

# **AUTOMATIC DETECTION OF PULMONARY EMBOLISM USING COMPUTATIONAL INTELLIGENCE**

**Simon J. Scurrall**

A dissertation submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, October 2006

# Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

Signed this \_\_\_\_ day of \_\_\_\_\_ 20\_\_\_\_

---

Simon J. Scurrall.

# Abstract

Pulmonary embolism (PE) is a potentially fatal, yet potentially treatable condition. The problem of diagnosing PE with any degree of confidence arises from the non-specific nature of the symptoms. In difficult cases, multiple tests will need to be performed on a patient before an accurate diagnosis can be made. These tests include Ventilation-Perfusion (V/Q) scanning, Spiral CT, leg ultrasound and d-Dimer testing. The aim of this research is to test the performance of using neural networks, namely Bayesian Neural Networks, to make a diagnosis based on available information. The information contains a set of 12 V/Q scans which have been processed, and from which features have been extracted to provide inputs to the neural network. This system will act as a second opinion, and is not intended to replace an experienced clinician.

The V/Q scans are analysed using image processing techniques in order to segment the lung from the background image and to determine if any abnormalities are present in the lung. The system must be able to discriminate between a genuine case of PE and of other diseases showing similar symptoms such as tuberculosis and parenchymal lung disease. Relevant features to be used in classification were then extracted from the images.

The goal of this system is to make use of Bayesian neural networks. Using Bayesian networks, confidence levels can be calculated for each prediction the network makes. This makes them more informative than traditional multi layer perceptron (MLP) networks. The V/Q scans themselves are greyscale images of [256x256] size. In order to reduce redundancy and increase computational speed, Principal Component Analysis (PCA) is used to obtain the most significant information in each of the scans.

Usually the Gold Standard for such a system would be pulmonary angiography, but in this case the Bayesian MLP (BMLP) is trained based on diagnosis by an

experienced nuclear medicine physician. The system will be used to look at new cases for which the accuracy of the system can be established. Results showed good training performance, while validation performance was reasonable. Intermediate cases proved to be the most difficult to diagnose correctly.

# Acknowledgements

Firstly, I would like to thank the staff of the Chris Hani Baragwanath Hospital, Johannesburg General Hospital and the Donald Gordon Medical Centre for their assistance in obtaining the imaging data. A special thanks must go to Dr Carlos Liebhaber.

Thanks must also go to my two supervisors: Prof Tshilidzi Marwala and Prof David Rubin, who's insight and expertise made this research possible.

I would also like to thank Kentron for the bursary which enabled me to conduct this research.

Lastly, I would like to thank my family, who have provided great support for the duration of this research.

# Contents

|                                               |             |
|-----------------------------------------------|-------------|
| <b>Declaration</b>                            | <b>i</b>    |
| <b>Abstract</b>                               | <b>ii</b>   |
| <b>Acknowledgements</b>                       | <b>iv</b>   |
| <b>List of Figures</b>                        | <b>x</b>    |
| <b>List of Tables</b>                         | <b>xiii</b> |
| <b>Nomenclature</b>                           | <b>xv</b>   |
| <b>Chapter 1: Background</b>                  | <b>1</b>    |
| 1.1 Diagnosis of Pulmonary Embolism . . . . . | 1           |
| 1.1.1 Clinical Prediction Rules . . . . .     | 2           |
| 1.1.2 Imaging Techniques . . . . .            | 4           |
| 1.1.3 The PIOPED Investigation . . . . .      | 7           |
| 1.2 Overview of Research Report . . . . .     | 8           |
| 1.3 Key Achievements of the Study . . . . .   | 9           |

|                                               |           |
|-----------------------------------------------|-----------|
| <b>Chapter 2: Implementation</b>              | <b>11</b> |
| 2.1 The Role of Automated Diagnosis . . . . . | 11        |
| 2.2 Previous Works . . . . .                  | 11        |
| 2.3 Data Collection . . . . .                 | 12        |
| 2.4 The Gold Standard . . . . .               | 13        |
| 2.5 System Block Diagram . . . . .            | 13        |
| 2.5.1 Image Segmentation . . . . .            | 14        |
| 2.5.2 Image Alignment . . . . .               | 18        |
| 2.5.3 Image Subtraction . . . . .             | 21        |
| 2.5.4 Feature Extraction . . . . .            | 22        |
| 2.5.5 Input Selection . . . . .               | 29        |
| 2.5.6 Classification . . . . .                | 30        |
| 2.6 System Training . . . . .                 | 33        |
| <b>Chapter 3: Software</b>                    | <b>34</b> |
| 3.1 PEDiag . . . . .                          | 34        |
| 3.2 Medical Image Viewer . . . . .            | 35        |
| <b>Chapter 4: Results and Discussion</b>      | <b>36</b> |
| 4.1 Training Performance . . . . .            | 37        |
| 4.1.1 16x16 Image Size . . . . .              | 37        |

