A METHODOLOGY AND A SYSTEM FOR ADAPTIVE, INTEGRATED SPEECH AND IMAGE LEARNING AND RECOGNTION

Akbar Ghobakhlou and Nikola Kasabov

Knowledge Engineering and Discovery Research Institute
Auckland University of Technology, Auckland, New Zealand
akbar@aut.ac.nz , nkasabov@aut.ac.nz

ABSTRACT
The paper presents a novel approach towards building adaptive speech processing systems based on the evolving connectionist systems paradigm (ECoS). The methodology proposed in this paper is applied on speech and image recognition and person identification based on speech and image. Adaptive connectionist classifier (ACC) is an implementation of the ECoS paradigm. The process of learning and recognition in ACC is achieved through a local adaptation of each module in interaction with the other modules. They can accommodate new input data and new classes through local element tuning. New connections and neurons are created during the adaptive learning process of the system. Experiments are conducted to illustrate this concept. It is demonstrated that a system can adapt to new data and add new outputs at any time of its operation without having to build the network from "scratch". The system is robust to forgetting when new output classes are added. The methodology is tested on several case studies and results are reported.

Keywords: Evolving connectionist systems (ECoS), adaptive connectionist classifier (ACC), adaptive speech recognition

1. INTRODUCTION

Despite of the advances in speech and image recognition methods and systems and of the connectionist learning models there are no efficient methods and systems that allow for incremental, on-line adaptive learning of new spoken words, new accents and pronunciations, for learning new images, and for adaptive integration of speech and image modules. Several prior publications have introduced methods for speech and image integration but not in a way that the whole system can be adapted incrementally, on-line, at all levels of its functioning.

Various connectionist techniques have been suggested and experimented for incremental and on-line learning from streams of data (Moody and Darken, 1989, Platt, 1991, Rosipal et. al., 1997, Saad, 1999, Schaal, and Atkeson, 1998, Blanzieri and Katenkam, 1996, Bottu and Vapnik, 1992, Edelman, 1992, Freeman and Saad, 1997, Fritzke, 1995, Heskes and Kappen, 1993). Among them are evolving connectionist systems (ECoS) (Kasabov, 2000, Kasabov, 2002, Kasabov et. al., 2000). These techniques have been further developed here for the purpose of adaptive, integrated speech and image processing.

The methodology proposed here and applied on speech and image separately, and in their integration, considers the process of learning and recognition as an adaptive one in its entirety, which is achieved through a local adaptation of each module in an interaction with the other modules (Kasabov, and Ghobakhlou, 2003). The adaptation includes:

- the level of pre-processing
- the level of end-point detection for word and image selection from a continuous signal
- the level of noise detection and noise suppression
- the level of modelling and pattern recognition (classification)
- the level of language (context) modelling and integration with some other sources of information

Consequently, the paper introduces a novel methodology for adaptive speech and image learning and recognition systems that include the following novel methods:

- a methodological framework and specific adaptive neural network for adaptive speech and image recognition systems
- an adaptive system for isolated word recognition
- a method for adaptive end-point detection for word and image selection from a continuous signal
- an adaptive system for image recognition
- a method for adaptive model integration

Based on the methodology, the paper also introduces a novel system for an adaptive speaker identification (speech input only) or person identification (speech and image).

2. METHODOLOGICAL FRAMEWORK

2.1. Adaptive Speech and Image Recognition System

The general framework for adaptive speech and image recognition systems consist of several modules. The central principle in each module is to utilize adaptive systems within each module. Figure 1 illustrates an overview of a framework for adaptive speech and image recognition systems. Inputs to the system come from two primary sources of speech and image. Two separate specialised adaptive modules process the incoming inputs. The inspiration of such specialised modules initiated from the human brain (Arbib, 2003, Amari, and Kasabov, 1998). The outputs from each module are further processed in another module (integration/ language module) to generate final outputs. The adaptive structure of the framework allows the system to correct itself at any level of its operation through adaptation process. It mimics the ability of the human brain to adapt locally (only a single module or a group of neurons adapt with every new input) but the effect is global (these neurons have connections and interact with other parts of the brain.

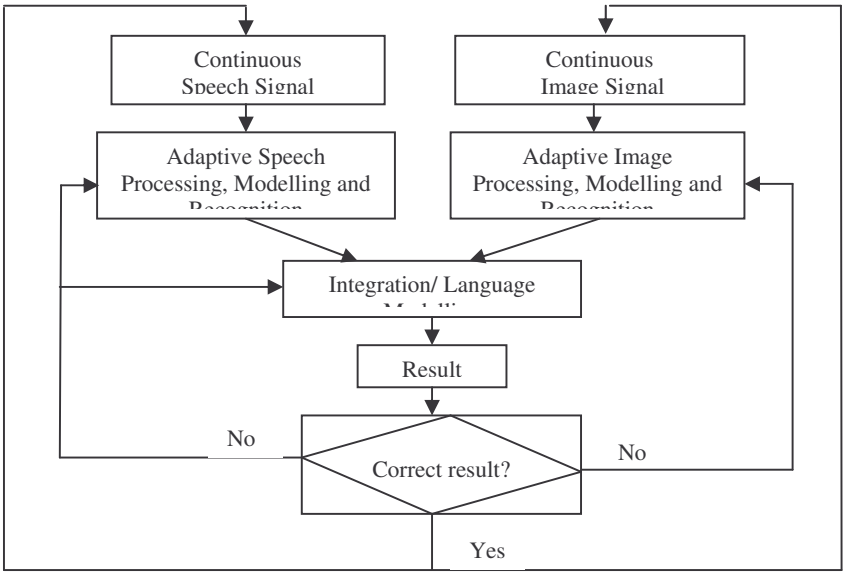

Figure 1: An overview of the proposed framework for adaptive speech and image recognition

