DIRECTORY STRUCTURE

aifr.ClassificationAlgorithm

Description: interface for classifiers to implement:

public ClassificationResult<String> classify(FImage object)

public void train(List<? extends Annotated<FImage, String>> data)

Methods:

void train(HashMapSerializable<FloatFV, String> readFeatureVector)

inputs a HashMapSerializable<> and trains the classifier with the provided features..

aifr.Constant

Description: a configuration type file for storing constant filepaths and functions. It contains:

```
final public static String trainPath = "resources/CACD_Training";
final public static String testPath = "resources/CACD_Testing";
final public static String resultPath = "resources/results/";
final public static String classifierPathSVM = "resources/classifier/SVM.ser";
```

aifr.ExtractFeature

Description: interface for feature extractor to implement:

```
FloatFV extractFeature(DetectedFace face);
```

aifr.ExtractLocalLBPHistogram

Description: class to implement ExtractFeature which extracts the Local Binary Patterns from an image.

ScalingAligner<DetectedFace> lbpAlign = new ScalingAligner<DetectedFace>();

- Face aligner.

final private LocalLBPHistogram.Extractor<DetectedFace> lbpHist = new LocalLBPHistogram.Extractor<DetectedFace>(new ScalingAligner<DetectedFace>());

- feature extractor

Methods:

public static String getFeaturePath()

returns the feature path from *Constant* class

public FImage alignFace(DetectedFace face)

returns the FImage type face aligned w.r.t. eyes.

public FloatFV extractFeature(DetectedFace face)

returns the Feature Vector from the face as Local Binary Patterns.

aifr.HashMapSerializable

Description: class for HashMap to implement Serializable to save the object as feature vector.

aifr.KNearestNeighbor

Description: class to implement ClassificationAlgorithm interface which provides methods to apply the K Nearest Neighbor classifier on the feature vectors as a whole.

Methods:

public void train(List<? extends Annotated<FImage, String>> data)

trains the classifier with the input data as a List collection of Annotations.

public void train(HashMapSerializable<FloatFV, String> data)

trains the classifier with the input data as a HashMapSerializable collection of FloatFV – String maps.

public ClassificationResult<String> classify(FImage image)

returns the classification result as ClassificationResult class type from a FImage type input data to classify.

protected FloatFV getFeatureVector(FImage image)

returns the feature vector as FloatFV class type object from an FImage type input image.

aifr.ManageClassifier

Description: class for handling the classifiers, to save, read, and train.

Methods:

public boolean saveClassifier(ClassificationAlgorithm ca)

returns a Boolean for classifier instructing about the dump of the feature vector.

public ClassificationAlgorithm trainClassifier(ClassificationAlgorithm classifier)

inputs a ClassificationAlgorithm as a classifier, trains it with feature vectors, and returns the same. It extracts the features at runtime from the Training samples.

public ClassificationAlgorithm trainClassifier(ClassificationAlgorithm classifier, String featurePath)

inputs a ClassificationAlgorithm as a classifier, trains it with feature vectors provided as another argument, and returns the same. It reads the features directly from the memory dump.

public ClassificationAlgorithm readClassifier()

returns the classifier as ClassificationAlgorithm object by reading from the memory dump.

aifr.ManageFeatureVector

Description: class for handling the feature vectors, to save, read, and train.

Methods:

public boolean saveFeatureVector(HashMapSerializable<FloatFV, String> hM)
returns a Boolean for feature vector instructing about its dump.

public ClassificationAlgorithm trainClassifier(ClassificationAlgorithm classifier)

inputs a ClassificationAlgorithm as a classifier, trains it with feature vectors, and returns the same. It extracts the features at runtime from the Training samples.

public ClassificationAlgorithm trainClassifier(ClassificationAlgorithm classifier, String featurePath)

inputs a ClassificationAlgorithm as a classifier, trains it with feature vectors provided as another argument, and returns the same. It reads the features directly from the memory dump.

public HashMapSerializable<FloatFV, String> readFeatureVector()

returns the feature vector HashMap collection as HashMapSerializable object by reading from the memory dump.

public HashMapSerializable<FloatFV, String> getFeatureVectorMap(ExtractFeature extractor)

returns the feature vector HashMap collection as HashMapSerializable object by extracting the features at runtime from training files.

aifr.OrganizeLocalLBPHistogram

Description: class for handling the Local Binary Pattern features.

Methods:

public static void fetchAndSave()

function that extracts the features from ExtractLocalLBPHistogram class and dumps the classifier into the memory.

aifr.PlayNeighbors

Description: class for training the K Nearest Neighbor classifier.