|                   |                                                                   |           |
|-------------------|-------------------------------------------------------------------|-----------|
| 4.1.2             | 32x32 Image Size . . . . .                                        | 38        |
| 4.1.3             | 64x64 Image Size . . . . .                                        | 39        |
| 4.2               | Validation Performance . . . . .                                  | 40        |
| 4.2.1             | 16x16 Image Size . . . . .                                        | 40        |
| 4.2.2             | 32x32 Image Size . . . . .                                        | 41        |
| 4.2.3             | 64x64 Image Size . . . . .                                        | 42        |
| 4.3               | Individual Class Performance : All Image Sizes . . . . .          | 43        |
| 4.4               | Discussion . . . . .                                              | 43        |
| 4.4.1             | Performance . . . . .                                             | 43        |
| 4.4.2             | Variance Retained . . . . .                                       | 44        |
| 4.4.3             | Image Size & Number of Inputs . . . . .                           | 44        |
| 4.4.4             | Sensitivity, Specificity, Positive and Negative predictive values | 46        |
| 4.5               | Conclusion . . . . .                                              | 47        |
| <b>Appendix A</b> |                                                                   | <b>A1</b> |
| A.1               | Layout of User Structure . . . . .                                | A1        |
| <b>Appendix B</b> |                                                                   | <b>B1</b> |
| B.1               | Installation . . . . .                                            | B1        |
| B.2               | Initial Startup . . . . .                                         | B1        |
| B.3               | Basic Controls . . . . .                                          | B2        |



|                   |                                                       |           |
|-------------------|-------------------------------------------------------|-----------|
| B.3.1             | Lung Viewing Windows . . . . .                        | B2        |
| B.3.2             | Case Selection . . . . .                              | B3        |
| B.3.3             | Image Selection . . . . .                             | B3        |
| B.3.4             | Presets . . . . .                                     | B3        |
| B.3.5             | Probability of Pulmonary Embolism . . . . .           | B3        |
| B.3.6             | Palette Adjustment . . . . .                          | B4        |
| B.3.7             | Submit Results . . . . .                              | B4        |
| B.4               | Saving Progress . . . . .                             | B4        |
| B.4.1             | Loading Progress . . . . .                            | B4        |
| B.4.2             | Recovery Using the <i>Autosave</i> Function . . . . . | B4        |
| B.5               | Submitting Results . . . . .                          | B5        |
| <b>Appendix C</b> |                                                       | <b>C1</b> |
| C.1               | Sensitivity . . . . .                                 | C1        |
| C.2               | Specificity . . . . .                                 | C2        |
| C.3               | Positive Predictive Value . . . . .                   | C2        |
| C.4               | Negative Predictive Value . . . . .                   | C2        |
| <b>Appendix D</b> |                                                       | <b>D1</b> |
| D.1               | Image Processing Functions . . . . .                  | D1        |
| D.1.1             | imhotspots.m . . . . .                                | D1        |

|       |                                                           |     |
|-------|-----------------------------------------------------------|-----|
| D.1.2 | imfshs.m . . . . .                                        | D3  |
| D.1.3 | imsegment.m . . . . .                                     | D4  |
| D.1.4 | imartifacts.m . . . . .                                   | D6  |
| D.1.5 | imfillcontour.m . . . . .                                 | D8  |
| D.1.6 | imgenerategrid.m . . . . .                                | D9  |
| D.1.7 | imgrid.m . . . . .                                        | D10 |
| D.1.8 | imquotient.m . . . . .                                    | D11 |
| D.1.9 | imselect.m . . . . .                                      | D12 |
| D.2   | Image Alignment Functions . . . . .                       | D13 |
| D.2.1 | imalignsub_ga.m . . . . .                                 | D13 |
| D.2.2 | objective_function.m . . . . .                            | D15 |
| D.3   | Feature Extraction and Classification Functions . . . . . | D17 |
| D.3.1 | transform_inputs.m . . . . .                              | D17 |
| D.3.2 | classify_inputs.m . . . . .                               | D19 |
| D.4   | Miscellaneous Functions . . . . .                         | D21 |
| D.4.1 | writecase.m . . . . .                                     | D21 |

# List of Figures

|      |                                                                                                                  |    |
|------|------------------------------------------------------------------------------------------------------------------|----|
| 2.1  | System block diagram . . . . .                                                                                   | 14 |
| 2.2  | Full scale histogram stretch . . . . .                                                                           | 15 |
| 2.3  | Image hotspot removal . . . . .                                                                                  | 16 |
| 2.4  | Image Iso-Contours . . . . .                                                                                     | 17 |
| 2.5  | Dethroating of image . . . . .                                                                                   | 18 |
| 2.6  | Image segmentation . . . . .                                                                                     | 18 |
| 2.7  | Result of image alignment using Schepp-Logan digital phantoms . .                                                | 20 |
| 2.8  | Polar co-ordinate representation of lung . . . . .                                                               | 22 |
| 2.9  | PCA data reduction for varying sizes of input images. All graphs are<br>from an anterior quotient image. . . . . | 24 |
| 2.10 | Reconstruction of posterior lung image [16x16] using varying amounts<br>of VR . . . . .                          | 27 |
| 2.11 | Reconstruction of posterior lung image [32x32] using varying amounts<br>of VR . . . . .                          | 27 |
| 2.12 | Reconstruction of posterior lung image [64x64] using varying amounts<br>of VR . . . . .                          | 28 |
| 2.13 | Illustration of the Statistical Overlay Function ( $\delta$ ). . . . .                                           | 30 |