The adaptation process in the system shown in Figure 1 may be achieved with the use of different adaptive learning techniques. Here we propose a simplified version of the evolving the Evolving Connectionist System (ECoS) paradigm and develop a specific neural network for the purpose of fast adaptive learning, called Adaptive Connectionist Classifier (ACC). ACC is used as a specific model for the implementation of all the methods and systems described in this paper thus making the framework an integrated and uniformly implemented. First the principles of ECoS are presented here:

2.2. ECoS

ECoS was developed to address several of the perceived disadvantages of traditional connectionist systems (Kasabov, 2000, Kasabov, 2002). It is a structurally evolving paradigm that modifies the structure of the network as training examples are presented. Although the seminal ECoS architecture was the Evolving Fuzzy Neural Network (EFuNN) (Kasabov, 2000) several other architectures have been developed that utilise the ECoS paradigm. These include the minimalist Adaptive Connectionist Speech and Image Classifier (ACC, see Section 2.3). The general principles of ECoS are:

1. ECoS learn fast from a large amount of data through one-pass training;
2. ECoS adapt in an on-line mode where new data is incrementally accommodated;
3. ECoS "memorise" data exemplars for a further refinement, or for information retrieval;
4. ECoS learn and improve through active interaction with other systems and with the environment in a multi-modular, hierarchical fashion;

The advantages of ECoS are such that they avoid several problems associated with traditional connectionist structures. They are hard to over-train, they learn quickly, and they are far more resistant to catastrophic forgetting than most of the other models. Conversely, it is these advantages that cause some of their disadvantages. Since they deal with new examples by adding nodes to their structure, they rapidly increase in size and can become unwieldy if no aggregation or pruning operations are applied. They also have some sensitivity to their parameters, which require constant adjustment for optimum performance.

An ECoS network always has at least one evolving layer. This is the layer that will grow and adapt itself to the incoming data, and is the layer with which the learning algorithm is most concerned. The meaning of the incoming connections, activation and forward propagation algorithms of the evolving layer all differ from those of classical connectionist systems. The ECoS learning algorithm can be found in (Kasabov, 2002).

2.3. Structure of the Adaptive Connectionist Classifier (ACC)

The **A**daptive **C**onnectionist **C**lassifier (ACC) is a simple implementation of the ECoS paradigm. It was created as a simpler version to Evolving Fuzzy Neural Network (EFuNN), based on the rational that for some situations, fuzzified inputs are not only unnecessary but harmful to performance. There are several advantages to using ACC over EFuNN. Firstly, their much simpler architecture means they are easier to understand and analyse. Secondly, their unfuzzified input space is of a lower dimensionality than a corresponding EFuNN (which always have at least two condition nodes attached to each input node, thereby doubling the dimensionality of the input space), which allows the ACC to model the training data with fewer nodes in the evolving layer than an equivalent EFuNN.

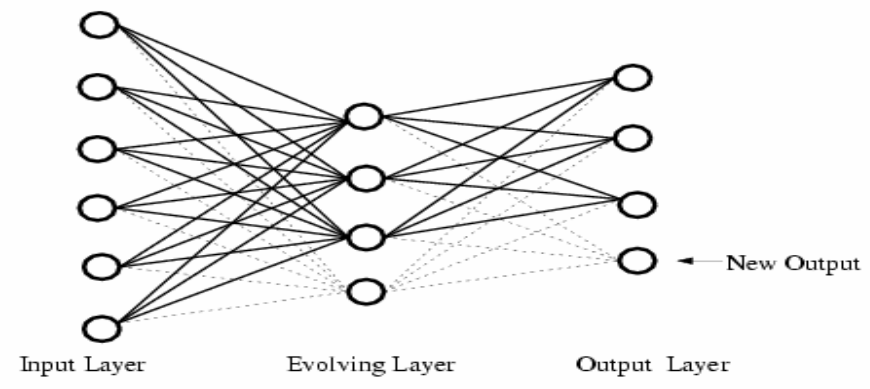

Figure 2: A simplified graphical representation of the ACC architecture

Figure 2 is a simplified graphical representation of the ACC architecture. An ACC consists of three layers of neurons, the input layer, with linear transfer functions, an evolving layer based upon the rule

layer of the EFuNN model, and an output layer with a simple saturated linear activation function. The evolving layer activation is calculated as with the EFuNN, with the exception of the distance measure D_n being calculated as the normalised Hamming distance, as shown in Equation 1:

$$D_n = \frac{\sum_i^I |E_i - W_i|}{\sum_i^I |E_i + W_i|} \quad (1)$$

where: I is the number of input nodes in the ACC, E is the input vector, and W is the input to evolving layer weight matrix.

The ACC architecture is similar to the Zero Instruction Set Computer (ZISC) architecture (ZISC, 2002). However, ZISC is based on RBF ANN and requires several training iteration over input data.

Aggregation of nodes in the evolving layer can be employed to control the size of the evolving layer during the learning process. The principle of aggregation is to merge those nodes which are spatially close to each other. Aggregation can be applied for every (or after every n) training examples. It will generally improve the generalisation capability of ACC. The aggregation algorithm is as follows:

FOR each rule node $r_{j,} j = 1:n,$

Where n is the number of nodes in the evolving layer and $W1$ is the connection weights matrix between the input and evolving layers and $W2$ is the connection weights matrix between the evolving and output layers.

