SALIENCE-AFFECTED NEURAL NETWORKS

Written by Leendert Amani Remmelzwaal
Supervised by Jonathan Tapson, and George Ellis

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN ENGINEERING

UNIVERSITY OF CAPE TOWN
November 2009

Abstract

The cortex biases neuronal responses to affect both learning and memory, through the use of *diffuse projections* from the limbic system to the cortex.

Standard artificial neural networks (ANNs) do not usually model this non-local flow of information represented by the ascending systems, which are a significant feature of the structure of the brain. This structure of non-local connections allows modeling of global effects such as salience, at the same time as the local network processes task-specific or local information. Although, non-local connections allow for associational learning with multiple-trial in standard neural networks, they seldom provide the capacity for one-time learning.

In this research, the *salience* of an entity refers to its state or quality of standing out, or receiving increased attention, relative to *neighboring* entities. By *neighbouring* entities we refer to both spatial (i.e. similar visual objects) and temporal (i.e. related concepts).

In this research we model the effect of non-local connections using an ANN, creating a *salience-affected neural network* (SANN). We adapt an ANN to embody the capacity to respond to an *input salience signal* and to produce a *reverse salience signal* during testing. The *input salience signal* applied during training to each node has the effect of varying the node's thresholds, depending on the activation level of the node. Each node produces a *nodal reverse salience signal* during testing (a measure of the threshold bias for the individual node). The *reverse salience signal* is defined as the summation of the *nodal reverse salience signals* observed at each node.

This research demonstrates that input combinations similar to the inputs in the training data sets will produce similar *reverse salience signals* during testing.

Furthermore, this research demonstrates how an ANN can be effectively trained using a single training iteration; a feature which will have significant impact in the field of neural networks.

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously, in its entirety or in part, submitted it at any university for a degree.

I hereby declare that my research was carried out in such a way that:

- there was no apparent legal objection to the nature or the method of research
- the research did not compromise staff or students or the other responsibilities of the University
- the stated objective were achieved, and the findings have a high degree of validity
- limitations and alternative interpretations were considered
- the findings were subject to peer review and is publicly available

Signed by candidate

Acknowledgements

I would like to express my gratitude to Jonathan Tapson for his patience and supervision and to George Ellis for his inspiration and guidance during this thesis, and to my friends and family who provided me with continued support and motivation.

Table of contents

Abstract i
Declaration ii
Acknowledgements iii
Table of contents iv
List of figures vii
Glossary x
Glossary x
Acronyms x
1. Introduction 1
   1.1. Models of the human cortex 1
   1.2 Neurons 3
   1.3. Local connections 4
   1.4. Diffuse projections, ascending and value systems 4
   1.5. Salience 5
   1.6. Single trial learning 5
   1.7. GasNets 6
2. Thesis proposal 7
3. Research implications 8
   3.1. Computational implications 8
   3.2. Biological implications 8
   3.3. Behavioural implications 8
4. Research overview 10
5. Artificial neural network architecture 11
   5.1. Introduction 11
   5.2. ANN layers 11
   5.3. ANN dynamic links 12
   5.4. ANN sigmoidal functions 12
   5.5. ANN training 13
6. Salience signal 14
   6.1. Embedded salience signal 14
   6.2. General threshold variation 15
   6.3. Threshold adjustment definition 17
   6.4. Reverse salience signal 18

7. Face recognition application 19
7.1. Introduction 19
7.2. Principal Components 19
7.3. Vector Quantization 19
7.4. Non-Negative Matrix Factorization 20
7.5. Algorithm comparison 20
7.6. Algorithm selection 21
8. Computing with NMF 22
8.1. Image representation 22
8.2. Mixing and encoding matrices 22
8.3. Cost functions 22
8.4. Minimization of the Cost Function 22
8.5. Reconstruction algorithm 23
9. Software implementation 24
9.1. Data collection 24
9.2. NMF source code 24
9.3. ANN source code 24
9.4. ANN input, hidden and output layers 24
9.5. ANN training and testing 25
9.6. Salience implementation 25
10. General analysis and optimization 26
10.1. Introduction 26
10.2. Residual reverse salience signal value 26
10.3. Hidden layer units 27
10.4. Salience signal magnitude 29
11. Results 31
11.1. Introduction 31
11.2. Multiple-trial learning technique 31
11.3. Single-trial learning technique 33
12. SANN Application 37
12.1. Introduction 37
12.2. Training stage 38
12.3. Testing stage 38
12.4. Reverse salience signal test 38
12.5. Results 41

13. Conclusions 42
13.1. SANN 42
13.2. Multiple-trial training 42
13.3. Single-trial training 43
13.4. Theory of diffuse projections 43
13.5. Three principles of salience 43
13.6. Face recognition application 43
14. Recommendations and future work 45
References 46
Bibliography 49
Appendixes 50
Appendix A: SANN application operation manual 50

List of figures

Figure 1: Three main topological arrangements of fundamental neuroanatomy as proposed by Edelman[8]. (A) The top diagram shows the thalamocortical system. (B) The middle diagram depicts long, polysynaptic loops that are arranged in parallel and that leave the cortex, enter the so-called cortical appendages, and return to the cortex. (C) The bottom diagram indicates one of the diffusely projecting value systems. 2

Figure 2: Drawing of a neuron and synapse[4] 3

Figure 3: A visualization of a fully-connected multilayer perceptron artificial neural network with a 2-node input layer, a 2-node hidden layer and a single-node output layer. Each inter-nodal connection is weighted, and can be trained. Each node is responsible for summing the weighted signals it receives. The response of each unit, in the form of an output signal, is a function, usually sigmoidal, of the weighted sum of the signals entering the unit. 12

Figure 4: An SANN with a 2-node input layer (n_i), a 2-node hidden layer (h_{1i}), and a single output node (o_i). Each node receives an *input salience signal* (S), which has the effect of varying the threshold value of the sigmoidal function at each individual node, during training, depending on the node's activation level. From each node, *nodal reverse salience signals* are summed, to create a *reverse salience signal* (S')for the SANN. 14

Figure 5: Varying thresholds of an approximate *tanh-based* sigmoidal function. It can be seen that the *input salience signal* applied to each node has the effect of pushing active node's thresholds closer to their activation signal (for positive salience), or moving them further away from this signal (for negative salience). 16

Figure 6: Direction of threshold adjustment as a function of salience (S) and instantaneous *node activation level* (x_i)...17

Figure 7: NMF, PCA and VQ learn to represent a face as a linear combination of basis images, but with qualitatively different results[31]. This figure demonstrates how each of the mentioned feature-extraction techniques creates a unique feature-matrix and weighting-matrix. Every trained image could be approximately reconstructed as a weighted sum of global features (Equation 9). Each technique results in an approximation, as feature based-extraction is an optimization technique.............. 20

Figure 8: Correlation between SANN output signal and residual reverse salience signal after being trained and tested, without salience, with 100 images from the CBCL database.. 27

Figure 9: Iterations required to reduce the NN error to <= 10% of the initial error, for various hidden layer sizes... 28

Figure 10: NN Error as a function of training iterations for various magnitudes of *input salience signal* values 29

Figure 11: Relative *reverse salience signals* for various images of faces (a sample of 20 input combinations are shown). For the training data set used, certain images were similar. For example, images 2, 3, 9, 10 and 11 were each unique images, but were all photos of the same person, and therefore very similar. Images 9, 10 and 11 were trained with an *input salience signal* of value 1, while the remaining images were trained with an *input salience signal* of value 0. Noticeably, images 2 and 3 returned a relatively high *reverse salience signal* value, despite being trained with an *input salience signal* of value 0. 32

Figure 12: Relative *reverse salience signals* for various images of faces (a sample of 20 input combinations are shown). Figure 11 explains the multiple-trial training method. For the single-trial training methods, an amplification factor was introduced. The amplification factor corresponds to the magnitude of the salience (or significance) attached a certain input combination. Amplification factors in the integer range [1-6] were tested, to observe the effects of varying magnitude of the *input salience signal*, and the *reverse salience signals* were recorded for various faces. Results attained from single-trial training with an amplification factors of 2 and 5 respectively, were plotted alongside multiple-trial training results, as seen above. A

single-trial training method produced a similar *reverse salience signal* profile to the multiple-trial training method, although of a noticeably smaller magnitude............ 35

Figure 13: Correlation between single-trial and multiple trial *reverse salience signal* values, for varying *input salience signal* amplification factors during single-trial training. A *correlation* only demonstrated a similarity in data profile, and not in magnitude. An SANN was trained using a single-trial training method, and with a dataset of 200 unique images of faces. First the SANN was trained without the influence of *input salience signals*. Thereafter a single training iteration was executed, where an *input salience signal* of value 1 attached only to images 9, 10 and 11. Amplification factors in the integer range [1-6] were tested. Correlation coefficients were calculated between the single-trial and the multiple-trial training, as a function of *input salience signal* amplification factors. A high correlation coefficient would indicate a similar profile, which we desire. 36

Figure 14: Reverse salience signal values from the SANN application for when a face was on the LHS, Center and RHS of the frame. Only images with a face in the center were trained with a positive salience, and as a result the *reverse salience signal* received during testing corresponded to the training.. 40

Figure 15: Pre-training stage of the SANN application 50

Figure 16: Post-training stage of the SANN application.. 51

Figure 17: Testing stage of the SANN application 52

Glossary

For the purpose of this research, the following terms have been defined:

- The *salience* of an entity refers to its state or quality of standing out, or receiving increased attention, relative to *neighbouring* entities. By *neighbouring* entities we refer to both spatial (i.e. similar visual objects) and temporal (i.e. related concepts).
- *The input salience signal* is the additional input signal affecting each node during training
- *The nodal reverse salience signal* is the additional output signal produced by each node during testing.
- *The reverse salience signal* is the additional output signal produced by the neural network (the sum of all the *nodal reverse salience signals*)
- The *salience influence* is defined as the inverse of the rate at which the thresholds were influenced by the salience signals.

Acronyms

For the purpose of this research, the following acronyms have been defined.

- ANN – artificial neural network
- DLA – dynamic link architecture
- MLP – multilayer perceptron
- NMF – non-negative matrix factorization
- NN – neural network
- PCA - principal component analysis
- SANN – salience affected neural network
- VQ - vector quantization

1. Introduction

1.1. Models of the human cortex

Researchers have been constantly improving on models of the human cortex. For example, associative memory (AM) systems have arisen from work on models of human memory systems[1]. AM systems are memory systems that allow one data item to be associated to another. Access to one data element allows access, by association, to the other data element.

Many conventional neural network architectures fall into the category of AM systems. There are two major classes of neural-network-based AM systems, namely *feed-forward neural networks* and *recurrent neural networks.* We have chosen to specifically focus on the standard *feed-forward* artificial neural network (ANN) in this research, as an abstract model of the neural structure of the human cortex. In this research, we will be referring to this ANN model as a *standard ANN*. We accept that an ANN is not an accurate approximation of the human cortex, but rather a *useful* abstraction for use in this research. Furthermore, we acknowledge that many researchers in computational neuroscience use more detailed biological models[2]. This research discussed here can be effectively and easily applied to an ANN, and can be easily adapted for use with alternative neural network models.

Apart from motor and sensory systems, Edelman describes the human brain has at least three classes of connections found within the cortex[3]. These classes consist of localized inter-cortical connections (**Figure 1A**), longer range inter-cortical connections (**Figure 1B**), and diffuse projections from the limbic system to the cortex (**Figure 1C**). By *inter-cortical* connections, we refer to connections between neurons within regions of the cortex.