|      |                                               |    |
|------|-----------------------------------------------|----|
| 2.14 | Multi-Layer perceptron architecture . . . . . | 32 |
| 3.1  | Screen shot of PEDiag. . . . .                | 35 |
| 4.1  | Individual Class Performance . . . . .        | 43 |
| 4.2  | Performance vs. VR for 10 inputs . . . . .    | 45 |
| 4.3  | Performance vs. VR for 20 inputs . . . . .    | 45 |
| 4.4  | Performance vs. VR for 30 inputs . . . . .    | 46 |
| 4.5  | ROC curve . . . . .                           | 48 |
| A.1  | Structure of User . . . . .                   | A1 |
| B.1  | Medical Image Viewer Screenshot . . . . .     | B2 |
| B.2  | Save progress . . . . .                       | B4 |
| B.3  | Loading progress . . . . .                    | B5 |
| B.4  | Recover using autosave . . . . .              | B5 |
| B.5  | Entering user details . . . . .               | B5 |
| B.6  | Results success . . . . .                     | B6 |
| D.1  | imhotspots.m - Flow chart . . . . .           | D2 |
| D.2  | imfshs.m - Flow chart . . . . .               | D3 |
| D.3  | imsegment.m - Flow chart . . . . .            | D5 |
| D.4  | imartifacts.m - Flow chart . . . . .          | D7 |
| D.5  | imfillcontour.m - Flow chart . . . . .        | D8 |

|      |                                             |     |
|------|---------------------------------------------|-----|
| D.6  | imgenerategrid.m - Flow chart . . . . .     | D9  |
| D.7  | imgrid.m - Flow chart . . . . .             | D10 |
| D.8  | imquotient.m - Flow chart . . . . .         | D11 |
| D.9  | imselect.m - Flow chart . . . . .           | D12 |
| D.10 | imalignsub_ga.m - Flow chart . . . . .      | D14 |
| D.11 | objective_function.m - Flow chart . . . . . | D16 |
| D.12 | transform_inputs.m - Flow chart . . . . .   | D18 |
| D.13 | classify_inputs.m - Flow chart . . . . .    | D20 |
| D.14 | writcase.m - Flow chart . . . . .           | D21 |

# List of Tables

|     |                                                            |    |
|-----|------------------------------------------------------------|----|
| 1.1 | Canadian (Wells) clinical prediction rule . . . . .        | 2  |
| 1.2 | Geneva (Wicki) clinical prediction rule . . . . .          | 3  |
| 1.3 | Clinical factors used in PEscore regression . . . . .      | 4  |
| 1.4 | PE score as different cutoff values . . . . .              | 4  |
| 1.5 | PIOPED Criteria . . . . .                                  | 10 |
| 2.1 | Summary of GA optimisation . . . . .                       | 20 |
| 2.2 | Summary of quotient image states . . . . .                 | 21 |
| 2.3 | Anterior PCA Data Reduction . . . . .                      | 25 |
| 2.4 | Posterior PCA Data Reduction . . . . .                     | 25 |
| 2.5 | Left Lateral PCA Data Reduction . . . . .                  | 25 |
| 2.6 | Right Lateral PCA Data Reduction . . . . .                 | 26 |
| 2.7 | Left Posterior Oblique PCA Data Reduction . . . . .        | 26 |
| 2.8 | Right Posterior Oblique PCA Data Reduction . . . . .       | 26 |
| 2.9 | Summary of Hybrid Monte-Carlo network parameters . . . . . | 33 |
| 4.1 | Example Confusion Matrix . . . . .                         | 36 |

|     |                                                  |    |
|-----|--------------------------------------------------|----|
| 4.2 | Training Results - Image Size: 16x16 . . . . .   | 37 |
| 4.3 | Training Results - Image Size: 32x32 . . . . .   | 38 |
| 4.4 | Training Results - Image Size: 64x64 . . . . .   | 39 |
| 4.5 | Validation Results - Image Size: 16x16 . . . . . | 40 |
| 4.6 | Validation Results - Image Size: 32x32 . . . . . | 41 |
| 4.7 | Validation Results - Image Size: 64x64 . . . . . | 42 |
| 4.8 | Sensitivity, Specificity, PPV and NPV . . . . .  | 47 |