- find a subset R of nodes in evolving layer for which the normalised Euclidean distances $D(W1_{rj}, W1_{ra})$ and $D(W2_{rj}, W2_{ra})$ $r_j, r_a \in R$ are below the thresholds W_{thr}.
- merge all the nodes from the subset R into a new node r_{new} and update $W1_{r_{new}}$ and $W2_{r_{new}}$ using the following formulae:
-

$$W1_{r_{new}} = \frac{\sum r_a \in R(W1_{r_a})}{m} \quad (2)$$

$$W2_{r_{new}} = \frac{\sum r_a \in R(W2_{r_a})}{m} \quad (3)$$

where m denotes the number of nodes in the subset R.

- delete the nodes $r_a \in R$

Node aggregation is an important regularization that is not present in ZISC.

It is highly desirable in some application areas, such as speech or image recognition systems. In speech recognition, vocabulary of recognition systems needs to be customised to meet individual needs. This can be achieved by adding words to the existing recognition system or removing words from existing vocabulary.

This study is an extension to our pervious work which introduced an on-line output expansion algorithm for EFuNN. ACC is also suitable for this task because it uses local learning which tunes only the connection weights of the local node, so all the knowledge that has been captured in the nodes in the evolving layer will be local and only covering a "patch" of the input-output space. Thus, adding new classes or new inputs does not require re-training of the whole system on both the new and old data as does with traditional neural networks.

The task is to introduce an algorithm for on-line expansion and reduction of the output space in ACC. As described in Section 3 the ACC is a three layer network with two layers of connections. Each node in the output layer represents a particular class in the problem domain. This local representation

of nodes in the evolving layer enables ACC to accommodate new classes or remove an already existing class from its output space.

In order to add a new node to the output layer, the structure of the existing ACC first needs to be modified to encompass the new output node. This modification affects only the output layer and the connections between the output layer and the evolving layer. The graphical representation of this process is shown in Figure 2. The connection weights between the new output in the output layer and the evolving layer are initialised to zero (the dotted line in Figure 2). In this manner the new output node is set by default to classify all previously seen classes as negative. Once the internal structure of the ACC is modified to accommodate the new output class, the ACC is further trained on the new data. As a result of the training process new nodes are created in the evolving layer to represent the new class.

The ACC Output Expansion Algorithm

The process of adding new output nodes to ACC is carried out in a supervised manner. Thus, for a given input vector, a new output node will be added only if it is indicated that the given input vector is a new class. The output expansion algorithm is as follows:

FOR every new output class,

- insert a new node into the output layer.

FOR every node in the evolving layer $r_i, i = 1:n$,

where n is the number of nodes in the evolving layer.

- modify $W2$ the outgoing connection weights from the evolving to output layer by expanding $W2_{i,j}$ with set of zeros to reflect the zero output.

This is equivalent to allocating a part of the problem space for data that belong to new classes, without specifying where this part is in the problem space.

The ACC Output Deletion Algorithm

This is an inverse process of the output expansion algorithm introduced in Section 4.1. The process of removing an output class from ACC is performed in supervised manner. It only affects the output and evolving layer of ACC architecture.

FOR every output class o to be removed,

- find set of nodes S in the evolving layer which are committed to that output o
- modify $W1$ the incoming connection from input layer to evolving layer by deleting $S_i, i = 1:n$, where n is the number of nodes in the set S committed to output o
- modify $W2$ the outgoing connection weights from the evolving to output layer by deleting output node o

The above algorithm is equivalent to dis-allocating a part of the problem space which had been allocated for the removed output class. In this manner, there will be no space allocated for the deleted output class in the problem space. In other words the network is unlearning a particular output class.

3. AN ADAPTIVE SYSTEM FOR ISOLATED WORD RECOGNITION

In speech recognition systems, there is a dichotomy between Speaker Independent (SI) and Speaker Dependent (SD) systems. Although SI systems are desirable, their performances are usually worse than that of SD systems. Thus, it is necessary to tune the recognition system to recognize new speakers. To address these issues, we attempt to modify the speaker-independent system using a small amount of data from the specific speaker to improve its performance. The main principle of speaker adaptation is to modify the recognition system to accommodate the acoustic variation of a new speaker. Figure 3 illustrates the overall view of adaptive speech recognition systems.