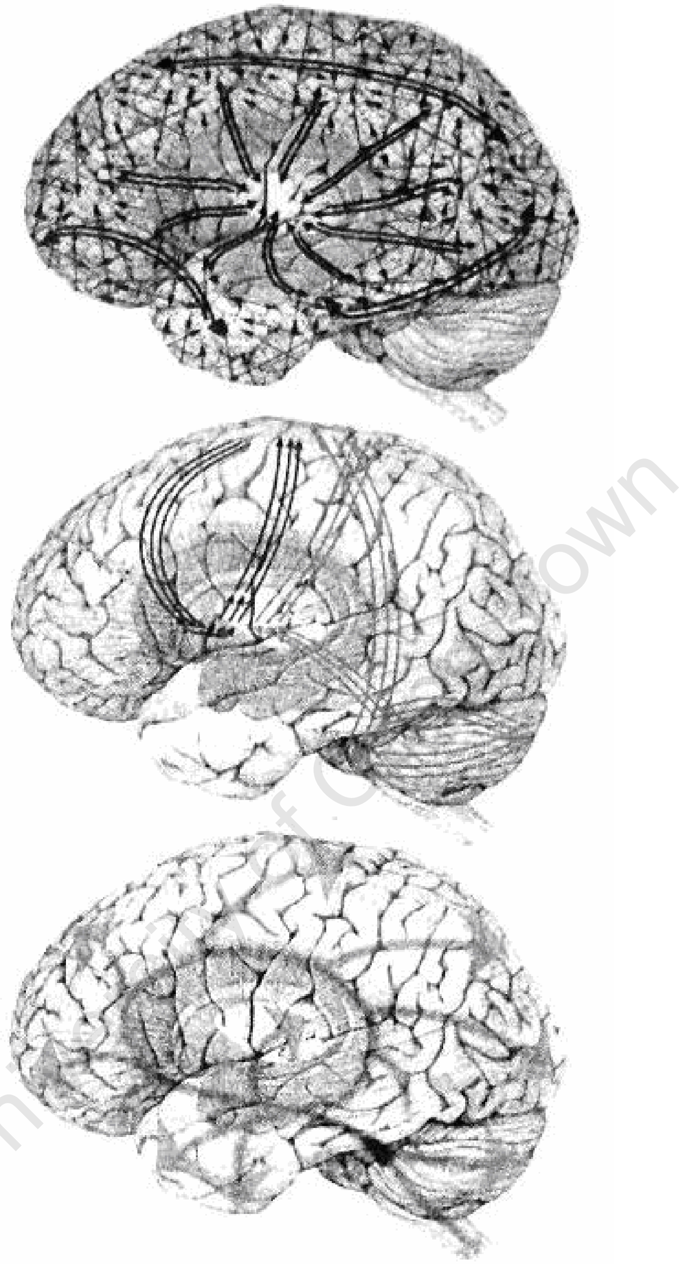

Figure 1: Three main topological arrangements of fundamental neuroanatomy as proposed by Edelman[8]. (A) The top diagram shows the thalamocortical system. (B) The middle diagram depicts long, polysynaptic loops that are arranged in parallel and that leave the cortex, enter the so-called cortical appendages, and return to the cortex. (C) The bottom diagram indicates one of the diffusely projecting value systems.

1.2 Neurons

Neurons are specialized cells as shown in **Figure 2**. Each neuron is characterized by having many dendrites and a single axon. Electrical information is transmitted from the nucleus, down the axon as shown by the arrows. Axons are protected with a *myelin* insulation which is required for rapid electrical signaling.

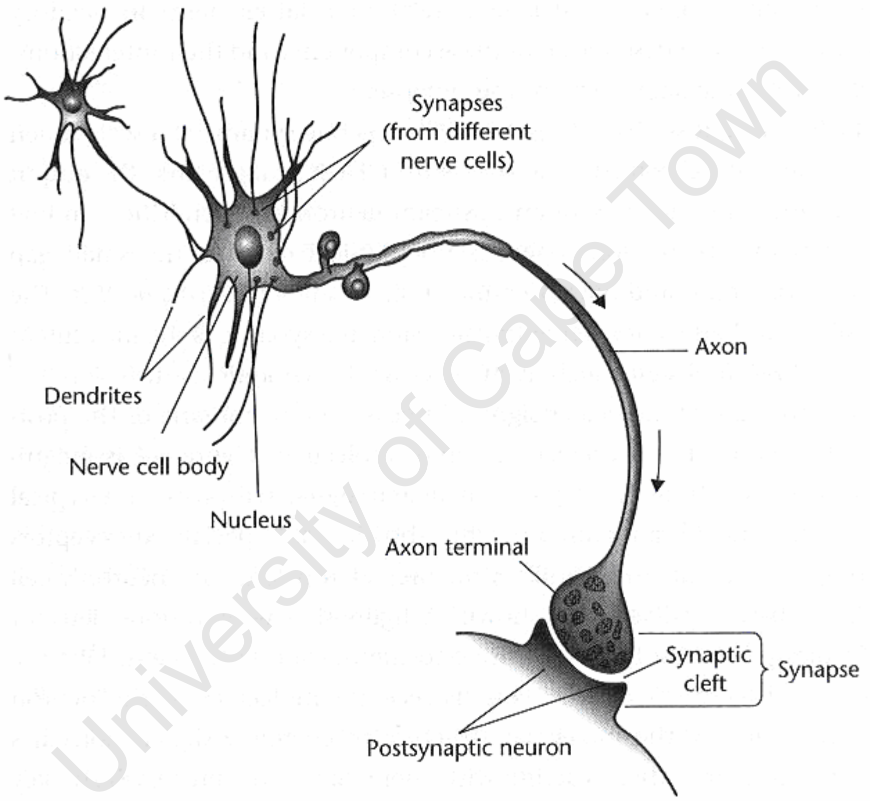

Figure 2: Drawing of a neuron and synapse[4]

In a neuron, positive and negative ions are responsible for the signaling[4]. If the positive ions in the neuron sufficiently outweigh the negative ions, the neuron emits a signal, which travels from the nucleus down the axon. In transmitting a signal, a neuron rapidly changes its internal charge from negative to positive – a process referred to as *depolarization*[5].

The connection between two neurons occurs at the *synapses*. These synaptic connections are crucial in the processes of learning and memory[4]. In *synaptic connections* the electric inter-neural signal is converted to a chemical signal that crosses the synaptic gap and then gets converted back to an electrical signal by the postsynaptic neuron. One particular variation of the biological *synaptic connections* is a model known as *gap-junction connections,* which will be discussed in later sections.

1.3. Local connections

The first class consists of localized inter-cortical connections forming layered neural networks (**Figure 1A**). As mentioned, *inter-cortical* connections refer to connections between neurons within regions of the cortex. Standard ANNs have the ability to model the way signals flow *locally* in the columns in the cortex through *local connections* in those columns[10]. This ability to manipulate the strength of connections between neurons effectively allows for complex processes such as pattern recognition[6] and naming[7] to be achieved with ANN models. However, this research continues to investigate and attempts to model the more complex concept of *diffuse projections*.

1.4. Diffuse projections, ascending and value systems

Another class consists of diffuse projections from the limbic system to the cortex and are known as *monoamine systems*[7], or *ascending systems*[5]. From their nuclei of origin they send axons up (towards the center of the cortex) and down (away from the center of the cortex) the nervous system in a diffused spreading pattern (**Figure 1C**).

The effect of the monoamine systems projecting profusely is that each associated neurotransmitter (for example norepinephrine and dopamine) affects large populations of neurons, allowing non-local interactions to occur in the brain. The release of the neurotransmitter affects the probability that neurons in the neighbourhood of value-system axons will fire after receiving glutamatergic input, thus they are an important mechanism effecting neural plasticity. These systems bias neuronal responses affecting both learning and memory by guiding neuronal group selection, and for this reason that they are sometimes termed *value systems*[8,9].

Standard ANNs do not model this non-local flow of information represented by the ascending systems, which are a significant feature of the structure of the brain[11]. The following section will discuss the concept and operational definition of *salience*, adopted in this research.

1.5. Salience

In this research, the *salience* of an entity refers to its state or quality of standing out, or receiving increased attention, relative to entities that are either spatially (i.e. similar visual objects) and temporally (i.e. related concepts) related. For example, a salient memory would be one that significantly stands out among others.

Beebe and Lachmann[11] defined the three principles of salience that describe the interaction structures in the first year of life. These are the principles of *ongoing regulations*, *disruption and repair*, and *heightened affective moments*. In this research, we focus on *ongoing regulations*, and *heightened affective moments.* Ongoing regulation describes the characteristic pattern of repeated interactions, such as a child interacting with their mother or father, resulting in increased salience towards certain associations and memories[11]. A heightened affective moment, on the other hand, describes a dramatic moments standing out among other events[11], often leading to one-time learning, and an instantaneous attachment of salience to certain associations. Standard ANNs allow associational learning with multiple-trial (i.e. *ongoing regulations* without salience)*,* but do not provide the capacity for *one-time learning*[24], which has been the subject of discussion in the psychology literature for over 50 years.

1.6. Single trial learning

Heightened affective moments describes one dramatic moment standing out in time, among other events[11], for example, experiencing a sudden fright when opening the fridge door. Neural network models have attempted to model both multiple-trial and single-trial learning. However, standard ANNs simply don't provide the capacity for single-trial learning (i.e. *heightened affective moments)*, which is an established psychological phenomenon[12], and as a result, single-trail learning has been the subject of discussion in the psychology literature for over 50 years.

While the details of how single-trial training works in the cortex are in dispute, the fact that single-trial training sometimes occurs is well established[13,14]. When it occurs, single-trial learning is often associated with the neurotransmitter dopamine[15,16]. Dopamine is one of the neurotransmitters that is released throughout the cortex by the ascending systems, and is associated both with the neural coding of basic rewards and in reinforcement signals defining an agent's goals[17,18]; it is known to play an important role in brain functioning.

Standard ANNs do not usually model the salience effects of neurotransmitters such as dopamine.

1.7. GasNets

Research has been done on modeling the effects of non-local signaling in neural networks, specifically on modeling the effects of Nitric Oxide (NO), which is a freely diffusing neurotransmitter[19]. A very abstract model of gas diffusion has given rise to an ANN in which units are capable of modulating the behaviour of other units. These models are called GasNets[19], as they simulate the presence of the gas NO in the environment surrounding the neurons. This form of modulation allows a kind of plasticity in the network in which the intrinsic properties of units are changing as the network operates.

Although much experimentation has been done to modify and manipulate GasNets[20], to the best of our knowledge, GasNets have not been modified to model the way memories associated with emotionally-laden events may be embodied in neural networks, and the way those emotional associations can be recalled when the events are remembered. Furthermore, research on the GasNet model has not been extended to investigate the effects of single-trial leaning in neural networks.

2. Thesis proposal

In this research we propose a new class of ANN, which is an extension of a standard ANN that integrates significantly novel features in a unified model, which we refer to as a Salience-Affected Neural Network (SANN).

Value systems are not the only form of non-local signaling in the cortex. The SANN described models the physiological feature of general non-local signaling in the cortex via neurotransmitters associated with rewards and reinforcement signals broadcast from the limbic system, and demonstrates that this makes one-time learning possible. Thus it throws light on the relation between these important structural and functional aspects of the human brain, and thereby opens up a new class of ANN that may be useful in computational applications.

In particular an SANN demonstrates that these kinds of networks not only enable one-time learning by laying down memory patterns associated with strong salience signals, but enables recall of those salience signals when the relevant stimulus is encountered at later times; that is, it models both the way memories associated with emotionally-laden events may be embodied in neural networks, and the way those emotional associations can be recalled when the events are remembered.

3. Research implications

The research explored here has many implications, especially at computational, biological and psychological levels.

3.1. Computational implications

At a computation level, the modeling of *ongoing regulations*[1] in neural networks will allow us to extract more information (in the form of *salience*) from a standard ANN, without significantly adding to the complexity of the ANN structure. Furthermore, exploring the effect of *heightened affective moments* on a neural network will enable the future training of neural networks with a single training iteration.

3.2. Biological implications

At the biological level this research has limited biological implications, because the ANN model that we adapt is already a significant abstraction of a biological neural network, and poorly models individual biological neurons. Despite this abstraction, this research will provide insight into the way signals flow *locally* in the columns in the cortex through *local connections* in those columns.

An SANN enables modeling of the effects of the peculiar nature of *synaptic connections* as opposed to *gap-junction connections*. With the simplified model of *gap-junction connections*, the signal propagates as the direct transmission of an electric signal. The key point is that a *synaptic connection* allows non-local modulation of local synaptic processes via diffuse projection[3] of neurotransmitters to the synapse region, while the simplified model of *gap-junction connections* does not allow such effects. It is this non-local modulation of local synaptic processes that can be modeled by an SANN.

3.3. Behavioural implications

[1] Refer to section 1.5. Salience

At a behavioural level, an SANN allows the modeling of the effect of affective states on brain activity and on memory. The term *affective* refers to an emotional process, such as subjectively experienced feelings, and an *affective memory* could include adding an emotional tag to significant memories. Affective memories are possible because the ascending systems originate in the limbic system which is the seat of affective states. Thus a SANN potentially represents the effects of emotions on cortical activity, which are known to be significant[21].

4. Research overview

This research initially investigated the literature covering the role of salience in the cortex in both adults and infants, and the shortcomings of current neural network models in approximating activity in the cortex.

After the literature review, we describe the architecture of standard ANNs, the implications of altering the threshold of a node, and the design of a salience-affected neural network.

We then design an application of the SANN, namely face recognition, which requires a feature extraction technique, hence the literature behind different techniques is explored. Once the most appropriate feature extraction technique is chosen, this dissertation then designs a simple software implementation of the application. Thereafter, a general analysis is then performed on the SANN to optimize its performance. This dissertation continues to compare the *single-trial* and *multiple-trial* learning techniques, which is the primary focus of this dissertation.