# Nomenclature

|             |                                     |
|-------------|-------------------------------------|
| <b>ANT</b>  | Anterior                            |
| <b>AUC</b>  | Area Under Curve                    |
| <b>BMLP</b> | Bayesian Multi Layer Perceptron     |
| <b>CI</b>   | Confidence Interval                 |
| <b>CT</b>   | Computed Tomography                 |
| <b>CXR</b>  | Chest X-ray                         |
| <b>DTPA</b> | Diethylenetriamine-pentaacetic Acid |
| <b>DVT</b>  | Deep Vein Thrombosis                |
| <b>ECG</b>  | Electrocardiogram                   |
| <b>FSHS</b> | Full Scale Histogram Stretch        |
| <b>GA</b>   | Genetic Algorithm                   |
| <b>HMC</b>  | Hybrid Monte-Carlo                  |
| <b>LLAT</b> | Left Lateral                        |
| <b>LPO</b>  | Left Posterior Oblique              |
| <b>MAA</b>  | Macroaggregated Albumin             |
| <b>MLP</b>  | Multi Layer Perceptron              |
| <b>PCA</b>  | Principal Component Analysis        |
| <b>PE</b>   | Pulmonary Embolism                  |
| <b>PNG</b>  | Portable Network Graphics           |
| <b>POST</b> | Posterior                           |
| <b>PTP</b>  | Pre-test Probability                |
| <b>RLAT</b> | Right Lateral                       |
| <b>ROC</b>  | Receiver Operating Characteristic   |
| <b>RPO</b>  | Right Posterior Oblique             |
| <b>sCT</b>  | Spiral Computed Tomography          |
| <b>SoF</b>  | Statistical Overlay Function        |
| <b>SPD</b>  | Segmental Perfusion Defect          |
| <b>V/Q</b>  | Ventilation / Perfusion             |
| <b>VR</b>   | Variance Retained                   |



# Chapter 1

## Background

Pulmonary embolism (PE) is a common disease that effects many people throughout the world, and could be life threatening. Pulmonary emboli are caused by blood clots, that usually begin forming in the lower leg. Pieces of the blood clot then break off and travel to the lungs. Once in the lung, the blood clot restricts the flow of blood in a particular region, resulting in reduced transfer of oxygen to the blood. Blood supply to the lung tissue is also reduced which can result in a lung infarction (or dead tissue). A probable region for a recent pulmonary embolism is characterised by the presence of ventilation in the absence of perfusion.

To develop an autonomous diagnostic tool for PE, a clear understanding of how PE is diagnosed and what procedures are used is required. Section 1.1 gives an overview of the tools and information used by nuclear medicine physicians to diagnose PE.

### 1.1 Diagnosis of Pulmonary Embolism

The diagnosis of pulmonary embolism is made difficult due to the non-specific nature of the presenting symptoms, while less the 35% of patients with suspected PE actually have it (Rodger and Wells [2001]) resulting in many patients needlessly using anticoagulants while awaiting confirmation of their status (Hyers et al. [2001]). Cases that are clearly positive or clearly negative for PE are relatively simple to diagnose. The difficulty lies in diagnosing the cases in between these two extremes.

Limitations in the imaging techniques used in the diagnosis of PE necessitate the use of these techniques in conjunction with clinical pre-test probability assessment

(Rodger and Wells [2001]). Section 1.1.1 summarises a few empirical rules for determining a Pre Test Probability (PTP) for PE, while Section 1.1.2 explains the various imaging techniques that can be used for visualising the lungs.

### 1.1.1 Clinical Prediction Rules

Many prediction rules have been developed (Wells et al. [2000], Wicki et al. [2001], Stöllerberger et al. [2000]) to assist in the clinical evaluation of patients with suspected pulmonary embolism. To be able to develop an accurate prediction model, all the potential clinical factors need to be analysed for statistical significance. A summary of the statistical significance of these clinical factors can be reviewed in (Rodger and Wells [2001]). All of the clinical predictions use a points-based system, assigning more points to clinical factors which show greater positive predictive values.

#### Wells et al. (Canadian Score)

The Wells or Canadian Score was devised by Wells (Wells et al. [2000]) and has been shown to be a meaningful (Moore et al. [2004]) and cost effective (Humphreys et al. [2004]) method for determining the clinical probability of PE. The breakdown of the prediction rule can be viewed in Table 1.1.

Table 1.1: Canadian (Wells) clinical prediction rule. (Wells et al. [2000])

| Item                                                                          | Points |
|-------------------------------------------------------------------------------|--------|
| a. Clinical signs and symptoms of deep vein thrombosis (DVT)                  | 3      |
| b. An alternative diagnosis is less likely than PE                            | 3      |
| c. Heart rate greater than 100                                                | 1.5    |
| d. Immobilisation of surgery in the previous 4 weeks                          | 1.5    |
| e. Previous DVT PE                                                            | 1.5    |
| f. Hemoptysis                                                                 | 1      |
| g. Malignancy (on treatment, treated in last 6 months or palliative)          | 1      |
| Low PTP (<2 points), intermediate PTP (2-6 points), and high PTP (> 6 points) |        |

## Wicki et al. (Geneva Score)

The Geneva score prediction rule is shown below in Table 1.2. The Geneva score prediction rule requires values obtained from an arterial blood gas sample, which because they are not always performed, may limit the usefulness of the Geneva score (Moore et al. [2004]).