Methods:

public static void trainKNN()

trains the kNN classifier by extracting the features from the training samples.

public static void trainKNN(String featurePath)

trains the kNN classifier by reading the features from the memory dump.

aifr.PlaySupportVectors

Description: class for training the SVM classifier.

Methods:

public static void trainSVM()

trains the kNN classifier by extracting the features from the training samples.

public static void trainSVM(String featurePath)

trains the kNN classifier by reading the features from the memory dump.

public static void crossValidate(String featurePath)

reads the feature vector from the memory dump and trains the SVM classifier and performs the cross validation for the best parameters.

public static svm_model loadModel()

loads the model into svm_model object from the SVM classifier memory dump.

aifr. PrintableClassificationResult

Description: class for handling the classification result and saving them to a file.

Methods:

public String bestResult()

returns the best result class as a String object.

public String resultList()

returns the complete result class as a String object.

aifr.ReadImages

Description: class for reading the images to array of bytes from a file.

Methods:

public void display(FImage img)

displays the image in a window reading as a FImage object.

public MBFImage readImage(String img)

returns the MBFImage object by reading the image from the filepath provided as a String object.

public MBFImage readImage(FImage img)

returns the MBFImage object by reading the image as a FImage object.

public DetectedFace extractFace(MBFImage img)

returns the face portion coordinates as a DetectedFace object by recognizing the face in the provided image.

aifr. ReloadedSVM

Description: class for training and recognizing the faces from the testing directory.

It read the classifier object from the memory dump and applies it to the testing images for their prediction, which saves the result to a text file in 'resources/results' directory.

aifr. SupportVectorMachine

Description: class to implement ClassificationAlgorithm interface which provides methods to apply the Support Vector Machine classifier on the feature vectors as a whole.

Methods:

protected svm_model svmTrain()

trains the SVM classifier from the LBP feature vector and returns the trained model as svm_model object.

protected double[] svmPredict(double[][] xtest, svm_model model)

returns the classified result as a double array object by applying the SVM classifier on the 'xtest' double feature vector array taking in the classifier as svm_model object.

public ClassificationResult<String> classify(FImage image)

returns the classification result as ClassificationResult class type from a FImage type input data to classify.

public ClassificationResult<String> classify(FloatFV featureVector)

returns the classification result as ClassificationResult class type from a FloatFV Feature Vector type input data to classify.

protected FloatFV getFeatureVector(FImage image)

returns the feature vector as FloatFV class type object from an FImage type input image.

public void train(HashMapSerializable<FloatFV, String> readFeatureVector)

trains the SVM classifier with the input data as a HashMapSerializable collection of FloatFV – String maps by reading from the memory dump.

public void cross_validation(HashMapSerializable<FloatFV, String> readFeatureVector)

reads the feature vector from the memory dump and trains the SVM classifier and performs the cross validation for the best parameters.

public static svm_model loadModel()

loads the model into svm_model object from the SVM classifier memory dump.

aifr. TraverseDirectory

Description: class for traversing the directory nodes and indexing them for reading the files for training of the classifier.

All the traversal is performed in the constructor itself which takes the directory name as the parameter, it sets:

the filepaths to an ArrayList collection,

the fileclass names to an ArrayList collection,

the filemap to HashMap collection as ClassName to Path of the sample.

aifr.Utilities

Description: class for handling the classifiers' performance based on their type.

Methods:

public static void runClassifier(ClassificationAlgorithm classifier, String classifierName, String classifierPath, Object extractor)

runs the classifier as provided in parameter ClassificationAlgorithm, classifierName for setting up the text file, classifierPath for reading the classifier object from the memory dump, extractor for specifying the FeatureExtractor type.

application.App

Description: class for running the Graphical User Interface of the recognition panel for test subjects.

application.AppController

Description: class for handling the controls and events of the test subject.

Methods:

public void initialize(URL url, ResourceBundle rB)

initializes the user interface with different components' state.

public void initClassifier()

initializes or loads the classifier into the program for the recognition of the test subject at the loading of the application.

public void browseImage(ActionEvent e)

browses an image file and displays it in the 'browseImage' ImageView frame.

public void processImage()

processes the test image by reading and extracting the features and applying it to the classifier. It also sets the UI values to show the current progress of the process.

public void reset(ActionEvent e)

resets the complete application to its initial state.

public void exit(ActionEvent e)

exits the application by shutting down all the subprocesses.

app.App.fxml

XML file that contains the structure, alignment and view of the User Interface.