers: file name, probability and class assignment. (Would be nice to have a thumbnail of test input images too, but not required).

User profiles

The user interface shall store all current settings into a user profile to be reloaded on startup.

Directory Defaults

The interface shall allow the user to select default locations for testing and training results.

Saving results

The interface shall allow the user to save the training and testing results to XML or comma delimited format.

If a file is not specified to save the results to, the UI shall prompt the user to select a file.