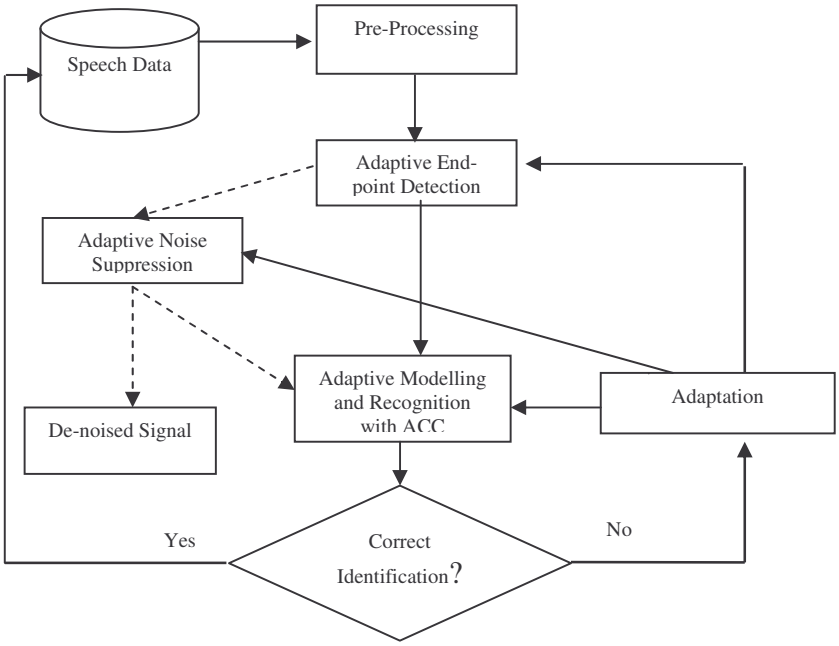

Figure 3: An overall view of the adaptive speech recognition system

Speech Pre-Processing

The first step in any speech recognition system is to prepare the captured speech into a suitable format so that they can be used to extract necessary features vectors. The raw speech signal often captured with noise and non-speech (silence) which are not always desirable. Endpoints detection is one of the problems in the pre-processing part of speech recognition systems. Endpoints detection refers to determination of the exact moment in which speech begins or ends.

We propose a neural network based method to accurately determine the beginning and end of each word. This method is described in Section 4.

Feature Extraction

The next step in speech recognition system is to extract necessary input features from the speech signals. Speech signal obtained from the pre-processing is divided into windows of 20 msec. duration. Each window is overlapped 50% with its previous window. Spectral analysis of the speech signal is performed for every window to calculate Mel-Frequency Cepstrum Coefficients (MFCC). Discrete Cosine Transformation (DCT) is applied on the MFCC of the whole word in the following manner. For an m frame segment, DCT transformation will result in a set of m DCT coefficients. This sequence is truncated to achieve a fixed-size input vector consisting of $20 \times d$, where d is the dimensionality of the feature space.

Speaker Adaptation

Speaker adaptation is needed when words within the existing vocabulary of ACC is not recognised correctly. Speaker adaptation is achieved through a fast and straightforward procedure. Once a given word pronounce by a new speaker is not recognised correctly, the new speaker can adapt the ACC by additionally training network on his/her speech. This is achieved through a supervised adaptation procedure.

New speakers required to pronounce the misrecognised word and adapt the recognition system by structurally evolving the ACC to accommodate the new speakers. Experimental results show that only few instances of data from new speakers are adequate for the ACC to learn the acoustic variation of the new

speakers. In addition unlike traditional neural networks, further training does not degrade the performance of the ACC on its previous knowledge.

Vocabulary Expansion

In any practical speech recognition systems, one of the desirable features is to enable adding new words into existing vocabulary of the system. The online expansion algorithm described in Section 2.3.1 allows new words to be added to the existing vocabulary of the system. The vocabulary expansion takes place in a supervise manner. Once a new word is presented to the ECoS, the user is required to provide a label for the new word. The expansion algorithm creates an additional output node and adapts the ACC to the new word.

The motivation behind designing this experiment was to examine the performance of ACC on isolated spoken word recognition. In addition, an on-line output expansion algorithm was applied to allow for vocabulary expansion of a small connectionist-based speech recognition system. The experiments were carried out in two distinct phases. In the first phase, an ACC is trained for digit recognition and speaker adaptation was applied on already trained ACC. In the second phase, the output space of the ACC was expanded to recognise a new set of words.

The speech data was recorded in a quiet room environment to obtain clean speech signals. 23 native New Zealand English speakers participated in the recording sessions using close-mouth microphone. The speech was sampled at 22.05 kHz and quantised to a 16 bit signed number. Each word was uttered five times with distinct pauses in between. Each of these words was then carefully manually segmented and labelled.

In the first phase, speech data was prepared for the English digits “zero” to “nine”. The speech data used in these experiments obtained from two groups of speakers; group A (8 male and 8 female) and group B (3 male and 4 female). The data from each group was divided into two sets; training set A, testing set A, training set B and testing set B.

Training set A with 480 examples was obtained from 3 utterances of each word in group A. The remaining 2 utterances were used in the testing set A for a total of 320 examples. Training set B with 140 examples was obtained from 2 utterances of each word in group B. Testing set B with 210 examples was obtained from the remaining 3 utterances of each word in group B.

Spectral analysis of the speech signal was performed according to feature extraction procedure described in Section 3 to obtain input feature vectors.

Training ACC

An ACC was initialised with 100 nodes in the input layer and 10 nodes in the output layer (one for each word). In phase one of the experiments the ACC was trained on the training set A. Aggregation was used to control the size of the evolving layer during the learning process.

Table 1. Performance of ACC on NZ English Digits on “Testing Set A" and “Testing Set B".

	Testing Set A		Testing Set B	
	Positive	Negative	Positive	Negative
Words	Accuracy	Accuracy	Accuracy	Accuracy
Zero	96.88	100.00	85.71	96.83
One	100.00	98.26	95.24	96.83
Two	100.00	99.65	100.00	98.41
Three	100.00	100.00	95.24	100.00
Four	100.00	99.65	100.00	98.41
Five	93.75	100.00	95.24	100.00
Six	100.00	100.00	95.24	100.00
Seven	96.88	99.65	61.90	99.47
Eight	100.00	100.00	100.00	100.00
Nine	87.50	100.00	76.19	99.47
Overall	97.50	99.72	90.48	98.94

There were 196 nodes created during the training period. The trained ACC was then tested on training set A and both testing sets A and B. The ACC performance on training set A was 100%. Table 1 shows the performance of the ACC on testing sets A and B.