The feature extraction code is then developed to include a reconstruction algorithm, after which a stand-alone face recognition application is created, with the ability to include salience training.

Results obtained are discussed throughout the dissertation, where applicable. This dissertation then discusses the findings, and conclusions are then drawn from these results. Recommendations are then made based on the conclusions drawn, and finally the scope for future work is discussed.

5. Artificial neural network architecture

5.1. Introduction

Numerous variations of neural networks have been suggested to represent learning and memory in the cortex, as well as many variations in the connectivity of a feed-forward ANN (e.g. fully-connected[22], incompletely-connected[23]).

In this research, we have chosen a generic fully-connected multilayer perceptron (MLP) artificial neural network (ANN) in this research to demonstrate the significance and effect of the salience. The research discussed here can be easily adapted for use with many alternative ANN types, and therefore this research can focus on a specific neural network. The MPL ANN was selected because it is a common neural network, frequently used in the literature, from which many other neural networks are easily derived.

This chapter continues to discuss the architecture of this specific ANN.

5.2. ANN layers

An MLP ANN is a loop-free, *feed-forward* network which has its nodes (called units) arranged in layers, with a unit providing input only to units in the next layer of the sequence. MLP ANNs have the first layer comprised of fixed input units, then there may then be several layers of trainable 'hidden units' carrying an internal representation, and finally there is the layer of output units, also trainable[24], as seen in **Figure 3.**

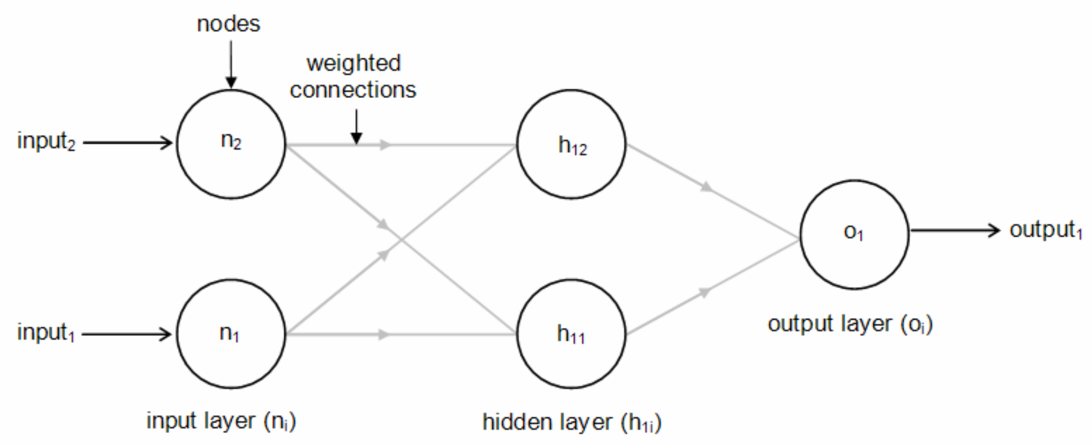

Figure 3: A visualization of a fully-connected multilayer perceptron artificial neural network with a 2-node input layer, a 2-node hidden layer and a single-node output layer. Each inter-nodal connection is weighted, and can be trained. Each node is responsible for summing the weighted signals it receives. The response of each unit, in the form of an output signal, is a function, usually sigmoidal, of the weighted sum of the signals entering the unit.

5.3. ANN dynamic links

The brain's data structure has been approximated by the *graphs,* which are composed of units connected by dynamic links, known as dynamic link architecture (DLA) [25]. Both units and links bear "activity variables" changing within fractions of a second. The variables attached to units refer to the strength of a unit's signal while the variable attached to the link refers to the strength of the connection between units. The neural network approximates this characteristic by assigning weights to each link (**Figure 3**), which can be varied both in magnitude and sign.

5.4. ANN sigmoidal functions

The response of each unit is a function, usually sigmoidal, of the weighted sum of the signals entering the unit. Each node can have a uniquely defined function. The ANN is defined such that each unit has inputs x_k with corresponding weights w_{ik}, and the output x_i is given by the **Equation 1**, with θ_i being the unit's threshold bias[24].

$$x_i = f_i\left(\sum w_{ik} x_k\right) \qquad \textbf{Equation 1}$$

The function f_i is generally a sigmoidal function, commonly defined as **Equation 2**.

$$f_i(x) = \frac{1}{1+e^{-(x+\theta_i)}} \qquad \textbf{Equation 2}$$

For mathematical simplicity, the sigmoidal approximation is used in this research is a tanh-based sigmoidal approximation, defined as **Equation 3**.

$$f_i(x) = \tanh(x+\theta_i) \qquad \textbf{Equation 3}$$

5.5. ANN training

During training, a neural network is repeatedly exposed to a training set of input combinations. The neural network is exposed to each element in the training dataset, and, during training. The ANN will be trained using a process of backpropagation – a process of altering the weights between nodes slightly in favour of the corresponding output determined by the dataset. Over time, and after repeated training iterations, the neural network becomes more likely to produce the required output. In this research, the training iterations are repeated until a certain error margin is achieved.

6. Salience signal

6.1. Embedded salience signal

The SANN proposed in this research is simply an ANN with an additional *input salience signal* feeding into each node. The *input salience signal* is an additional common signal able to affect each unit, and is fed into each individual node. This *input salience signal* has the effect of varying the threshold value of the sigmoidal function at each individual node during training, depending on the node's activation level. The effect of the *input salience signal* will be discussed in detail in section **6.2. General threshold variation**. The *input salience signals* are illustrated as green arrows in **Figure 4.**

Furthermore, each node of an SANN produces a *nodal reverse salience signal* during testing (a measure of the threshold bias for the individual node). The *reverse salience signal* is defined as the summation of the *nodal reverse salience signals* observed at each node. The *nodal reverse salience signal*s are illustrated as red arrows in **Figure 4**, and will be described in more detail in **6.4. Reverse salience signal**.

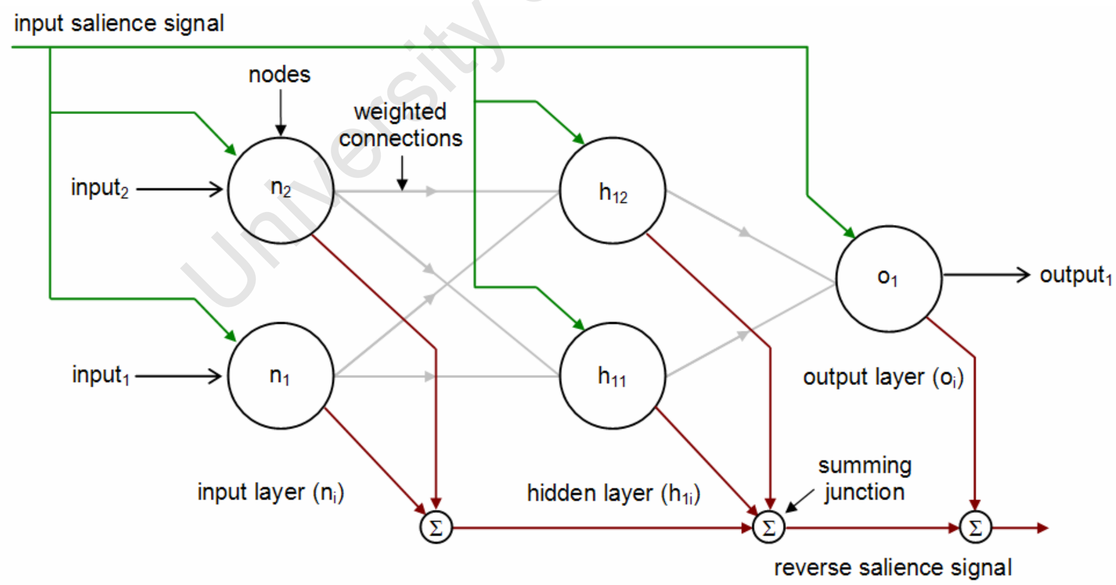

Figure 4: An SANN with a 2-node input layer (n_i), a 2-node hidden layer (h_{1i}), and a single output node (o_i). Each node receives an *input salience signal* (S), which has the effect of varying the

threshold value of the sigmoidal function at each individual node, during training, depending on the node's activation level. From each node, *nodal reverse salience signals* are summed, to create a *reverse salience signal* (S') for the SANN.

Although the *input salience signal* is a common signal, with a single source, it permanently affects the *sigmoidal threshold function* of each node in the ANN independently, by varying the *threshold value* of the sigmoidal function at each node, affecting the output signal for each node (**Figure 5**).

6.2. General threshold variation

Previous research has been performed on varying unit thresholds in *associative nets*[23]. In *associative nets* weights between units are restricted to a binary value of either 0 or 1 by definition[26]. Therefore, the varying of unit thresholds was investigated to more efficiently train the neural network.

Although both an *associative net* and an SANN can vary the threshold values of the sigmoidal function at each unit, the SANN allows for continuous variation of the weights between units, compared to discrete variation as seen in *associative nets*. Furthermore, threshold variations in SANNs model *diffuse projections* from the limbic system to the cortex[8], which is not done in *associative nets.*

The threshold variations proposed in this research for use with the SANN closely mirror the variations described by Phil Husbands et al[19]. This research focuses primarily on the *threshold offset variable*, referred to as b_i by Phil Husbands et al[19]. In the SANN the *threshold offset variable* is referred to in terms of the *old value* (T_{old}) and the *new value* (T_{new}), which relate to the threshold values before and after a round of training respectfully. T_{new} is defined by **Equation 5**.

Our research is different from the research on GasNets by Phil Husbands et al. as it models both the way memories associated with emotionally-laden events may be embodied in neural networks, and the way those emotional associations can be recalled when the events are remembered. Practically speaking, this research introduces an additional *input salience signal* during training, and each node produces a *nodal reverse salience signal* during testing. This research also demonstrates that these kinds of networks enable one-time learning by associating strong salience signals with input combinations.

The SANN has been designed such that during training, the *input salience signal* applied to each node has the effect of pushing active node's thresholds closer to their activation signal (for positive salience), or moving them further away from this signal (for negative salience). In other words, a positive *input salience signal* will encourage a higher output value from the sigmoidal function in future iterations.

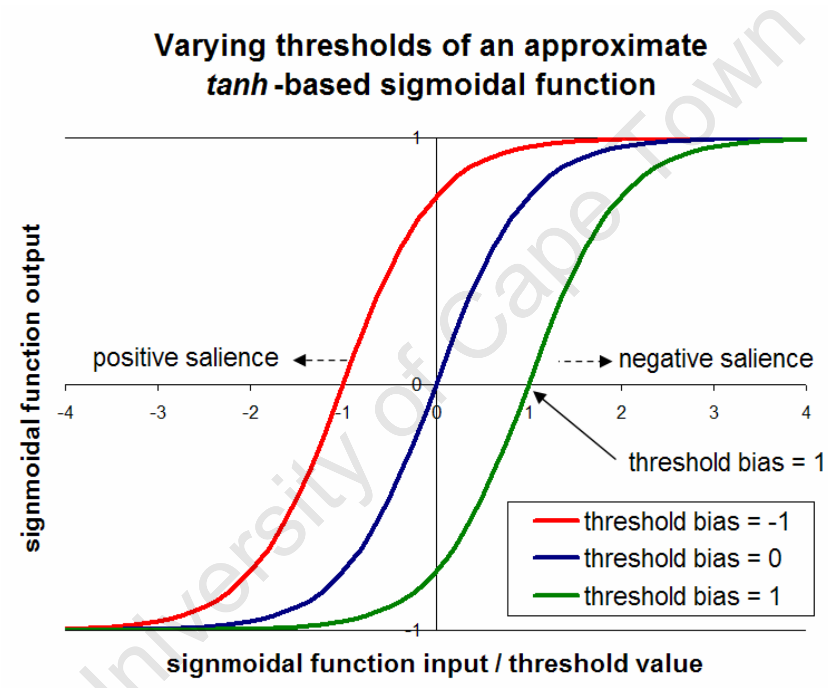

Figure 5: Varying thresholds of an approximate *tanh-based* sigmoidal function. It can be seen that the *input salience signal* applied to each node has the effect of pushing active node's thresholds closer to their activation signal (for positive salience), or moving them further away from this signal (for negative salience).

The SANN is designed such that the *reverse salience signal* is the summation of the *nodal reverse salience signals* observed at each node. *Nodal reverse salience signals*, defined in **6.4. Reverse salience signal** describes the differences between the *acti-*

vation level, and the *node threshold value*, and therefore similar combinations of inputs would produce a similar combination of *nodal reverse salience signals,* and hence, a similar *reverse salience signal*.