Table 1.2: Geneva (Wicki) clinical prediction rule. (Wicki et al. [2001])

| Item                                                                            |                               | Points |
|---------------------------------------------------------------------------------|-------------------------------|--------|
| a.                                                                              | Age                           |        |
|                                                                                 | 60-79                         | 1      |
|                                                                                 | $\geq 80$                     | 2      |
| b.                                                                              | Previous PE or DVT            | 2      |
| c.                                                                              | Recent surgery                | 3      |
| d.                                                                              | Pulse rate $>100/\text{min}$  | 1      |
| e.                                                                              | PaCO <sub>2</sub>             |        |
|                                                                                 | $<4.9\text{kPa}$ (36mm Hg)    | 2      |
|                                                                                 | 4.8-5.19kPa (36-38.9mm Hg)    | 1      |
| f.                                                                              | PaO <sub>2</sub>              |        |
|                                                                                 | $<6.5$ (48.75 mm Hg)          | 4      |
|                                                                                 | 6.5-7.99 (48.75-59.9 mm Hg)   | 3      |
|                                                                                 | 8-9.49 (60-71.18mm Hg)        | 2      |
|                                                                                 | 9.5-10.99 (71.25-82.42 mm Hg) | 1      |
| g.                                                                              | CXR                           |        |
|                                                                                 | Plate-like atelectasis        | 1      |
|                                                                                 | Elevation of hemidiaphragm    | 1      |
| Low PTP (0-4 points), intermediate PTP (5-8 points), and high PTP (9-16 points) |                               |        |

## PEScore

The PEScore algorithm described in (Stöllberger et al. [2000]) is based on a multiple regression analysis of a several clinical factors described in Table 1.3. A condensed version of results for the PEScore prediction rule is presented in Table 1.4.

The analysis in Table 1.4 indicates that as the PEScore increases, specificity and the probability of pulmonary embolism (positive predictive value) increase while the

Table 1.3: Clinical factors used in PEscore regression (Stöllerberger et al. [2000])

| Clinical Factor                     | Coefficient |
|-------------------------------------|-------------|
| Proven leg thrombosis (0=no;1=yes)  | 0.29        |
| ECG right heart strain (0=no;1=yes) | 0.25        |
| Neck vein distention (0=no;1=yes)   | 0.22        |
| Dyspnoea (0=no;1=yes)               | 0.20        |
| Suspicious chest X-ray (0=no;1=yes) | 0.13        |
| Constant                            | -0.17       |

Table 1.4: PE score at different cutoff values: Sensitivity, specificity, and positive and negative predictive values for detecting pulmonary embolism. (Stöllerberger et al. [2000])

| PEscore | Sensitivity (%) | Specificity (%) | Positive Predictive value (%) | Negative Predictive value (%) |
|---------|-----------------|-----------------|-------------------------------|-------------------------------|
| >0.02   | 100             | 5               | 35                            | 100                           |
| >0.10   | 100             | 26              | 26                            | 100                           |
| >0.20   | 100             | 64              | 59                            | 100                           |
| >0.30   | 100             | 79              | 71                            | 100                           |
| >0.40   | 94              | 86              | 77                            | 96                            |
| >0.50   | 70              | 99              | 97                            | 87                            |
| >0.60   | 60              | 99              | 97                            | 83                            |
| >0.65   | 55              | 100             | 100                           | 81                            |

sensitivity to detect pulmonary embolism decreases.

A review of the definitions for Sensitivity, Specificity, Positive Predictive Value and Negative Predictive Value can be found in Appendix C.

### 1.1.2 Imaging Techniques

There exist various medical imaging techniques available for diagnosing suspected PE. A more thorough explanation of each technique can be found in (Rodger and Wells [2001]). As mentioned earlier, V/Q scanning is used as the imaging technique for this research.

## V/Q Lung Scanning

For a long time V/Q scanning has been the imaging protocol of choice for the evaluation of patients with suspected PE. The procedure has two parts.<sup>1</sup> One example of such a procedure is as follows:

**Perfusion:** The patient is injected with 5mCi Technetium-99m MAA (macroaggregated albumin). This allows the nuclear medicine physician to visualise the flow of blood in the patients lungs.

**Ventilation:** The patient is asked to inhale 20mCi of Technetium-99m DTPA (diethylenetriamine-pentaacetic acid) labelled aerosol which indicates airflow in the patients lungs. In some instances the patient may swallow the gas instead of inhaling it. Due to the deposition of the aerosol in the stomach and throat, artifacts in the image appear which need to be removed before diagnosis. This concept is discussed further in Section 2.5.1.

**Image acquisition** is achieved using a  $\gamma$ -camera. In both the ventilation and perfusion, six projections of the patients lungs are captured. These projections are labelled as follows:

- Anterior (ANT)
- Posterior (POST)
- Left lateral (LLAT)
- Right lateral (RLAT)
- Left posterior oblique (LPO)
- Right posterior oblique (RPO)

V/Q scanning has the advantages that it is a non-invasive and inexpensive test. A near-normal lung scan essentially excludes PE and a high probability scan has an

---

<sup>1</sup>The explanation of the Perfusion and Ventilation tests relate to the procedures used in this research. Other radiopharmaceuticals such as Krypton-81m, Technegas and Xe-133 could also be used.(Hustinx [2002])

80-85% positive predictive value (Rodger and Wells [2001]). However, the main disadvantage is that V/Q scans often fall into the non-diagnostic category (Stöllberger et al. [2000]), so further tests are required to confirm the presence of PE.

### **Pulmonary Angiography**

Pulmonary Angiography is almost always used as the *Gold Standard* test for PE. Using a contrast dye which is injected into the patient, a nuclear medicine physician is then able to visualise the blood flow in the lungs, however many clinicians choose not to use it for a number of reasons (Rodger and Wells [2001]):

1. Fear of mortality related to pulmonary angiography
2. Limited availability in small centers
3. Expensive and requires expertise

These limitations often lead patients to being improperly managed (Schluger et al. [1994]). Furthermore, pulmonary angiogram tests are not perfect, and a patient with a normal pulmonary angiogram could expect a 2.2% (95% CI of 3.8-8.0%) venous thromboembolic event rate at 1-year followup (Hull et al. [1983]).

