Our implementation of the DUAL classifier is similar to that in (Qiao, Davois 2006). As the name implies the Dual classifier makes use of two classifiers where each classifier makes its own separate prediction (assignment of a label to the unknown example). The results of these two classifiers are then fused together for a finale prediction. The procedure for this fusion is simple; if they both agree then the agreed upon prediction is assigned otherwise the label “Other” is assigned.

The two classifiers that were used are both implementations of a Support Vector Machine (SVM) (Christopher J. C. Burges 1998). One is a traditional implementation of a one-vs-one where a common set of features and parameters are used for all the class pairs. The features used are the result of a WRAPPERS based Feature Selection process (John Ron Kohavi 1997). The other makes use of pairwise feature selection (K. Kramer 2011) where features and parameters are tuned for each class pair combination. Since the two classifiers are making use of different set of features they do not always agree on their predictions. When they do agree the probability that the prediction is correct is greater than when depending on the prediction of just one classifier. A detailed description of the pool of features used for both classifiers can be found in (Kramer 2010).

Another modification of the dual classifier from that implemented in (Qiao, Davois 2006) is the ability to predict on the agreement of a partial match. The idea is that classes have levels of discrimination such as crustacean\_copepod\_calanoid has three levels. Rather than expecting the two classifiers to agree upon the full name we allow for partial agreement. For example if classifier one predicts “crustacean\_copepod\_calanoid” and classifier two predicts “crustacean\_copepod\_copilia” the dual classifier would assign the prediction “crustacean\_copepod” which consists of the first two levels.

**Grade Training Model**

When computing classifier accuracy we normally compute accuracy by dividing the number of correct predictions by the total number of examples in the ground truth. To deal with the fact that our implementation of the dual classifier can make partial predictions we modified the grading to allow for partial credit. That is it will assign a number between 0.0 and 1.0 that reflects the number of levels of the known class that the predicted class matches.

Table below shows examples of credit assigned to different predictions

|  |  |  |
| --- | --- | --- |
| Known Class | Predicted Class | Credit |
| crustacean\_copepod\_copilia | crustacean\_copepod | 2/3 = 0.6666 |
| crustacean\_copepod\_copilia | crustacean\_copepod\_calanoid | 2/3 = 0.6666 |
| crustacean\_copepod\_copilia | Chaegnath | 0/3 = 0.0 |
| crustacean\_copepod\_copilia | crustacean\_ostracod | 1/3 = 0.3333 |
| crustacean\_copepod | crustacean\_copepod\_calanoid | 2/2 = 1.0 |
| Chaegnath | Chaegnath | 1/1 = 1.0 |

The PICES system is an application consisting of several programs that manage the plankton images that are stored in the open source database MySQL. The database consists of several tables that store the plankton images itself, computed feature data, collected instrument data (CTD, GPS), and current class assignments. This allows the user to more easily manage the 30+ million discrete plankton images. Images can be browsed by Cruise, Station, Deployment, Class, Depth, Size and other criteria. The interface to this database is through an application called PICES Commander. A list of the more important tables can be found in Table 1.

Table 1 List of the major tables in the PICES database.

|  |  |
| --- | --- |
| Classes | A list of the currently know classes(Labels) that can be assigned to each image. |
| Cruise | One entry for each cruise (In the case of this database there are three entries) |
| Station | One entry for each station, Keyed by Cruise and Statiom. |
| Deployment | One entry for each deployment of SIPPER. |
| SipperFiles | One entry for each SIPPER file, a deployment will consist of one or more SIPPER files. |
| Images | One entry for each mage extracted from SIPPER files, keeps track of where the image was located, thumbnail version of the image, label assigned with corresponding confidence value, user assigned validated label, |
| ImagesFullSize | A full size copy of each image is saved in this table. |
| InstrumentData | Snap shots at every 4096 scan lines (~ every 1/8 second) of collected instrument values are stored here. Keyed by the Sipper File and scan line allows for quick joining of instrument data with plankton images. |
| FeatureData | There will be one entry for each extracted image consisting of the feature values computed during image extraction from the SIPPER files. These are the feature values used by the classifiers to make predictions. |

Qiao, . Davis CS (2006) Accurate Automatic Quantification of Taxa-specific Plankton Abundance Using Dual Classification with Correction. Mar Ecol Prog Ser 306: 51-61

Christopher J. C. Burges 1998, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, June 1998.

John Ron Kohavi and H George, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1, pp. 273-324, December 1997.

K. Kramer, D. Goldgof, L. Hall, A. Remsen "Increased Classification Accuracy and Speedup Through Pair-wise Feature Selection for Support Vector Machines", IEEE Symposium Series on Computational Intelligence, pp. 318-324, Paris France, April 2011.

Kurt Kramer, Dissertation, System for Identifying Plankton form the SIPPER Instrument Platform. Tampa: University of South Florida, 2010.