6.3. Threshold adjustment definition

For the SANN, the *direction of threshold adjustment* was defined in terms of the *salience signal* (S), and the *instantaneous node activation level* (x_i), as seen in (**Figure 6**). In this section the *instantaneous node activation level* (x_i) is equivalent to output (x_i) used in **Equation 1**.

For example, if the node had a positive activation level and a positive salience level was assigned to it, the threshold should be reduced, to allow the node to produce a higher activation level the next time around. Conversely, increasing the threshold level will reduce the output signal of a node given any input signal.

		Salience	
		+	-
Activation	+	-ve	+ve
	-	+ve	-ve

Figure 6: Direction of threshold adjustment as a function of salience (S) and instantaneous *node activation level* (x_i)

The adjustment factor (D_{adj}), in the integer range [-1, 0, 1], related to the magnitude of the *input salience signal* (S) and *instantaneous node activation level* (x_i) is defined in **Equation 4**.

$$D_{adj} = -\frac{x_i \times S}{\left| x_i \times S \right|}$$ **Equation 4**

To prevent the threshold from increasing/decreasing indefinitely, *threshold limits* were defined. The salience-influence definition adjusts the threshold of the active nodes relative to the pre-defined threshold limits. The magnitude of the adjustment factor used in this research was 20% of the distance between the threshold and the threshold limit, in the appropriate direction. This method ensured that the threshold never exceeded the *threshold limits*.

The final definition of threshold variation for a node, with respect to its *instantaneous node activation level* (x_i) and its *threshold limit* (T_{limit}) is defined as seen in **Equation 5**.

$$T_{new} = T_{old} - B \times \left(T_{old} + \left| x_i \right| \times D_{adj} \times T_{\lim} \right) \qquad \textbf{Equation 5}$$

In **Equation 5**, the variables T_{new}, T_{old} and B represent the *new threshold*, *old threshold* and *salience influence* respectively. The *salience influence* (B) is defined as the rate at which the thresholds were influenced by the salience signals.

6.4. Reverse salience signal

For this research, the *reverse salience signal* (S') was defined as the relationship between its instantaneous *node activation level* (x_i), the *node thresholds* (T_i) and the *sum of the weighted signals* (V_i) received by the units (**Equation 6**).

$$S' = x_i \times \left(T_i - V_i \right) \qquad \textbf{Equation 6}$$

Although various methods could be used, a standard summing method was used in this research to collect the *reverse salience signals* (S') from all the units, for simplicity. In the summing method, the individual *reverse salience signals* (S') are summed, and the results are an indication of the salience attached to the various input-signal combinations.

7. Face recognition application

7.1. Introduction

To test the SANN, an easily-adaptable face recognition software application was developed. Before a computer software program can associate saliencies to a face, each face needs to be broken down into features. This process is known as feature-based analysis. To perform feature-based analysis, methods to separate images into unique components were investigated. This process falls under the category of *unsupervised learning algorithms*, as opposed to *supervised learning algorithms.*

Unsupervised learning algorithms begin with a data set, and result in a final data set that is an approximation of the original data set, by factorization subject to different constraints[27]. Common factorization algorithms include *principal component analysis*, *vector quantization* and *non-negative matrix factorization*, and this thesis will continue to discuss each of them.

7.2. Principal Components

Principal component analysis (PCA) is a statistical technique that transforms an original set of variables into a substantially smaller set of uncorrelated variables that represents most of the information in the original set of variables[29].

This optimization is achieved by transforming to a new set of variables, the principal components (PCs), and which are ordered so that the first few retain most of the variation present in all of the original variables[30].

7.3. Vector Quantization

Vector quantization (VQ) results in clustering the data into mutually exclusive prototypes[27]. VQ has a unary constraint, permitting only a single basis images to represent a face[31].

Typically in a vector quantization application, vectors are sequentially extracted from the original data set, and are individually coded by a memoryless vector quantizer[32].

7.4. Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) is a process by which a matrix of features (W) is created, whereby each image in the data set is comprised of a combination of these features, as determined by the encoding matrix (H).

Unlike PCA and VQ algorithms, NMF does not allow negative entries in the matrix factors W and H. These non-negativity constraints permit the combination of multiple basis images to represent a face[31].

NMF, along with certain other computational theories of object recognition, utilize parts-based representations[31]. Both VQ and PCA are whole-based algorithms while non-negative matrix factorization is considered a parts-based.

7.5. Algorithm comparison

Lee and Seung[31] applied NMF, together with PCA and VQ, to a database of facial images. All three methods learn to represent a face as a linear combination of basis images (**Figure 7**), but with qualitatively different results.

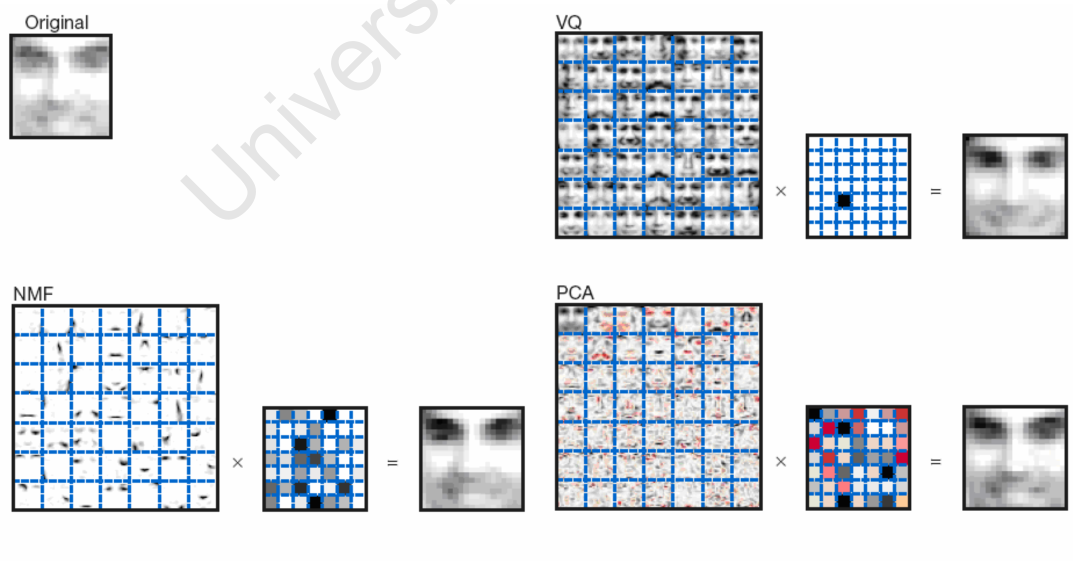

Figure 7: NMF, PCA and VQ learn to represent a face as a linear combination of basis images, but with qualitatively different re-

sults[31]. This figure demonstrates how each of the mentioned feature-extraction techniques creates a unique feature-matrix and weighting-matrix. Every trained image could be approximately reconstructed as a weighted sum of global features (Equation 9). Each technique results in an approximation, as feature based-extraction is an optimization technique.

7.6. Algorithm selection

There is psychological and physiological evidence for parts-based representations in the brain[31]. Non-negative signalling is a precursor for parts-based representations, as discussed in **7.4. Non-Negative Matrix Factorization**. Non-negative signalling, and hence parts-based representations in the cortex, emerges by virtue of two properties, namely the firing rates of neurons are never negative, and the synaptic strengths do not change sign[31].

As mentioned VQ and PCA are both whole-based algorithms and only NMF is a parts-based algorithm, while NMF is a parts-based algorithm. Furthermore, only NMF and VQ restrict their weighting matrices to positive values. Inhibitory connections can be considered neutral connections, rather than negative connections, and hence these positive-value techniques more accurately mirrors cortical neuron functions.

Therefore for this research paper, NMF was selected as the feature-extraction algorithm, as it best approximates the feature-extraction capability of the cortex.

8. Computing with NMF

8.1. Image representation

The image database is regarded as an *n x m* matrix (V) of *non-negative pixel values*, each of the *m* images containing *n* pixels[29].

8.2. Mixing and encoding matrices

The *data matrix* ($V \in R_{n \times m}$) of an NMF with rank *r* is composed of two matrices, $W \in R_{n \times r}$, also called the *mixing matrix*, and $H \in R_{r \times m}$, named as the *encoding matrix*[33]. NMF aims to find an approximate factorization of the data set (V) such that the original data set (V) is can be approximately reconstructed from the *mixing matrix* (W) and *encoding matrix (H)* (see **Equation 7**), in such a way to minimize the reconstruction error.

$$\mathbf{V} \approx \mathbf{WH} \qquad \textbf{Equation 7}$$

8.3. Cost functions

Different cost functions based on the reconstruction error have been defined in the literature, but because of its simplicity and effectiveness[33], the *squared error technique* is used in this research (see **Equation 8**).

$$F = \| \mathbf{V} - \mathbf{WH} \|^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(V_{ij} - (\mathbf{WH})_{ij} \right)^2 , \qquad \textbf{Equation 8}$$

8.4. Minimization of the Cost Function

The aim of the NMF is to minimize the *mean squared error* (F) which is a convex function of W and H (**Equation 8**)[33]. Lee and Seung[27] devised a multiplicative algorithm to minimize the cost function, which is simpler to implement than, and equal in performance to the gradient algorithm designed by Paatero and Tapper[28]. For this research, we therefore chose to adopt the multiplicative algorithm to minimize the cost function.

8.5. Reconstruction algorithm

The approximate reconstruction of the original data set can be attained by the **Equation 9**[33].

$$V_{i\mu} \approx (WH)_{i\mu} = \sum_{a=1}^{r} W_{ia} H_{a\mu} \qquad \textbf{Equation 9}$$

Hoyer[34] warned that while updating H (assuming W is fixed), one must be careful not to do any vector normalization in the iteration. Normalization, in this thesis, is the technique of changing the magnitudes of a vector or matrix so that the spread of data occupies the complete range of possible values [0,1] rather than being particularly high or low. The purpose of normalization is to ensure relatively similar range of values between columns. In the SANN, normalization of the rows of H makes sense when there are many columns, but not when there is a single column, hence vector normalization was omitted during reconstruction.

A NMF reconstruction algorithm was specifically designed for the purpose of this research, based on the **Equation 9** and recommendations provided by Hoyer[34].

9. Software implementation

In order to implement the SANN application in software, a collection of data and certain source code was required.

9.1. Data collection

Although the Matlab R2008a software application was designed to use real-time image capturing via the means of a webcam, additional tests were conducted using face images provided by the CBCL face image database. The face image database (CBCL data) can be found at: http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html

9.2. NMF source code

The NMF section of the source code designed in this research was adapted from an existing NMF algorithm implementation[35]. As mentioned above, a NMF reconstruction algorithm was specifically designed for the purpose of this research, based on the equations and recommendations provided by Hoyer. The code for the NMF reconstruction algorithm was not provided in the original code written by Hoyer.

9.3. ANN source code

The ANN section of the source code designed in this research was originally written in Python and placed in the public domain by Neil Schemenauer (nas@arctrix.com). For this research I adapted and translated the source code from Python into Matlab R2008a code.

The neural network is an artificial abstraction of the computational function of a biological neuron[36]. The program uses the standard backpropagation algorithm to train the MLP. The neural network used in this project has a single hidden layer of units.

9.4. ANN input, hidden and output layers

For this research, the neural network was designed with 49 inputs (limiting the image sizes to 7x7 pixels), a single hidden layer of 10 units and a single output layer. An input layer of 49 nodes was selected because images of 7x7 pixels were large enough to capture recognizable images for users to see, but small enough to ensure that the computations were fast enough to allow for real-time image processing. A hidden layer of 10 nodes was selected as it was large enough to ensure generalized learning for the neural network, and small enough to ensure rapid processing time.

The inputs used for the neural network were the 49 weights assigned to a different element in the mixing matrix (W). For a neural network to operate and be trained an output value must be chosen. For this research, an arbitrary, easy-to-calculate output variable was created from the images in the dataset, namely the *average* pixel grayscale value for each image.

9.5. ANN training and testing

For this research, the neural network was trained with 200 iterations for all tests performed.

9.6. Salience implementation

For this research, the ANN was adapted to include *input salience signal* (S) and *reverse salience sign*al (S'), as described in detail in **6. Salience signal**.

10. General analysis and optimization

10.1. Introduction

Having designed the SANN and implemented the SANN in software, a general analysis and thereafter variable optimization was required. Certain factors were observed and optimized, namely the *residual reverse salience signal*, the *number of units* in the single hidden layer of the SANN, and the *effects of the magnitude* of the salience signal used to train the SANN.

For the duration of the SANN analysis and optimization, the SANN was trained using the *multiple-trial* training technique.