### **Venous Ultrasound Imaging Studies of the Legs**

The greatest utility for venous ultrasound is in those patients with a high pre-test probability for PE, and those showing signs and symptoms of deep vein thrombosis (DVT). In these groups, ultrasound will be positive in 46% and 15%, respectively (Wells et al. [1997, 1998]).

### **D-dimer for Diagnosis of Venous Thromboembolism**

Plasma D-dimer is produced during the degrading of crosslinked fibrin and has been under intense investigation in recent years. It has been shown to be highly sensitive to acute pulmonary embolism at a threshold value of  $500\mu\text{g.l}^{-1}$  (Bounameaux et al. [1997]). What this tells us, is that any D-dimer level below this value reasonably

rules out PE. The problem with D-dimer fibrin is that it has a very low positive predictive value and can therefore not reliably diagnose positive cases of PE.

### **Spiral CT Angiography**

Recent technical advances in CT have sparked great interest in the use of spiral CT (sCT) for the diagnosis of PE. Unlike conventional V/Q scanning, sCT imaging enables direct visualisation of pulmonary emboli within the pulmonary arteries (Remy-Jardin et al. [1992], Teigen et al. [1993]).

In almost all circumstances, sCT is performed as a single contrast series through the thorax. A more comprehensive description of the imaging protocol can be found in Bounameaux et al. [1997].

Initial studies of the diagnostic accuracy of sCT in the diagnosis of PE report sensitivities and specificities approaching 100%, however new studies have been conducted resulting in an extension of the sensitivity to between 53% and 89%, while the specificity ranges between 78% to 100%. (Rossum et al. [1996], Remy-Jardin et al. [1996], Mayo et al. [1997], Drucker et al. [1998], Garg et al. [1998]).

#### **1.1.3 The PIOPED Investigation**

The PIOPED, and following that the PISA-PED investigations have been the two largest studies of pulmonary embolism. Both studies used prospective data to determine a set of empirical rules which could be applied to V/Q scans to determine the probability of PE.

The following paragraph, quoted directly from the abstract of the PIOPED investigation, explains the findings, (PIOPED Investigators [1990]).

"To determine the sensitivities and specificities of ventilation/perfusion lung scans for acute pulmonary embolism, a random sample of 933 of 1493 patients was studied prospectively. Nine hundred thirty-one underwent scintigraphy and 755 underwent pulmonary angiography; 251 (33%) of 755 demonstrated pulmonary embolism. Almost all patients with pulmonary embolism had abnormal scans of high, intermediate, or low probability, but so did most without pulmonary embolism (sensitivity, 98%; specificity, 10%). Of 116 patients with high-probability scans and definitive angiograms, 102 (88%) had pulmonary embolism, but only a minority with pulmonary embolism had high-probability scans (sensitivity, 41%; specificity, 97%). Of 322 with intermediate-probability scans and definitive angiograms, 105 (33%) had pulmonary embolism. Follow-up and angiography together suggest pulmonary embolism occurred among 12% of patients with low-probability scans. Clinical assessment combined with the ventilation/perfusion scan established the diagnosis or exclusion of pulmonary embolism only for a minority of patients—those with clear and concordant clinical and ventilation/perfusion scan findings".

## **The PIOPED Criteria**

The PIOPED criteria shown in Table 1.5 gives a summary of the criteria defined by the PIOPED investigators for determining a patient's probability of pulmonary embolism.

### **1.2 Overview of Research Report**

**Chapter 1** Discusses the various ways in which a patient's probability of pulmonary embolism can be established.

**Chapter 2** Describes the implementation of the system from image segmentation through to the classification of the inputs.

**Chapter 3** Contains functional breakdowns of the MATLAB© functions that are used to implement the system.

**Chapter 4** Contains the results of the system followed by a discussion on the findings and possible improvements.

**Appendix A** Provides the structures of the data in MATLAB©



**Appendix B** User manual for the Medical Image Viewer software package. Developed to further future research by using inputs from other nuclear medicine physicians.

**Appendix C** Review of definitions for sensitivity, specificity, positive and negative predictive value.

**Appendix D** Contains PEDiag flow charts.

### 1.3 Key Achievements of the Study

Research into the current methods of PE diagnosis was conducted. Areas of image segmentation were also studied. Following this research a system was developed in MATLAB© to provide a platform for the automated diagnosis of pulmonary embolism using V/Q lung scintigraphy. These methods include:

- Image Segmentation
- Image Alignment
- Feature Extraction
- Feature Classification

The collected data was processed in MATLAB©, The extracted features were then classified using a Bayesian Neural Network. The results of the system can be found in Chapter 4.

In summary, a system has been developed for the automated diagnosis of PE. The system shows excellent training performance and reasonable validation performance.