To test the adaptation vs forgetting capabilities of ACC, the testing set B (ie: new speakers), the trained ACC was additionally trained on training set B. 43 additional nodes were added to the new ACC to accommodate the variations in the new speakers. The performance of the adapted ACC on its training set B was 100%. The adapted ACC was then tested on the original training (training set A) and testing sets A and B. Performance of the adapted ACC on training set A was remained unchanged. Table 2 illustrates the performance of the ACC on testing sets A and B after adaptation on new speakers. ACC retained its recognition ability on old data while achieving excellent adaptation on new data.

Table 2. Performance of ACC on NZ English Digits on “Testing Set A" and “Testing Set B" After Adaptation on “Testing Set B".

	Testing Set A		Testing Set B	
Words	Positive Accuracy	Negative Accuracy	Positive Accuracy	Negative Accuracy
Zero	96.88	100.00	100.00	100.00
One	100.00	99.65	100.00	100.00
Two	100.00	100.00	100.00	100.00
Three	100.00	100.00	100.00	100.00
Four	100.00	99.65	100.00	100.00
Five	93.75	100.00	100.00	100.00
Six	100.00	100.00	100.00	100.00
Seven	100.00	99.65	100.00	100.00
Eight	100.00	100.00	100.00	100.00
Nine	100.00	100.00	100.00	100.00
Overall	99.06	99.90	100.00	100.00

The second phase of the experiment was designed to expand the current ACC to recognise seven new words. The same procedure was applied to prepare training and testing sets. For the 7 additional words, there were a total of 336 examples in training set A, 224 examples in testing set A and 147 examples in testing set B. As described in Section 4 the structure of the existing ACC was expanded and then ACC with it’s new structure was further trained on the training se A of the additional words. 130 new nodes were added to the evolving layer of the expanded ACC during the training process. Table 3 illustrates the performance of the new ACC on the original and additional testing set A and testing set B.

The results show that ACC is capable of learning a training set. After each training cycle of the original network, the network performed with 100% accuracy over the training set. This prototype learning (“memorisation”) is despite the fact that there are much fewer neurons in the evolving layer (prototypes) than there are examples in the training sets.

Despite the high level of memorisation shown by the network, a high level of generalisation was also demonstrated. Only one spoken digit word from testing set A was recognised with an accuracy of less than 90%, while only three digits from testing set B, zero, seven and nine, were recognised with a positive accuracy of less than 90%. For examples from all (both known and unknown) speakers (testing sets A and B), the true negative accuracy of the network is consistently over 90%. Thus, ACC is able to both memorise training data and generalise to new examples and new speakers to a very high degree.

Further training on training set B greatly improved the performance of the network over the testing set B, with positive and negative accuracies of 100% being recorded for each word. The network also learned the training set B perfectly well, with 100% recognition of all words. Despite this high level of adaptation, the accuracy over the testing set A did not decline. None of the positive accuracies were disturbed, while the negative accuracies over the words ‘one’ and ‘two’ increased slightly. This shows that not only is an ACC highly adaptable, but it is also highly resistant to forgetting.

Table 3. Performance of ACC on Both 10 Original and 7 Additional Words after Expansion of ACC on "Testing Set A" and "Testing Set B".

	Testing Set A		Testing Set B	
Words	Positive Accuracy	Negative Accuracy	Positive Accuracy	Negative Accuracy
Zero	96.88	100.00	100.00	100.00
One	100.00	99.80	100.00	99.70
Two	100.00	100.00	100.00	100.00
Three	100.00	100.00	100.00	100.00
Four	100.00	99.80	100.00	99.70
Five	93.75	99.61	100.00	99.40
Six	100.00	100.00	100.00	100.00
Seven	100.00	99.80	100.00	100.00
Eight	99.63	100.00	100.00	99.11
Nine	100.00	100.00	100.00	100.00
Plus	100.00	100.00	100.00	99.70
Minus	100.00	100.00	85.71	100.00
Times	100.00	100.00	100.00	100.00
Divides	93.75	100.00	95.24	100.00
Point	100.00	100.00	100.00	100.00
Back	100.00	99.41	85.71	99.70
Equals	100.00	100.00	90.48	100.00
Overall	98.53	99.91	97.48	99.84

Addition of new classes (output nodes) to the network does not cause any disturbance to the recognition rate of the existing classes. After adding new classes, the worst recognition rate for any of the existing words was 93.75%, for the word 'five', which is identical to the recognition rate before addition of the new classes. Of the added words, only the word 'divides' had less than perfect positive accuracy, with a performance of 93.75%. A single word, 'back', had a negative accuracy of less than 100%, at 99.41%. Thus, the addition of new outputs does not cause any forgetting of existing classes, nor does it disturb the memorisation ability of the network: the classes that were added are recognised as well as the existing classes.

Generalisation accuracy over the testing set B is very high, even though adaptation to this set was not performed for the additional words. The lowest positive accuracy was 85.71%, for the words 'minus' and 'back'. All other positive accuracies were over 90%, while all negative accuracies were over 99%. This shows that the ability of the network to generalise is the same for the added classes as it is for the previously existing classes.

Aggregation during training had a limited affect. Few nodes were aggregated in relation to the total size of the network. This is because of the very high aggregation threshold used during training, which caused very few nodes to be aggregated. Although the mild aggregation has improved the generalisation capability of the network, it highlights a problem with aggregation during training: as the distribution of the data for each class is different, it is dangerous to apply the same aggregation threshold to nodes that represent different classes. Ideally, the aggregation threshold should be set for each class. Thus, it would be possible to set the threshold for optimal performance for each class. The manner in which these thresholds are optimised is a matter of further research.