10.2. Residual reverse salience signal value

It was observed, due to the nature of the definition of *reverse salience signal* value, that a neural network, trained with salience, but tested with a "control input" still produced a non-zero *reverse salience signal*, which we will refer to as the *residual reverse salience value.* We define a "control input" as an input with all values set to 0.5. A correlation test was performed (**Figure 8**), and it was observed that this *residual reverse salience signal* value closely followed NN output values. These observations can be explained due to the instantaneous *node activation level* (x_i) (see **Equation 6**).

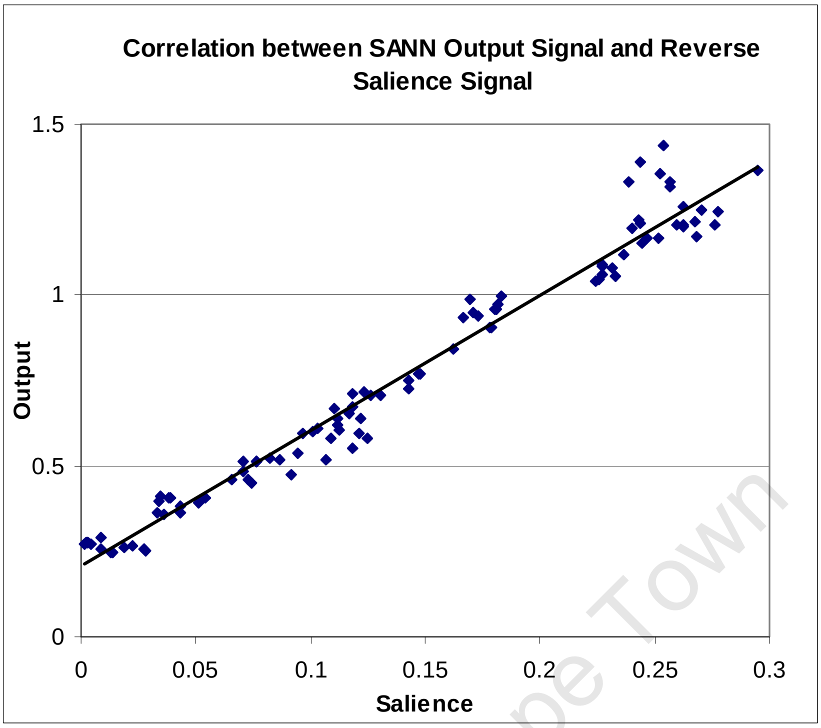

Figure 8: Correlation between SANN output signal and residual reverse salience signal after being trained and tested, without salience, with 100 images from the CBCL database

If a *residual reverse salience signal* did not exist, or if the *residual reverse salience signal* was not related to the output, then there would be a low correlation. From the test performed, a correlation coefficient of **0.985** was calculated. This indicates that a *residual reverse salience signal* exists, that is closely correlated to the output.

From this result, it is clear that the salience must be treated as a *relative value* (the current face relative to a previous value) rather than as an *absolute value*, to ensure that the value is useful.

10.3. Hidden layer units

It is known that that too few hidden layer nodes will not allow for efficient training of the NN, and initially increasing the hidden layer number will allow for the network to be trained with less error. However, further increasing the size of the hidden layer should result in the over training of the NN, at which point the NN will no longer be able to generalize, which will result in an increase in overall NN error.

The number of units in the single hidden layer of the SANN was varied to observe the effect on the learning curve, using a salience amplification of 2. *Salience amplification* is simply a variable used to amplify the salience signal. The reason for this amplification is due to the crude, uncalibrated nature of the *input salience signal* (S) in this research. The test was performed on a range of hidden layer sizes [2:18], and the number of iterations for the NN to show a reduction in error of <= 10% (of the initial error) was recorded. The smaller the number of iterations, the easier the NN is to train (**Figure 9**).

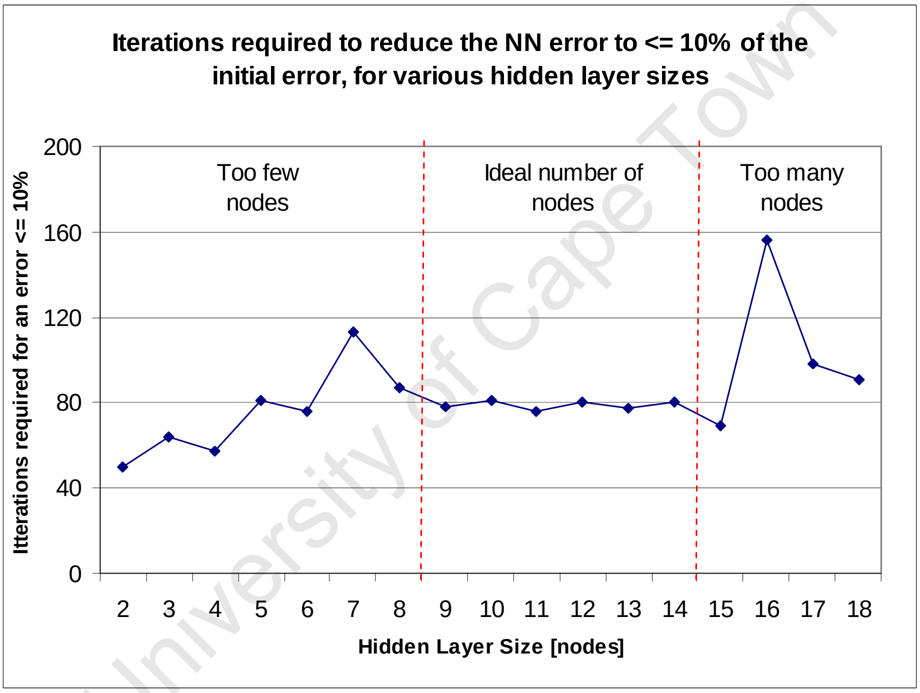

Figure 9: Iterations required to reduce the NN error to <= 10% of the initial error, for various hidden layer sizes.

As expected, as the size of the hidden layer increased (within the boundaries of 9 and 14), the time required to train the NN to <= 10% error decreased. A hidden layer less than 9 was considered too small to effectively train neural network, and a hidden layer size greater than 14 was considered too large (becomes too *specifically* trained). The size selected for a hidden layer in this research was between 10 and 13 nodes.

10.4. Salience signal magnitude

It was suspected that the introduction of an *input salience signal* would affect the training of the neural network. An experiment was performed to observe the effects on the learning curve of the neural network under multiple-trial training, for various magnitudes of salience and number of training iterations (**Figure 10**). A *training iteration* is defined as a single cycle, whereby the ANN is exposed and trained with each element of the dataset only once.

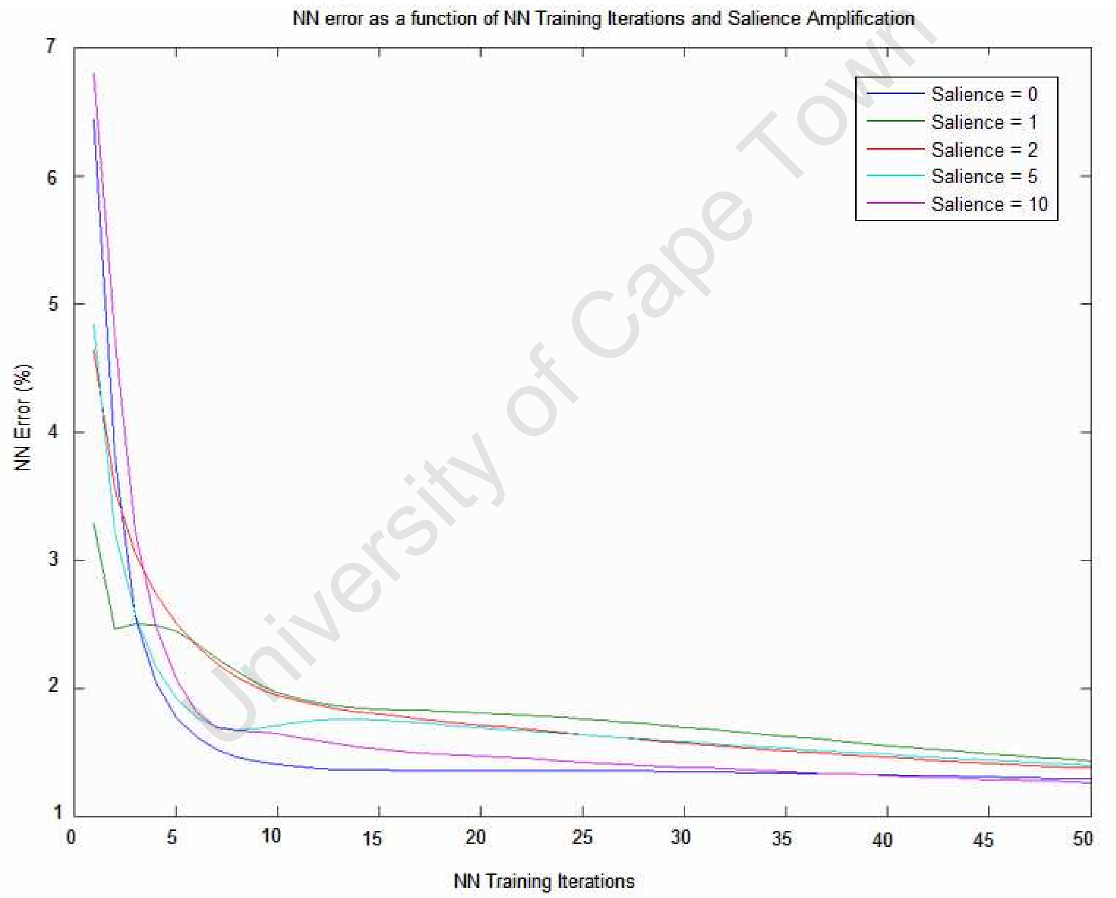

Figure 10: NN Error as a function of training iterations for various magnitudes of *input salience signal* values

It was observed that that the presence of an *input salience signal* retards the speed of response, deforming the shape of the learning curve. As expected, the fastest learning response of the NN occurs when the salience is set to 0. The reason for this is because at low levels of salience, less computational time is required to calculate and make changes to the threshold values. The presence of an *input salience signal* does not, however, affect the learning curve significantly.

11. Results

11.1. Introduction

To highlight the overall effect and significance of threshold variation, and hence salience at each node, multiple-trial training was investigated. Thereafter, single-trial training was simulated to test the hypothesis that an SANN could be trained with a single training iteration.

11.2. Multiple-trial learning technique

11.2.1. Cloud of faces

As designed, we expect the variation of node thresholds to result in a varying *reverse salience signal*. As mentioned earlier, similar SANN input combinations would be expected to produce similar *reverse salience signals*, because the *reverse salience signal* is defined as the summation of the *nodal reverse salience signals* observed at each node. We can therefore predict that faces with similar features will produce similar *reverse salience signals*. In other words, training on a few faces will result in faces with similar features returning similar *reverse salience signal* values.

To test this hypothesis, an SANN was trained using a multiple-trial training method, and with a dataset of 200 unique images of faces. For the training data set used, certain images were similar. For example, images 2, 3, 9, 10 and 11 were each unique images, but were all photos of the same person, and therefore very similar. Images 9, 10 and 11 were trained with an *input salience signal* of value 1, while the remaining images were trained with an *input salience signal* of value 0. The relative *reverse salience signals* were recorded for various faces (**Figure 11**).

