There are many applications for an acoustic sensor that can automatically recognize when a particular sound source passes within proximity of the device. Example sound sources of interest include motors, generators, any wheeled craft, or other rotating machinery. The sound pressure can be converted into a numeric sequence with a microphone, amplifier and analog to digital converter that is commonly available on any laptop computer.

A Discrete Fourier Transform (DFT) is a convenient way to evaluate the signal characteristics in a domain that facilitates separation of signal components from interfering noise. The DFT transitions the voltage versus time waveform into the frequency domain, to help the recognition of signal features. The goal is to diagnose the origin of the signal based upon the presence of signal characteristics such as the frequency and number of harmonic components received and their ratio of powers. A typical signal from an engine or other rotating machinery consists of multiple narrow bandwidth spikes in the frequency-power response that occur in specific parts of the spectrum and at specific frequency separations. These harmonics can be automatically detected and stored into a data base according to the specific source type.

The purpose of this project will be to validate an automatic classification method applied to acoustic signals made up of multiple harmonic components. Instead of sampling real audio, a digital simulator will be developed that produces samples of signal amplitude sequences associated with various sources. These simulated signal vectors will be processed by the automatic algorithm to evaluate performance (example probability of correct classification).

Specifically, there will be two central programs. The first program will be called training.py. This training script will simulate several samples of signals from different classes. An example of a class is a signal with some number of sine waves at different frequencies. Different samples from that class might be that same signal with the same frequencies with different signal-to-noise ratios (SNR). The difference between the signals with different SNR’s is just the amount of Gaussian noise added. The training program will input each of these samples into a Feature Extraction routine. This function will output signal features that are then stored in a file. This file of features will be known as the dictionary.

The actual classification function will be executed in the second program, called tester.py. This tester will read the features from the dictionary file and create a mean and standard deviation for each class. The tester will also generate another sample of a signal from one of the classes at some specified SNR. Perhaps the SNR is a random variable. The tester will then generate features for the new unknown signal and compare the features to the mean and standard deviation of each class. The comparison algorithm is the weighted mean square error (MSE) between the unknown features and the features of each class. The weight applied to each feature can be the sample variance of the feature amongst the samples within a class. The mean features generating the smallest MSE is chosen as the class most similar to the unknown signal. The tester can repeat the testing process many times so that a probability of correct classification and probability of incorrect classification can be computed for each class.

**Potential Audience/Consumers**

The potential audience for such research could be government organizations such as the US Army and the Marine Corps, as well as developers of military technology such as Lockheed Martin and Raytheon Company. In addition, city traffic monitors could use the application to automatically find out how many large trucks versus little cars transit over a bridge or alongside an important thoroughfare.

**Projected Timeline and Deliverables**

14 Week Plan: 4 tasks

Task 1: Algorithmic design and documentation (2 weeks)

Task 2: Software implementation (6 weeks)

Task 3: Software testing and performance evaluation (4 weeks)

Task 4: Documentation of results including final report (2 weeks)

The deliverable will be a report describing the algorithmic approach, software implementation, test results, and overall conclusions. Also all the code will be uploaded to the RCOS site.

**Team Members**

For this project, I believe that for 14-15 weeks such an objective is manageable for one person to accomplish. However, a mentor for this work is desired. I envision this work to be worthy of a two-credit independent study effort.

**Work Distribution**

N/A