4. WORD SELECTION FROM A CONTINUOUS SIGANAL

Interactive speech recognition systems are only useful if they can run with live (on-line) inputs. The problem with on-line systems, as opposed to off-line systems, is that the exact start and end of the utterance is unknown.

One of the problems in speech processing is to accurately determine the beginning and end of speech signals. This is known as endpoints detection problem. The simplest method of making a speech/non-speech decision is to use a combination of zero-crossings and RMS energy. Together these two features provide a reasonable separation of speech and silence since low energy speech (fricatives) tends to have high zero-crossing rates and low zero-crossing speech (vowels) tend to have high energy. However, this method is not reliable or accurate enough, especially when the signal to background-noise ratio is high. In the following section a method based on neural network is proposed to determine speech and non-speech segments of speech signals.

Adaptive speech/non-speech detection

Speech is a non-stationary (time-varying) signal; silence (background noise) is also typically non-stationary. Background noise may consist of mechanical noises such as fans or conversations, movements, and door slams that are difficult to characterize. The novel technique described in this section involves classifying these two non-stationary signals. None of the endpoints detections methods are designed to cope with the non-stationary nature of speech and non-speech signals referring to method proposed in (Platt, 1991). Since the speech and silence patterns are highly variable, it is desirable to use an adaptive solution to classify these two signals. The task is to detect the silence/background-noise between words (inter-word) or within a word (intra-word). Figure 4 illustrates an implementation of a neural network model for speech/non-speech detection.

Feature computation

Speech signals uttered by different speakers were captured using various microphones in different environments. Training and testing sets were prepared from the manually segmented and labelled of the speech signals. The acoustic features were calculated every 10 ms, with a frame length of 20 ms. For each frame, 12 Mel-Frequency Cepstrum Coefficients (MFCC), log energy and their corresponding first and second order derivatives were computed. In addition, zero-crossing rate for each frame was also included as one of the elements in the input feature vector.

Modelling and recognition with the ACC based *model*

The Adaptive Connectionist Speech and Image Classifier (ACC) as described in Section 2.3 is an implementation of the ECoS paradigm. The task is to train an ACC to classify the input vectors into two classes of speech or non-speech (silence). The outputs of ACC correspond to the given input frame.

A rule based system is applied to determine the sequence of speech and non-speech frames. The following are few of these design making rules:

IF c or more consecutive sequence of frames is *speech*
- these frames are *speech*. (c is determined experimentally)

IF c or more consecutive sequence of frames is *non-speech*
- these frames are *non-speech.*

IF frame n is signal instance of speech frame and preceding c or more frames are silence frame

ELSE IF subsequent c or more frames are also *non-speech.*
* frame n is *non-speech.*

ELSE IF there is less than c subsequent frame to the end of sequence
* frame n is *non-speech.*

ELSE IF there is less than c subsequent frame of *non-speech.* **AND** c or more subsequent frames after that are *speech*
* frame n is *speech*

Figure 4 illustrates a model for speech/non-speech detection. The output of neural network is a sequence of binary numbers, 0 representing non-speech and 1 representing speech for each frame of speech.

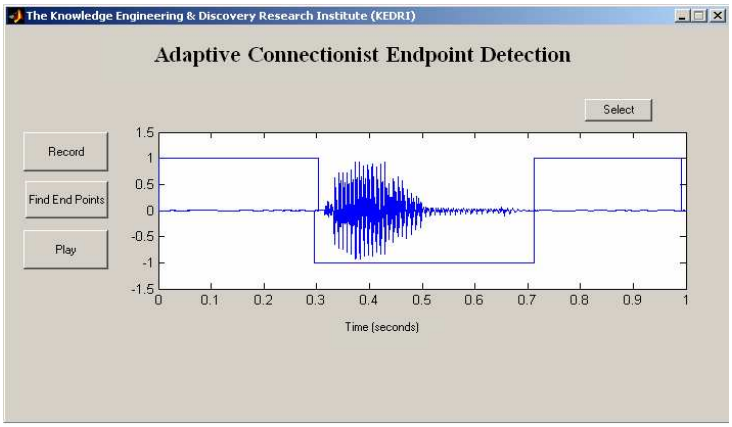

Figure 4: An implementation of the ACC model for speech/non-speech detection.

Experimental Results

Experiments conducted to examine the performance of the proposed speech/non-speech detection system. Training and testing data were prepared from speech and non-speech segments of speech signals. A total of 42 input features were computed for every input frame. The task is to classify these input frames into speech or non-speech classes.An ACC is trained on 1900 training samples and 293 nodes were created during the learning process. Table 4 shows the results of ACC on the training and testing samples.

Table 4 : Performance of ACC on speech and non-speech training and testing data

	Training data		Testing data	
	Number of samples	Correctly identified	Number of samples	Correctly identified
Speech	1000	98.30%	368	90.49 %
Non-Speech	900	99.78%	396	97.98 %

ACC is suitable for the task of classifying speech and silence because, it can be adapted to new background noises through a supervised adaptation. Figure 4 illustrates the performance of connectionist based end points detection on a spoken word. It also allows on-line adaptation of both silence and speech segments of a speech signal. This is done by user selecting the silence or speech segment and retraining the system to learn these as new data. This an essential feature of the adaptive end points detection system since the nature of background noise varies depending on the recording environment.

5. A METHOD FOR ADAPTIVE MODEL INTEGRATION

The aim of the integration module is to combine the speaker identification and image recognition modules to recognise identity of a person. This study attempts to increase the robustness of person identification system by applying integration module presented in this section. Figure 1 illustrates the overall proposed integration module for speech and Image.