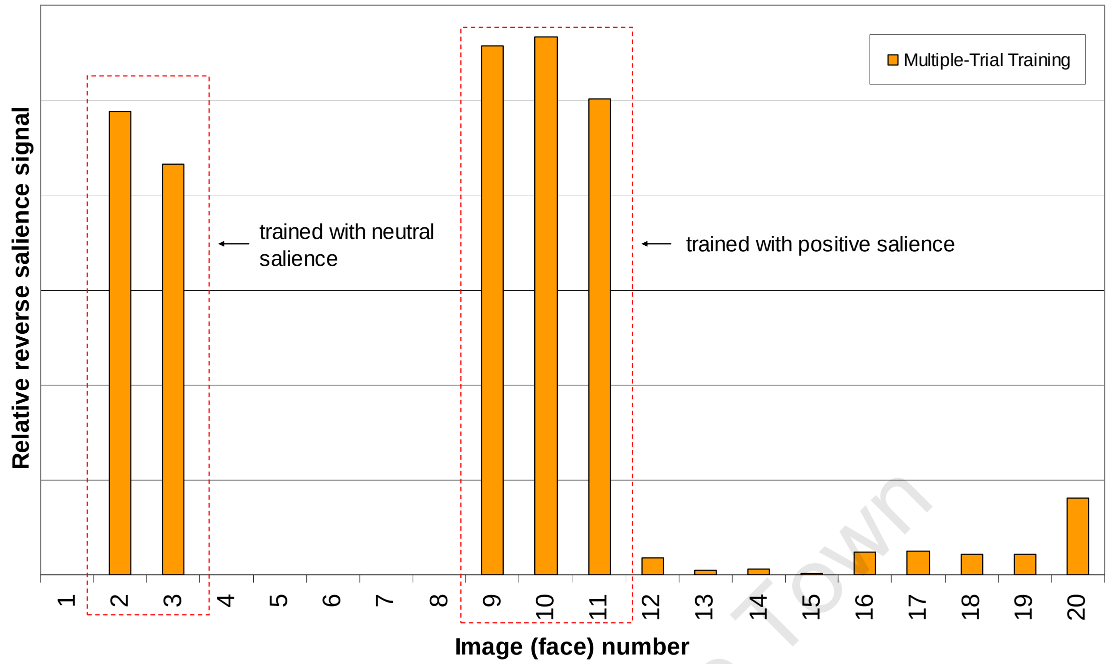

Figure 11: Relative *reverse salience signals* for various images of faces (a sample of 20 input combinations are shown). For the training data set used, certain images were similar. For example, images 2, 3, 9, 10 and 11 were each unique images, but were all photos of the same person, and therefore very similar. Images 9, 10 and 11 were trained with an *input salience signal* of value 1, while the remaining images were trained with an *input salience signal* of value 0. Noticeably, images 2 and 3 returned a relatively high *reverse salience signal* value, despite being trained with an *input salience signal* of value 0.

It was observed that images 2 and 3 returned a relatively high *reverse salience signal* value, despite being trained with an *input salience signal* of value 0. This confirms that our hypothesis was correct, and that training with a few images will result in images with similar features returning similar *reverse salience signal* values.

11.2.2. Conceptual orthogonality

To support Edelman's concept of the value system[8,9] and Beebe and Lachmann's theory of the three principles of salience[11], we desire that the addition of salience not affect the original cortical processes, or in the case of the ANN, the outputs of the nodes. We desire the *reverse salience signal* to be independent of the SANN output signal (i.e. not significantly affecting the output signal of the neural network). However, by altering the threshold values of the nodes, we expect to slightly affect the output signals of the nodes, when compared to a standard ANN.

To test this hypothesis, an SANN was trained using a multiple-trial training method, and with a dataset of 200 unique images of faces. An SANN was first trained with an *input salience signal* of value 0 applied to all input combinations, and then a second SANN was trained with an *input salience signal* of value 1 attached only to images 9, 10 and 11. Thereafter the output signals were recorded and compared, and correlation coefficients were calculated. A low correlation coefficient would indicate large variations in the output values, and a high correlation coefficient would indicate a small variation, which would be desired. Secondly, the *average* values (calculated as the mean value) of the data sets were also observed to test *similarity* between given data sets. If both a high correlation and a low difference in mean were observed, it would imply that the data sets were similar, hence implying that the SANN output and *reverse salience signal* can be regarded as were conceptually orthogonal. *Conceptual orthogonality* in this research refers to multiple values, attributes or functions acting independently of each other.

A correlation coefficient of **0.9998** was calculated between the output sets, hence indicating negligibly small variations, as desired. In other words, the output of the SANN experiences negligible *relative* variations when training with, or without salience. The comparison of the means resulted in means of **0.322** and **0.315** being recorded for salience-free and salience-affected training sets respectively. In other words, the output of the SANN experiences negligible *mean* variations when training with, or without salience.

This result indicates that the *reverse salience signal* can be regarded as significantly independent of, or conceptually orthogonal to the SANN output signal (i.e. with negligible variations).

11.3. Single-trial learning technique

The embedded salience signals in an SANN could enable the training of neural networks with a single training iteration. To test this hypothesis, an SANN was trained using a single-trial training method, and with a dataset of 200 unique images of faces, but without the influence of *input salience signals*. Thereafter a single training iteration was executed on the same SANN, where an *input salience signal* of value 1 attached only to images 9, 10 and 11.

For the single-trial training methods, a *salience amplification factor* was introduced. As mentioned in a previous section, the s*alience amplification factor* is simply a variable used to amplify the salience signal. The amplification factor corresponds to the magnitude of the salience (or significance) attached a certain input combination. Amplification factors in the integer range [1-6] were tested, to observe the effects of varying magnitude of the *input salience signal*, and the *reverse salience signals* were recorded for various faces.

The results were compared to the multiple-trial learning techniques for the same dataset (**Figure 12**). It was observed that a single-trial training method produced a similar *reverse salience signal* profile to the multiple-trial training method, although of a noticeably smaller magnitude.

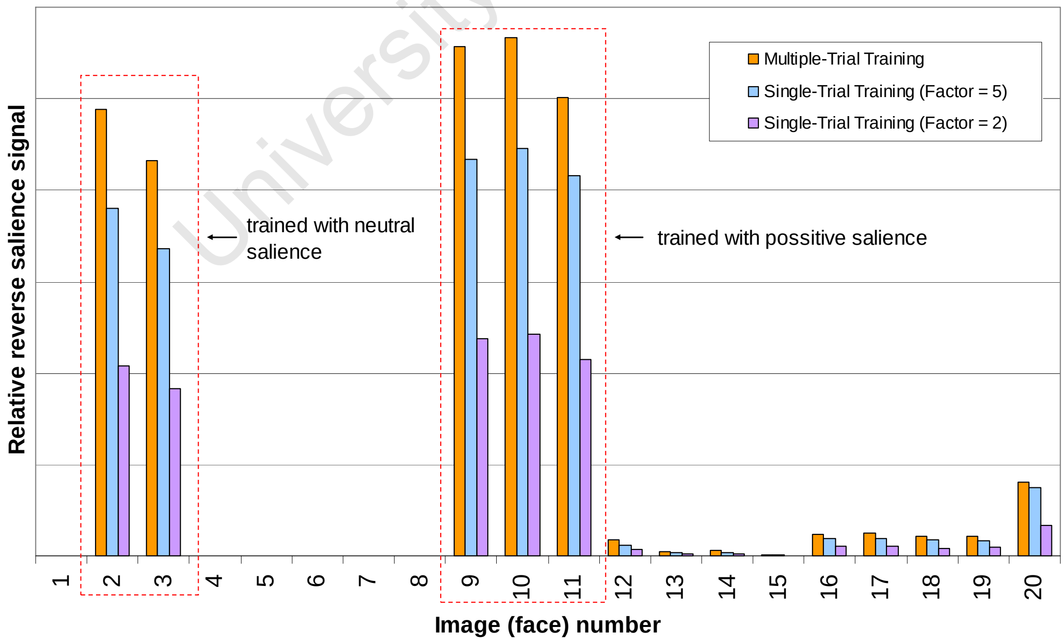

Figure 12: Relative *reverse salience signals* for various images of faces (a sample of 20 input combinations are shown). Figure 11 explains the multiple-trial training method. For the single-trial training methods, an amplification factor was introduced. The amplification factor corresponds to the magnitude of the salience (or significance) attached a certain input combination. Amplification factors in the integer range [1-6] were tested, to observe the effects of varying magnitude of the *input salience signal*, and the *reverse salience signals* were recorded for various faces. Results attained from single-trial training with an amplification factors of 2 and 5 respectively, were plotted alongside multiple-trial training results, as seen above. A single-trial training method produced a similar *reverse salience signal* profile to the multiple-trial training method, although of a noticeably smaller magnitude.

We then designed an experiment to test how similar the salience-profiles were for both single-trial training and multiple-trial training. Correlation coefficients were calculated between the single-trial and the multiple-trial training, as a function of *input salience signal amplification factors*. The *input salience signal amplification factor* is simply the *salience amplification factor* applied to the *input salience signal*. A high correlation coefficient would indicate a similar profile, which we desire.

It was found that the higher the amplification factor (in the tested range), the smaller the variations between the profiles (**Figure 13**). This result indicates that when a SANN is trained with single-trial training, a salience-profile similar to multiple-trail training is established.

The main difference between the training methods is the *magnitude* of the salience attached to various input combinations. Compared to single-trial training, multiple-trial training resulted in a salience-profile with a significantly higher magnitude.

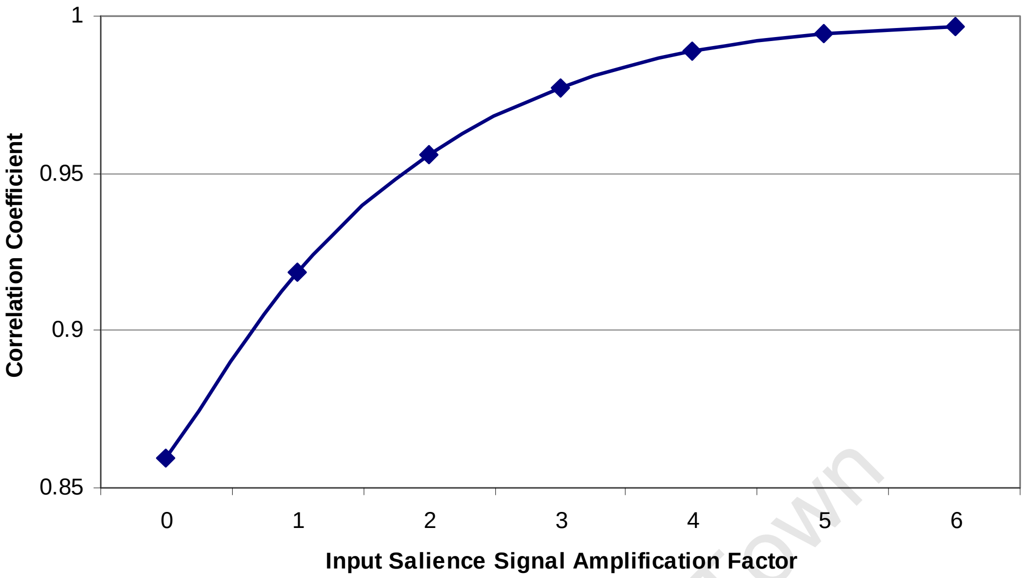

Figure 13: Correlation between single-trial and multiple trial *reverse salience signal* values, for varying *input salience signal* amplification factors during single-trial training. A *correlation* only demonstrated a similarity in data profile, and not in magnitude. An SANN was trained using a single-trial training method, and with a dataset of 200 unique images of faces. First the SANN was trained without the influence of *input salience signals*. Thereafter a single training iteration was executed, where an *input salience signal* of value 1 attached only to images 9, 10 and 11. Amplification factors in the integer range [1-6] were tested. Correlation coefficients were calculated between the single-trial and the multiple-trial training, as a function of *input salience signal* amplification factors. A high correlation coefficient would indicate a similar profile, which we desire.

12. SANN Application

12.1. Introduction

To demonstrate the functionality of an SANN designed in previous chapters, a stand-alone face recognition application was created. This chapter continues to discuss this application.

The face recognition software platform developed earlier was extended to produce a stand-alone face recognition program, using Matlab R2008a, in which real-time images were captured via a webcam. The Matlab R2008a source code used in this research can be found at http://remm.net/leenremm/. This chapter continues to describe the stand-alone application in more depth.

Before we explore the application, it is important to realize that the contents of the data set used to train a SANN are not important. The data set can be a series of similar images, such as the face image database used in the technique comparison tests in pervious chapters, but can also be a random set of images. If the images are similar, then the NMF algorithm will be able to extract more locally defined features, but if images are extremely dissimilar (e.g. wide range of possible images captured by a webcam), the NMF features will be less locally defined. The SANN can therefore be trained with all possible data sets, and although this application is referred to as a face recognition application, it can in fact be seen as a generic SANN application.

For this application the SANN application uses the technique of *multiple-trial learning*. The application has a graphical user interface (GUI) which allows the user to control the training and testing of the SANN. An application manual for the SANN application developed can be found in **Appendix A**.

The application was divided into two logical phases, namely the *training stage* (where the SANN was trained as desired by the user), and the *testing stage* (where the SANN would be tested and a *reverse salience signal* would be produced dependant on the real-time captured images).

12.2. Training stage

Starting the training process captured 100 real-time images over a period of 10 seconds, and assigned both an output and salience to each image. The short time period will result in similar images being taken, and will hence assist the NMF algorithm effectively identify distinct parts. As decided earlier in this research, the output variable simply consisted of the *average pixel value* of an image, for simplicity. The salience value was varied by the *user* in the integer range [-1, 0, 1], as described in **Appendix A**.

The training process included three main components, namely the image capturing, the NMF of the images to create mixing and encoding matrices, and the training of the SANN. The program (by default) created a mixing matrix of 100 basis images, and an encoding matrix, which defined how each of the 100 sample images taken by the webcam could be mapped from the basis images in the mixing matrix.