Table 1.5: PIOPED Criteria

| Probability  | Criteria                                                                                                                                                                                                                |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| High         | <ul style="list-style-type: none"> <li>• <math>\geq 2</math> large segmental perfusion defects (SPD)</li> <li>• 1 large SPD and <math>\geq 2</math> moderate SPD</li> <li>• <math>\geq 4</math> moderate SPD</li> </ul> |
| Intermediate | <ul style="list-style-type: none"> <li>• 1 moderate SPD</li> <li>• Corresponding V/Q defect and chest x-ray (CXR) opacity in lower lung</li> <li>• Single moderately matched V/Q defect</li> </ul>                      |
| Low          | <ul style="list-style-type: none"> <li>• Multiple matching V/Q defects</li> <li>• Corresponding V/Q defects and CXR parenchymal opacity in upper or middle lung</li> <li>• <math>&gt; 3</math> small SPD</li> </ul>     |
| Very Low     | <ul style="list-style-type: none"> <li>• <math>\leq 3</math> small SPD</li> </ul>                                                                                                                                       |
| Normal       | <ul style="list-style-type: none"> <li>• No perfusion defects and perfusion outlines the shape of the lung seen on CXR</li> </ul>                                                                                       |

---

## Chapter 2

# Implementation

### 2.1 The Role of Automated Diagnosis

The accurate diagnosis of PE is a challenging problem, but one which doctors make on a daily basis. The role of an automated system should at present be thought of as a decision support system, and not as replacement to an experienced doctor. The intrinsic characteristics of an automated tool would allow for a decrease in the inter-observer and intra-observer variability.

Automatic systems generally offer an increase in diagnostic speed when compared to manual and semi-automated diagnosis methods. This would reduce the time spent by patients in hospital, leading to reduced medical bills, while the hospital would be able to diagnose more patients in the same time period.

These benefits can only be achieved if the system is reliable, accurate and cost-effective in terms of implementation and maintenance.

### 2.2 Previous Works

According to Ericsson et al. [2003], computer-aided diagnosis was first presented by Tourassi et al. [1993] and Scott and Palmer [1993]. Both of these implementations made use of artificial neural networks to classify manually obtained features or inputs. The following properties define the two categories of systems that have been developed: those that use manually obtained inputs which are then given to the

classifier, and those systems that are completely autonomous.

Since that initial work by Tourassi and Scott, there have been a number of research efforts focusing on the automatic diagnosis of PE from V/Q scans, and various methods have been employed to achieve this. A technique presented in Serpen et al. [2003] used principal component analysis (PCA) (discussed further in Section 2.5.4) to match input lungs to templates. The templates therefore need to be representative of each possible outcome. The disadvantage of this method is using a small finite number of templates to represent the much larger set of anatomical variations present in human lungs. Interesting work is presented by Ericsson et al. [2003] using image profiles (the sum of an image in a specific direction, horizontal and vertical) for segmentation and Support Vector Machines (SVM) for classification. Fuzzy inference systems have also been developed by Serpen et al, and more information can be found in Dagli et al. [2000]. A novel method called *SymText* that uses text processing rather than image processing has been described by Fiszman et al. [1998]. This system extracts important keywords from the nuclear medicine physicians reports in order to determine a patients risk of PE.

Automated diagnosis has also been applied to the automatic diagnosis of bone metastases (Sadik et al. [2006]), myocardial infarction (Lindahl et al. [1999]) and cervical cancer (Mango [1994]).

## 2.3 Data Collection

Retrospective cases from Johannesburg General Hospital, Chris Hani Baragwanath Hospital and the Donald Gordon Medical Center archives were used. All of these institutions are located in the Johannesburg area in the Republic of South Africa. Ethics approval to use the studies was obtained from the Human Research Ethics Committee (Medical)<sup>1</sup>, protocol number M040506.

Each case consisted of 12 images, six for ventilation and six for perfusion which represent the views discussed earlier in Section 1.1.2. All images were exported from APEX View in PCX file format. The images were then imported into MATLAB® and stored in a single structure, the format of which can be found in Appendix A.

---

<sup>1</sup>Secretariat: Research Office, Room SH10005, 10th floor, Senate House, University of the Witwatersrand, Johannesburg. Private Bag 3, Wits 2050, South Africa.

The V/Q scans were obtained using the following protocol:

**Ventilation** 20mCi Tc-99m DTPA (diethylenetriamine-pentaacetic acid)

**Perfusion** 5mCi Tc-99m MAA (macroaggregated albumin)

Initially, clinical data was to be collected for each patient in order to determine a pre-test probability for PE, however due to the inconsistency in the available data on each patient, it was decided that clinical data would not be used in this research. An overview of the type of clinical data that can be used is presented in Section 1.1.1.

In total, 179 cases were collected and studied retrospectively.

## 2.4 The Gold Standard

In order for the results of this research to be quantifiable, a point of reference or *Gold Standard* is required. Usually pulmonary angiography is used as the *Gold Standard* for the diagnosis of PE, however, it is not used in this research as all the cases are retrospective and pulmonary angiography was either not performed or could not be performed in each of the cases.

The *Gold Standard* in this research was taken from the clinical reports of an expert nuclear medicine physician from the aforementioned hospitals. The diagnosis was either negative, intermediate probability or high probability of pulmonary embolism.