This module receives inputs from image and speaker recognition modules. The pre-processing of the contribution from the image and speaker recognition modules are performed by assigning appropriate weights to their recognition rates. The weights have values that are proportional to the probability of each module producing a correct classification results. These weights are determined experimentally based on the performances of each module over some testing data.

The method of assigning weights for the contribution from image and speaker speech recognition modules is computed in the following manner. Let assume that the image and speaker recognition modules are trained to be recognised n persons. The performance of each module is tested on a number of new samples. For instance, if 85 percent of images of output category j are correctly recognized, then the

reliability of this neural network for output j is 0.85. Thus, 0.85 is assigned as the weight for the contribution from image recognition module for output j and denoted as WI_j. The similar process is performed to assign weights for the contribution from speaker recognition module and denoted as WS_j. The final result for the integration module is made by applying a statistical specialization with the contribution from image and speaker recognition modules. The contribution from image (or speaker) recognition module is an array of output activations for each class.

The probability of output $j, \quad j = 1:n$ to be the correctly identified person is calculated according to the Equation.

$$P_j = (W_i \times A_{ij}) + (W_s \times A_{sj}) \tag{4}$$

Where: A_{ij} is the activation of image recognition module for output j , and A_{sj} is the activation of speaker recognition module for output j

Once the probability for each output category is calculated, the output class with the highest probability is the final recognition result (Kasabov et. al., 2000). For example, suppose there are three classes A, B and C. The contribution of image recognition module is [0.8, 0.6, 0.9] which means that the activation of class A is 0.8, activation of class B is 0.6 and class C is 0.9. The contribution of speaker recognition module is [0.7, 0.9, 0.5]. Suppose the weights for image module WI = [0.99, 0.97, 0.85] and weights for speaker modules WS = [0.85, 0.95, 0.99]. The statistical calculation for each class is:

Class A: $0.8 \times 0.99 + 0.7 \times 0.85 = 1.387$
Class B: $0.6 \times 0.97 + 0.9 \times 0.95 = 1.437$
Class C: $0.9 \times 0.85 + 0.5 \times 0.99 = 1.26$

The class with the highest activation value (class B) is the winner (the final result of classification).

6. PERSON IDENTIFICATION

The motivation of this work is to identify a person based on two separate sources of information. One module is designed to identify a person according to the characteristics of his/her voice and the other module recognises a person based on image of his/her face. In the following sections these two modules are described.

Image Recognition Module

Image recognition consists of locating and identifying known patterns in an image. Depending on the application, the patterns can be expected to appear within pre-defined zones in the field of view or anywhere. The object or pattern to identify is defined by a region including the edges of the objects along with a layer of background pixels, so that the recognition engine can learn variations of these edges and be able to differentiate anomalies from similarities.

The same object can be represented with different feature vectors. Several feature vectors can be combined to build a robust engine. Image recognition module uses evolving neural networks to recognise a given image.

Feature Extraction

Feature extraction is the process of extracting a defined area of an image, and transforming the information into a "feature" vector. There are a number of different feature extraction methods to choose from, all of which will generate different values for the vector. The generated vector can then be sent to the neural network engine for either learning or recognition (classification).

The available feature extraction options are: Pixel values, Gradient values, Standard histogram, Cumulative histogram, Gradient histogram, Vertical profile, Horizontal profile, Composite profile, Vertical gradient profile, Horizontal gradient profile, Composite gradient profile. The most frequently used one is

Composite profile. If the image of interest has a size of $m \times n$, denoted as $I_{m \times n}$. The composite profile vector V is an array of $m+n$ components, denoted as V_{m+n}. The m components represent the vertical profile (the average pixel value of each column), and calculated according to the Equation 5.

$$V_i = \frac{1}{n} \sum_{j=1}^{n} I_{i,j} \quad , i = 1 : m \tag{5}$$

The n components represent the horizontal profile (the average pixel value of each row), and calculated according to Equation 6.

$$V_i = \frac{1}{m} \sum_{j=1}^{m} I_{(i-m),j} \quad , i = m+1, m+2, ...m+n \tag{6}$$

Speaker Identification

Speech signals contain two types of information; individual and phonetic. They have mutual effects and not easy to separate. This represents one of the main problems the area of speaker and speech recognition systems. Speaker recognition systems can generally fall into two categories; *text-dependent* and *text independent*. Text-dependent means that the text used in training the system is the same as that used in testing the system. However in text-independent the text used in testing the system is limited to the text used is not limited to the text used to train the system. The speaker identification system proposed here is text dependent. The pre-processing and modelling steps are identical the one used for the adaptive speech recognition systems (see Section 3) and Figure 3 illustrates the overall proposed module. However, the task of neural network is to identify the speaker of the text based on input speech signal.

A neural network is trained on a key word or phrase from various speakers. The number of output nodes in the output layer of neural network corresponds to the number of known speakers. The network can be adapted to identify a new speaker using expansion algorithm and further adapted to the new speaker. In addition, output deletion algorithm can be used to remove an output (speaker) from the output layer of the neural network.

Figure 5 illustrates a graphical user interface developed for connectionist person identification system. The GUI is developed in MATLAB and allows online person identification, registration and adaptation. The integration of speech and image modules are based on the method described in Section 5.

