**READ ME**

**Folders**

NEMESYSCO CD – This folder contains all the files from the Nemesysco QA5 SDK.

Real-time Recognition – The main program is located in this folder. For more information continue reading bellow.

Tables – Contains all the results from the databases tested with the program.

**Required Software**

1 - Install Nemesysco QA5 Win32 SDK.

2 - Download and install the .NET SDK from the following website. If you already have it installed, just run “repair” on the installation. <http://www.microsoft.com/downloads/details.aspx?FamilyID=fe6f2099-b7b4-4f47-a244-c96d69c35dec&displaylang=en>

**Running the program and changing the code**

1 - Make sure that the Nemesysco verification USB is plugged in and lit red.

2 - Open the file “…\nmsVoiceAnalysisLibrary\nmsVoiceAnalysisLibrary.sln” in Microsoft Visual Studio 2008 (for some reason, **the code does not work properly in 2010 version**). Change whatever code you need, and Rebuild the program. Remember that the main chunk of the code is in nmsFunctions.cs file.

3 - After the alterations, open the .NET SDK command prompt from Start → Microsoft .NET Framework SDK 2.0 → SDK Command Prompt. Change the directory to the one that contains your .dll file, you should type something like “cd …\nmsVoiceAnalysisLibrary\bin\Debug\”. Now, insert the command “regasm nmsVoiceAnalysisLibrary.dll /tlb:nmsVoiceAnalysisLibrary.tlb /codebase” without quotes in the command prompt.

4 - Finally, open “…\nmsVoiceAnalysisRunCPP\nmsVoiceAnalysisRunCPP.sln” and run the program in C++.

OBS.: The program may not work properly if the microphone sensitivity is not adjusted correctly. Additionally, you can increase or reduce the value of the variable backgroundLevel, in the nmsFunctions.cs file, to adjust the program noise sensitivity.

**Changes in the code**

The first code was working fine; however, the classification algorithm used was inefficient. The original classification model was based in only 4 parameters and had only 4 emotions as outputs. In order to assure the necessity of improving the algorithm, the model was tested on 2 databases. As a result, only 32% of the segments were correctly classified. Therefore, a new classification model should be implemented to increase the algorithm accuracy. In order to identify the new model, the WEKA software (<http://www.cs.waikato.ac.nz/ml/weka/index.html>) was used together with the data collected from the databases. According to WEKA software, the best results were found with the logistic regression model.

The final model is based in logistic regression with ridge estimator. The new classifier model uses 19 parameters and has 6 emotions as outputs with 75% of average accuracy in 3 different databases.

**Understanding the code**

The main code remain almost the same, all the changes were done in the classification model. The current model was based in the WEKA java code for logistic regression (available at <http://grepcode.com/file/repo1.maven.org/maven2/nz.ac.waikato.cms.weka/weka-dev/3.7.5/weka/classifiers/functions/Logistic.java>). For the new model, 8 new classes were added to the program: Corematrix, EigenvalueDecomposition, Logistic, LUDecomposition, Maths, Matrix, Optimization, and QRDecomposition. With the exception of Logistic, all these classes are used for manipulating, filtering and organizing the data. The Logistic class builds the logistic regression model and calls the remaining classes. Now, the code has 6 emotions as outputs, each of them represented by a number: 1=anger, 2=disgust, 3=anxiety, 4=sadness, 5=neutral, 6=happiness. The regression model is build based on the file ‘traindata.txt’ which contains 70% of the data of each datasets used. Additionally, the program has some functions which implements old models tested, for example KNN. These functions can be disregarded, since their results were not as good the ones with logistic regression.

**Databases**

In order to train the model, 3 databases were used.

SAVEE Database, available at

<http://kahlan.eps.surrey.ac.uk/savee/>

Berlin Database of emotional Speech, available at

<http://pascal.kgw.tu-berlin.de/emodb/docu/>

Emotional Prosody Speech and Transcripts, samples available at

<http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2002S28>

**Logistic Regression with Ridge Estimato**r

Briefly, the logistic regression is a regression analysis used for predicting the outcome of a dependent variable with respect to predictor variables. The logistic regression model with ridge estimator was proposed by le Cessie and van Houwelingen, in their paper Ridge Estimators in Logistic Regression.

According to the logistic model, if there are *k* classes for *n* instances with *m* attributes, the parameter matrix *B* to be calculated will be an *m\*(k-1)* matrix.

The probability for each class *j* with the exception of the last class is

And the probability for the last class is

And the negative multinomial log-likelihood is

In order to find the matrix *B* for which *L* is minimised, a Quasi-Newton Method is used to search for the optimized values of the *m\*(k-1)* variables.

For more information see, le Cessie, S., van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics. 41(1):191-201.