On completion of the NMF process, both the *mixing matrix* (W) and *encoding matrix* (H) matrices, calculated by the NMF algorithm, were stored by the program to be used during the testing stage. Although only the *mixing matrix* (W) contains the parts-based representation required for the testing stage, both data sets were stored because they are both specific to the training set. Furthermore, once stored, the *encoding matrix* (H) matrix was visually presented to the user (see **Appendix A**).

12.3. Testing stage

The testing stage firstly captured real-time images from the webcam and performed a NMF reconstruction algorithm to determine the best fitting encoding matrix, based on a previously created mixing matrix. Thereafter the encoding matrix was sent into the previously trained SANN, and the *reverse salience signal* was calculated to retrieve the projected salience for that specific image. The process was repeated for every captured frame.

12.4. Reverse salience signal test

To observe the functionality of the SANN application, a simple test was designed and conducted. During the training stage, faces on the left-hand-side and right-hand-side of the images were assigned a base salience value of 0, while those with a face in the center of the image were assigned a positive salience value of 1. The results were recorded for three different positions of the face in the image, and can be seen in **Figure 14**. The low image resolution was chosen to ensure rapid computations, and hence real-time images. The low resolution has no impact on the quality of the salience training. A higher resolution image would only result in range of *reverse salience signals* with higher resolution produced during the testing stage.

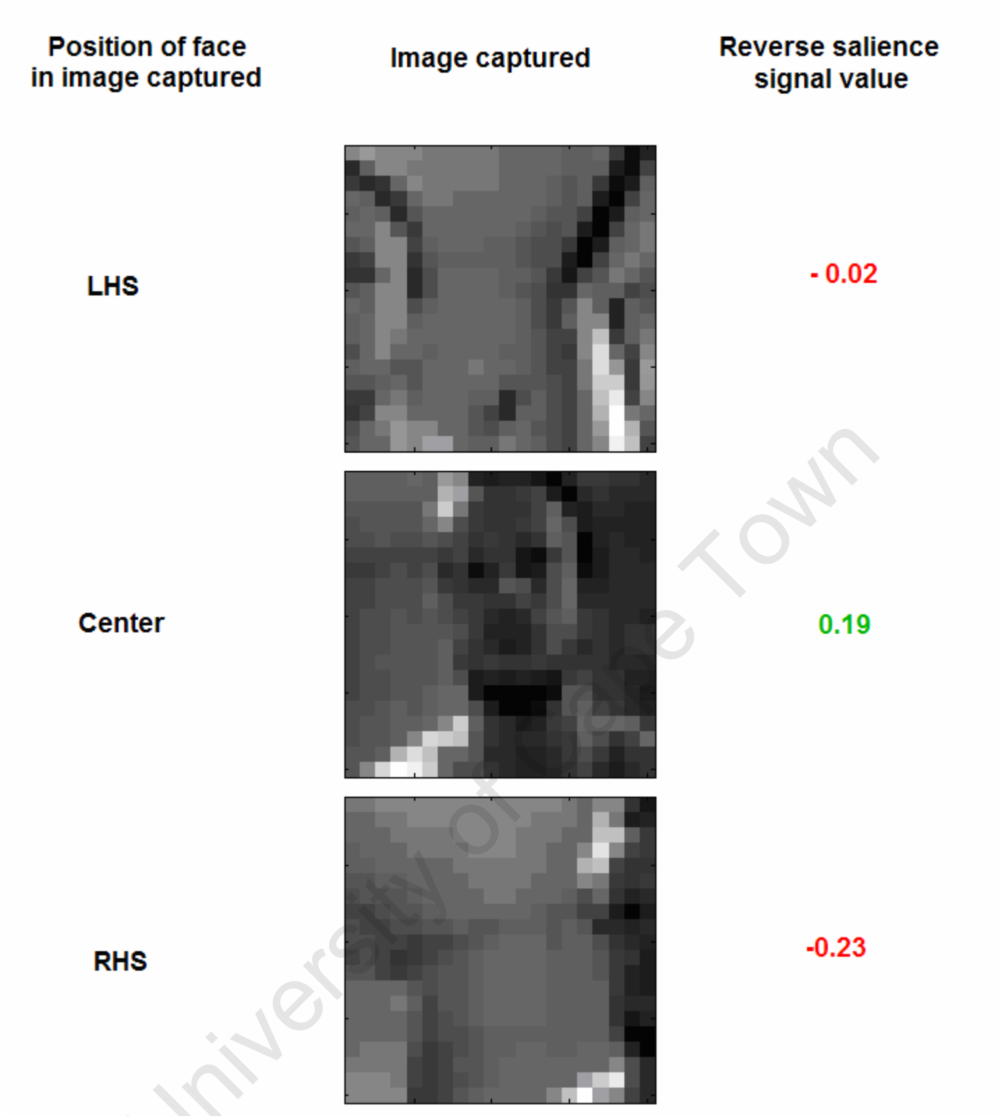

Figure 14: Reverse salience signal values from the SANN application for when a face was on the LHS, Center and RHS of the frame. Only images with a face in the center were trained with a positive salience, and as a result the *reverse salience signal* received during testing corresponded to the training.

12.5. Results

12.5.1. Face Recognition Application

As expected, the *reverse salience signal* values corresponded with close correlation to salience values assigned during the training stage of the SANN application. As seen in **Figure 14**, faces on the left-hand-side and right-hand-side of the images responded with lower *reverse salience signal* values, while those with a face in the center of the image responded with higher *reverse salience signal* values, during the testing stage.

Results indicate that the *reverse salience signal* values corresponded with close correlation to salience values assigned during the training stage of the SANN application, hence the face recognition application proved successful.

12.5.2. NMF Reconstruction Algorithm

As mentioned in a previous chapter, I specifically designed the NMF reconstruction algorithm for this thesis (more specifically for the face recognition application), based on the equations and recommendations provided by Hoyer. It was found that the NMF reconstruction algorithm code successfully reconstruct images, in accordance to **Equation 9**, provided by Hoyer. The only reason that reconstructed images do not always resemble the input images (see **Figure 17**) is due to the limited size and training of the *mixing matrix* (W) of the NMF algorithm.

13. Conclusions

Conclusions were drawn from the results obtained. Significant conclusions have been divided into various categories, as follows.

13.1. SANN

Artificial neural networks can be expanded to create a new class of neural network, namely a salience-affected neural network (SANN). The SANN allows salience-affected combinations of inputs to adjust the thresholds of individual units.

Furthermore, SANNs will produce a positive *reverse salience signal* to faces structurally similar to any previous face trained to be associated with high salience, as the overall *reverse salience signal* is a combination of the *reverse salience signal* provided by each unit (facial feature). This results in a "cloud of input combinations", each with structural similarities to the input combinations trained with salience, receiving a similar salience during the testing stage.

Furthermore, this research has demonstrated that SANN allows for further information (namely salience) to be embedded in an ANN without having to significantly change the *structure* or add features. It is noted, however, that this research used an arbitrary output variable, which, under normal circumstance would be tailored to serving a useful function. Applications that currently use an ANN will now be able to add salience signals without altering their structure or size of the ANN, as a result of the research performed in this thesis.

13.2. Multiple-trial training

Multiple-trial training allows for threshold variations to occur (as does single-trial training), at each unit in the neural-network, during training.

Testing the SANN after multiple-trial training confirmed that faces with similar features produce similar *reverse salience signals* during testing.

Multiple-trial training further demonstrated that the *reverse salience signal* can be regarded as independent of, or *conceptually orthogonal* to the SANN output signal.

13.3. Single-trial training

In contrast to standard an ANN, the SANN allows single-trial training to occur. Single-trial training also allows for threshold variations to occur (as does multiple-trial training), at each unit in the neural-network, during training.

The single-trial training method produced similar *reverse salience signal* profiles to the multiple-trial training method. This indicates that an SANN can be trained with a single training iteration, with similar effects to multiple-trial training.

It was found that the higher the *input salience signal* amplification factor (in the tested range), the closer single-trial training approximated multiple-trial training.

13.4. Theory of diffuse projections

From observing the performance of an SANN, we conclude that an SANN performs in support of the suggested model of *diffuse projections* and *value systems*, described by Edelman[8,9]. *Value systems* have been supported by the training and testing of the SANN with both an *input* and *reverse salience signal*.

13.5. Three principles of salience

The results obtained in this thesis support two of the three principles of salience described by Beebe and Lachmann[11], namely *ongoing regulations*, and *heightened affective moments*, as described in previous chapters. *Ongoing regulations* and *heightened affective moments* have been supported by the training and testing of the SANN under multiple-trial and single-trial conditions respectively.

13.6. Face recognition application

A face recognition application, as designed in this research, can be adapted to provide a *reverse salience signal* value which will correspond (with close correlation) to salience values assigned during the training stage.

As mentioned, the application is not restricted to a data set of faces, but rather any data set consisting of a series of images. This adaptation could therefore have significant application in video-analysis software, such as in surveillance software, as it allows for a *reverse salience signal* to be embedded in an artificial neural network. The use of an SANN instead of an ANN in video-analysis would have the benefit of not complicating the structure of the neural network, while being able to train for salience signals, which could be trained to detect various useful characteristics or images profiles.

14. Recommendations and future work

This research provides evidence that an SANN can be successfully developed to produce desired results. Based on the conclusions drawn, we can recommend that SANN be applied to applications attempting to simulate cognitive processes that otherwise would use a standard artificial NN.

This research is only a foundation for future work in this field. Future work on the SANN should begin with the following aspects being investigated in depth:

- It is recommended that future work could include the conversion of the ANN to a more biologically accurate NN. A possible modification would be to use a *spiking neuron* model. This would be in the interest of more accurately simulating the ability of the brain to direct attention to certain combinations of stimuli.

- The SANN can be designed so that salience has alternative effects on the sigmoidal functions of each node. For this research, the *threshold value* was changed, but future work should investigate changing other characteristic features of the sigmoidal function, such are the *shape*, or the *gradient*.

- The neural network used in this project has a *single hidden layer* of units. The variation of the number of hidden layers was not observed in this research, and an experiment should be conducted to test for changes in performance of the SANN.

- A *standard summing method* was used to collect the *reverse salience signals* from all the units, for simplicity. For optimal performance of an SANN for specific applications, this method could be optimized differently.

- For this research, the *magnitude of the adjustment factor* used for each node of the SANN was 20% of the distance between the threshold and the threshold limit. This factor still requires optimization, (although it is probably be application-specific), in order to achieve optimal performance from the SANN.

References

1. Austin, J. 1997. Associative memory. *Handbook of Neural Computation.* Oxford: Publishing Ltd and Oxford University Press.
2. Bower, J.M., and Beeman, D. (1998). *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System* (2nd Ed.). New York: Springer-Verlag.
3. Edelman, G. 2004. *Wider than the sky: A revolutionary view of consciousness.* USA: Yale University Press.
4. Feldman, J.A. 2008. *From Molecule to Metaphor: A Neural Theory of Language*. Paperback ed. Massachusatts: MIT Press.
5. Panksepp, J. 2004. Affective neuroscience: The foundations of human and animal emotions. illustrated ed.Oxford: Oxford University Press US.
6. Bishop, C. M. (1999). Neural Networks for Pattern Recognition. (Oxford: Oxford University Press)
7. Hawkins, J. & Blakeslee, S. (2005). On Intelligence. (Times Books: Henry Holt and Company, New York)
8. Edelman, G. M., Tononi, G. 2000. *Consciousness: How matter becomes imagination*. Allen Lane, USA.
9. Edelman, G. M., Reeke, G., Gall, W., Tononi, G., Williams, D., Sporns, O. 1992. Synthetic neural modeling applied to a real-world artifact. *Proceedings of the National Academy of Sciences USA*. 89:7267-7271
10. Hawkins, J. & Blakeslee, S. 2005. *On Intelligence*. Times Books: Henry Holt and Company, New York.
11. Beebe, B. & Lachmann, F.M. 1994. Representation and Internalization in Infancy: Three Principles of Salience. *Psychoanalytic Psychology.* 11:127-165.
12. Rock, I., Heimer, W. 1959. Further Evidence of One-Trial Associative Learning. *American Journal of Psychology*. 72(1):1-16.
13. Armstrong, C. M., DeVito, L. M. & Clelan, T. A. 2006. One-Trial Associative Odor Learning in Neonatal Mice. *Chem. Senses*. 31:343–349.
14. Lee-Teng, E. & Sherman, M. 1966. Memory consolidation of single-trial learning in chicks. *Proc Natl Acad Sci USA*. 56(3):926–931.
15. Waelti, P., Dickinson, A., Schultz, W. 2001. Dopamine responses comply with basic assumptions of formal learning theory. *Nature*. 412:43-48.