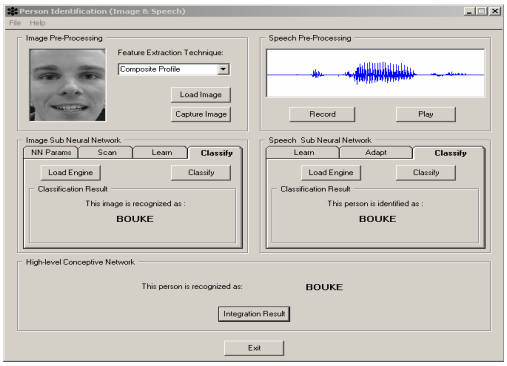

Figure 5. An implementation of adaptive integrated model for person identification

The image recognition module allows various evolving connectionist systems to be used. These include; zero instruction set computing (ZISC), evolving classifier Function (ECF) and adaptive connectionist classifier (ACC). They all allow structural adaptation by modifying their internal structure during the learning process. Our previous work on integrating speech and face image inputs for person identification (Zhang et. al, 2004) or verification task (Ghobakhlou et. al, 2004) shows that performance of multi-modal systems are superior to single modality.

7. CONCLUSIONS AND FUTURE RESEARCH

This paper describes several applications of adaptive connectionist systems. The notion of evolving while learning is realised in the implementation of adaptive connectionist classifier. Experiments were carried out with ACC for isolated word recognitions to explore the performance of the ACCs local learning algorithm. A novel idea for expanding the output space in SECoS was presented and successfully implemented. It was clearly demonstrated that SECoS is capable of learning data from new speakers and adding new class outputs.

It was also shown that ACC maintained its previously learned knowledge after adaptation on new speakers and after learning new classes. Further investigation into optimisation of the aggregation and its relevance to the output expansion algorithm and the overall generalisation of the system is currently being conducted.

8. REFERENCES

Arbib, M. (ed) (2003), *Handbook of brain theory and neural networks*, MIT Press.

Amari, S. and Kasabov (eds) (1998*), Brain like computing and intelligent systems*, Springer Verlag.

Blanzieri, E.and Katenkam, P. (1996), Learning radial basis function networks on-line, Proc. of Intern. Conf. On Machine Learning, 37-45.

Bottu and Vapnik (1992), "Local learning computation", *Neural Computation*, **4**, 888-900.

Edelman, G. (1992), *Neuronal Darwinism: The theory of neuronal group selection*, Basic Books.

Freeman, J., Saad, D. (1997), "On-line learning in radial basis function networks", *Neural Computation* , . **9**(7).

Fritzke, B. (1995), "A growing neural gas network learns topologies", *Advances in Neural Information Processing Systems*, 7.

Gaussier, T., and S. Zrehen (1994), "A topological neural map for on-line learning: Emergence of obstacle avoidance in a mobile robot", *From Animals to Animats*, 3, 282-290.

Ghobakhlou, A., Zhang, D. and Kasabov, N. (2004), An Evolving Neural Network Model for Person Verification Combining Speech and Image, Proc. of ICONIP, Calcutta, India, in print.

Heskes, T.M. and Kappen, B. (1993), "On-line learning processes in artificial neural networks", *Math. foundations of neural networks*, Elsevier, Amsterdam, 199-233.

Kasabov, N. (2002), *Evolving connectionist systems: Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*, Springer Verlag.

Kasabov, N., Postma, E. and van den Herik, J. (2000), AVIS: a connectionist-based framework for integrated auditory and visual information processing, *Information Sciences*, vol. 123, 127-148.

Kasabov, N., (2000), Adaptive learning system and method, PCT patent WO 01/78003, publication date 20.04.2000.

Kasabov, N., and Ghobakhlou, A. (2003), Adaptive Speech and Image Learning System and Method, Patent ref. No. 529081, 22 Oct.2003, New Zealand.

Moody, J. and Darken, C. (1989), " Fast learning in networks of locally-tuned processing units," *Neural Computation*, **1**(2), 281-294 .

Platt, J. (1991), "A resource allocating network for function interpolation," *Neural Computation*, **3**, 213-225.

Rosipal,R., Koska, M. and Farkas, I. (1997), "Prediction of chaotic time-series with a resource-allocating RBF network, *Neural Processing Letters*, **10**(26).

Saad, D. (eds.)(1999), *On-line learning in neural networks*, Cambridge University Press.

Schaal, S. and Atkeson, C. (1998), "Constructive incremental learning from only local information, Neural Computation," **10**, 2047-2084 .

ZISC Manual, (2002), Silicon Recognition, www.silirec.com.

Zhang, D., Ghobakhlou, A. and Kasabov, N. (2004) "An Adaptive Model of Person Identification Combining Speech and Image Information", International Conference on Control, Automation, Robotics and Vision (ICARCV) , Kunming, China, in print.

Received: Mar 10th 2004

Accepted in final format: Nov 20th 2004

About the authors:

Akbar Ghobakhlou is a research fellow and project leader of the Center for Signal Processing, Speech and Image in the Knowledge Engineering Discovery and Research Institute at the Auckland University of Technology. He obtained his BSc with Honours in information science in 1997from the University of Otago. He is about to submit his Ph.D. also with the University of Otago. His current research interest include speech recognition, signal processing, image recognition and neural networks. He can be reached by email at akbar@aut.ac.nz

Professor Nikola Kasabov is the Foundation Director of KEDRI, and a Chair of Knowledge Engineering at the School of Computer and Information Sciences at AUT, Fellow of the Royal Society of New Zealand, Fellow of the New Zealand Computer Society and a Senior Member of IEEE. He holds a MSc and PhD from the Technical University of Sofia. His main research interests are in the areas of: intelligent information systems, soft computing, neuro-computing, bioinformatics, brain study, speech and image processing, data mining and knowledge discovery. He can be reached by email at nkasabov@aut.ac.nz