16. Berridge, K.C. & Robinson, T.E. 1998. What is the role of dopamine in reward: hedonic impact, reward learning, or incentive salience? *Brain Research Reviews.* 28(3):309-369.
17. Montague, P. R., Hyman, S. E. & Cohen, J. D. 2004. Computational roles for dopamine in behavioural control. *Nature*. 431:760-767.
18. Downara, J., Mikulisa, D.J. & Davisa, K.D. 2003. Neural correlates of the prolonged salience of painful stimulation. *NeuroImage.* 20(3):1540-1551.
19. Husbands, P., Smith, T.M.C., O'Shea, M., Jakobi, N., Anderson, J. & Philippides, A. 1998. Brains, Gases and Robots. In Niklasson, N. et al. (editors). *Proceedings of the 8th International Conference on Artificial Neural Networks: ICANN98*. Springer.
20. P. Husbands. 1998. Evolving robot behaviours with diffusing gas networks. In P. Husbands and J.A. Meyer (editors): *Evolutionary Robotics*. 71–86. LNCS vol 1468, Springer-Verlag.
21. Damásio, A. 1995. *Descartes' error: Emotion, reason, and the human brain.* New York: Avon Books.
22. McNaughton, B. L. & Morris, R. G. M. 1987. Hippocampal synaptic enhancement and information storage within a distributed memory system. *Trends Neurosci*. 10:408-415.
23. Buckingham, J. & Willshaw, D. 1993. On setting unit thresholds in an incompletely connected associative net. *Network*. 4(4):441-459.
24. Aribib, M.A. 2003.The Elements of Brain Theory and Neural Networks. In *The Handbook of Brain Theory and Neural Networks*. Ed. M.A. Aribib. Second ed.Massachusatts: MIT Press. 1-24.
25. von der Malsburg, C. 2003. Dynamic Link Architecture. *The Handbook of Brain Theory and Neural Networks.* 365-368.
26. Buckingham, J. & Willshaw, D. 1992. Performance characteristics of the associative net. *Network*. 3:407-414.
27. Lee, D. D. Seung, H. S. 2001. Algorithms for Non-negative Matrix Factorization. *Advances In Neural Information Processing Systems.* (13):556-562.
28. Paatero, P. and Tapper, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.
29. Dunteman, G.H. 1989. *Principal Components Analysis.* Newbury Park: SAGE.

30. Jolliffe, I.T. 2002. *Principal Components Analysis.* 2, illustrated ed. New York: Springer.
31. Lee, D. D. Seung, H. S. 1999. Learning the parts of objects by non-negative factorization. *Nature (London).* (6755):788-791.
32. Gersho, A. & Gray, R.M. 1992. *Vector quantization and signal compression.* Boston: Kluwer Academic Publishers.
33. Hoyer, P.O. 2004. Non-negative Matrix Factorization with Sparseness Constraints. *J.Mach. Learn. Res.* 5:1457-1469.
34. Hoyer, P.O. & Remmelzwaal, L.A. 2009. *nmfpack Question.* RSA: Email.
35. Hoyer, P.O. August 2006. *NMF Pack.* Helsinki, Finland: http://www.cs.helsinki.fi/.
36. Maass, W. 2003. Spiking Neurons, Computation with. *The Handbook of Brain Theory and Neural Networks.* :1080-1083.

Bibliography

Additional sources, not directly referenced in the thesis include:

- Ghosh, A. & Shatz, C.J. 1992. Involvement of subplate neurons in the formation of ocular dominance columns. *Science.* 255:1441-1443.
- Kanold, P.O., Kara, P., Reid, R.C. and others. 2003. Role of subplate neurons in functional maturation of visual cortical columns. *Science.* 301:521-525.
- Kanold, P.O. 2004. Transient microcircuits formed by subplate neurons and their role in functional development of thalamocortical connections. *NeuroReport.* 15(14):2149-2153.
- Taylor, S.F., Phan, K.L., Britton, J.C. and others. 2005. Neural Response to Emotional Salience in Schizophrenia. *Neuropsychopharmacology.* 30:984-995.

Appendixes

Appendix A: SANN application operation manual

Training Stage

After initializing the connection between the SANN application and the webcam device, the user reaches the training stage of the SANN application. The training stage presents the user with an interface shown in **Figure 15**. The training commands available to the user are highlighted in box A.

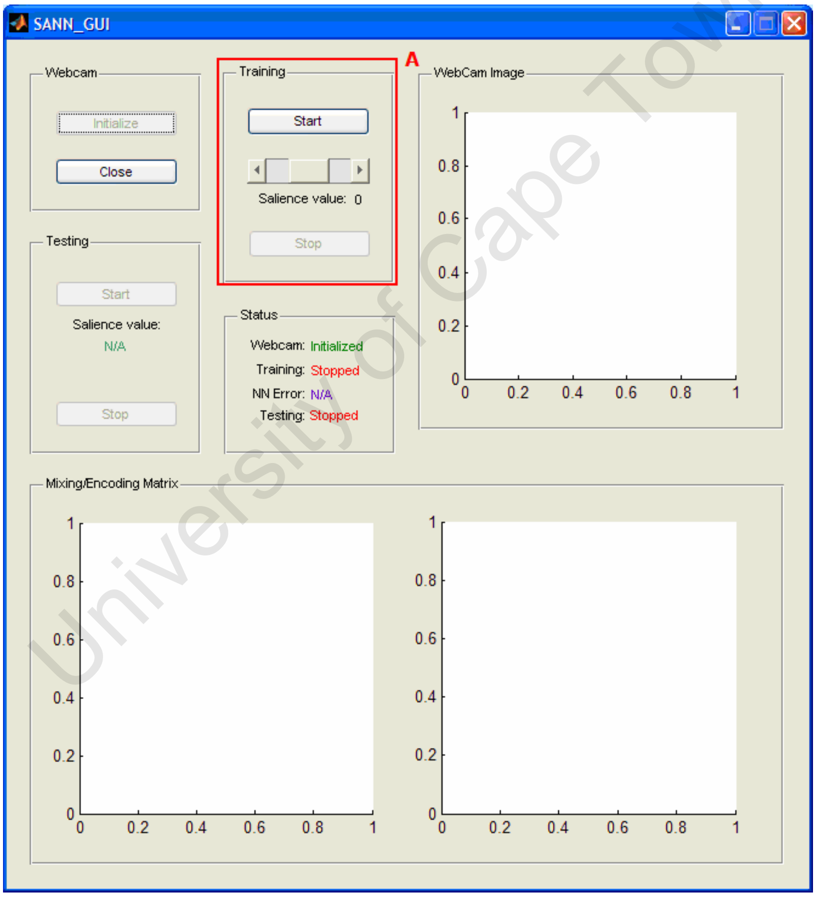

Figure 15: Pre-training stage of the SANN application

During the training period, the user will be presented with an interface layout similar to the one shown in **Figure 16**. The user will have the option to vary the salience range by using the features in Box B during the 10 second training window. The images captured will be shown in real-time in Box C.

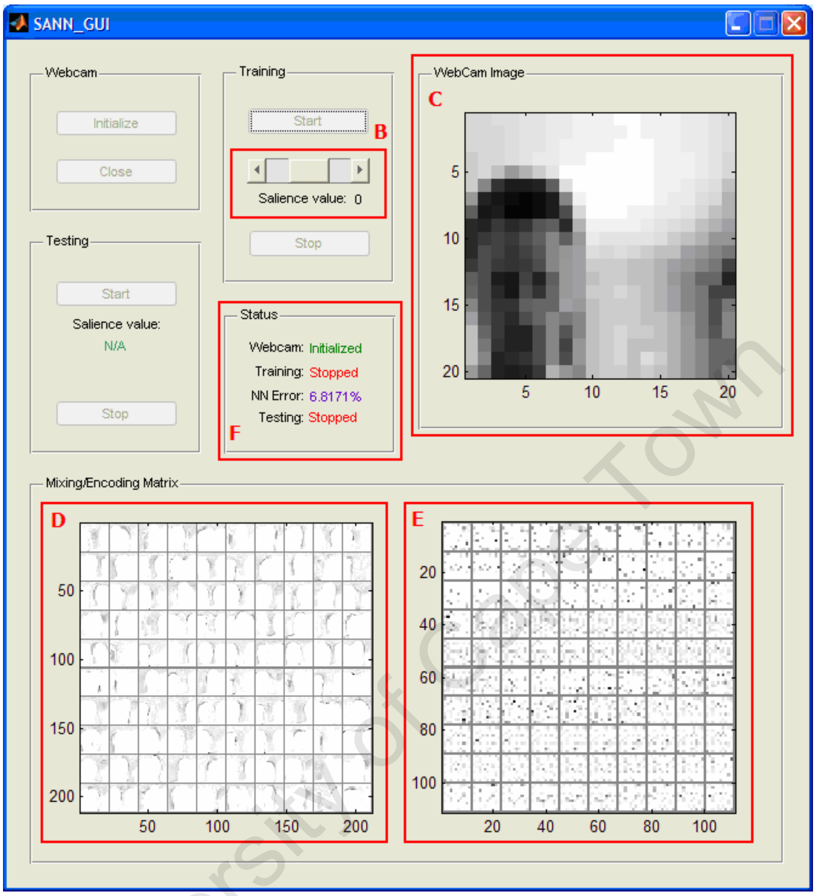

Figure 16: Post-training stage of the SANN application

On completion of the NMF process, the mixing and encoding matrices, calculated by the NMF algorithm, will be shown in box D and box E respectively. Box F presents the user with the applications status, updated in real-time.

On completion of training the SANN, the overall error will be stated in box F. The SANN error is a percentage and reflects the final error after training as a proportion of the initial error before training. The smaller the error percentage, the more accurately the SANN reflects the desired outputs.

Testing Stage

The testing stage is initialized (**Figure 17**) by using controls found highlighted in Box G. The real-time images captured by the webcam are shown in box L. The testing stage of the SANN application utilizes the trained SANN created in the training stage.

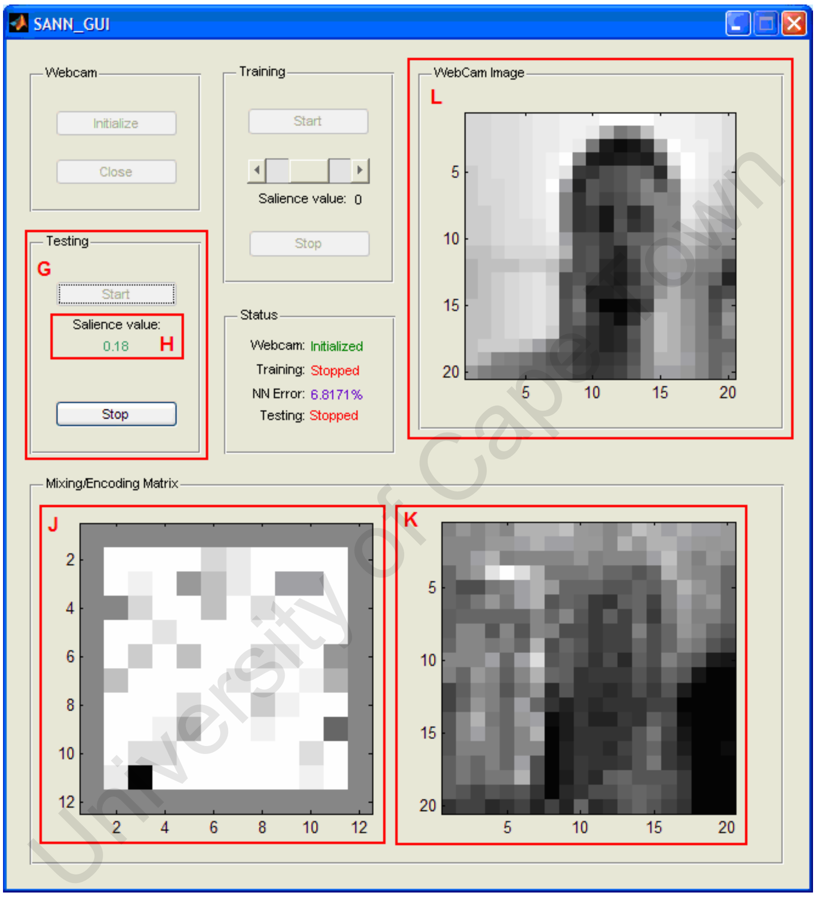

Figure 17: Testing stage of the SANN application

In real-time, the best-fitting encoding matrix calculated by the reverse NMF algorithm is shown in box J, while the recreated face (based on the new encoding matrix) is shown in box K.

Although the reconstructed image in box K may differ from the real-time image in box L, it is the best reconstruction the NMF algorithm achieved using the previously created basis images stored in the mixing matrix.