## 2.5 System Block Diagram

Figure 2.1 shows the implementation of the system. The V/Q Imaging Data has already been discussed in Sections 1.1.2 and 2.3.

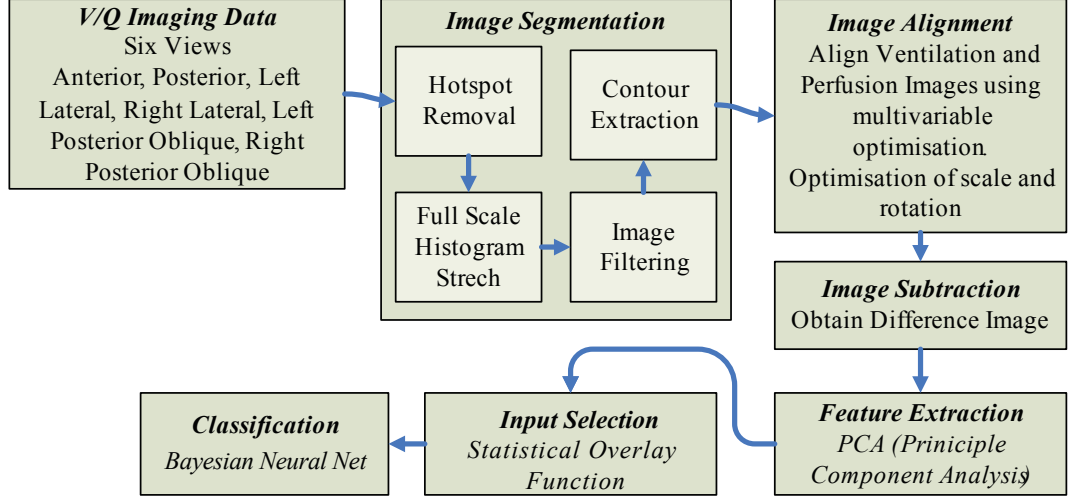


Figure 2.1: System block diagram

### 2.5.1 Image Segmentation

Image segmentation is the first, and most critical step in the entire system. If the images are not segmented correctly, vital information may be lost or redundant information may be retained. The following sections describe the image segmentation process in detail.

#### Full Scale Histogram Stretch

The Full Scale Histogram Stretch (FSHS) (Gonzalez and Wintz [1987]) is a technique which greatly increases the contrast in the image and allows for more accurate contour detection. A FSHS is applied to each image. The equation for the FSHS is given below in Equation 2.1.

$$V_o(x, y) = V_B(x, y) \times \left( \frac{K - 1}{B - A} \right) \quad (2.1)$$

where:

$V_o(x, y)$  = Output Image [256x256]

$V_B(x, y)$  = Input Image[256x256]

$K$  = Max Allowable Intensity

$B$  =  $\max\{V_B(x, y)\}$

$$A = \min\{V_B(x, y)\}$$

In this implementation K is set to 256. This value is chosen as it is the maximum intensity an 8-bit image ( $2^8 = 256$ ) can be.

Figure 2.2 shows the effects of performing a FSHS on an anterior lung scan. The histogram in the top left is typical of an image exhibiting low contrast. This is confirmed in the image below in which the lung is barely visible. The histogram on the right shows much better contrast, which can be seen in the lung image after the FSHS.

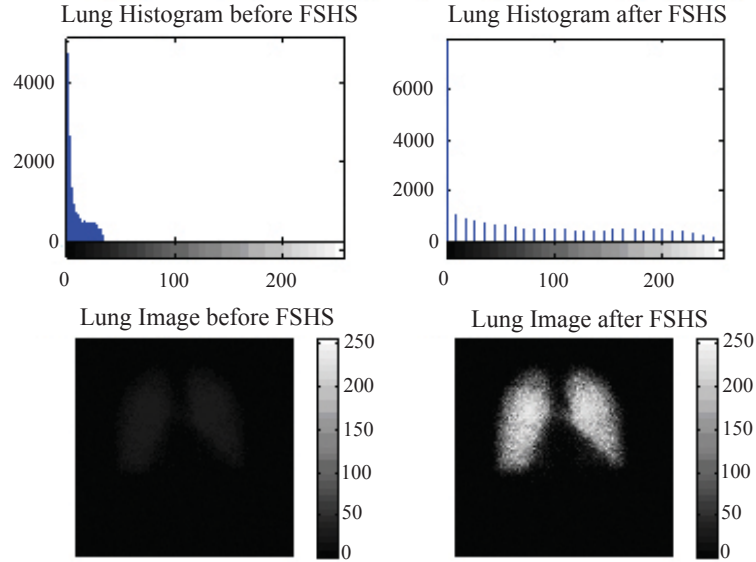


Figure 2.2: Full Scale Histogram Stretch. The left images represent the images before the FSHS is performed, notice the poor contrast in the histogram and the lungs are hardly visible. The right images represent the images after a FSHS is performed showing an increase in contrast

## Hotspot Removal

Hot spots are generally caused by the radiopharmaceutical getting trapped in a small area of the lung due to obstructive lung disease or the poor technical quality (Frigyesi [2003]) of the radio nucleotides. Hot spot removal provides a reduction in areas of high intensity relative to the surrounding area in the image. The equation for determining a hotspot is given in 2.2 and is taken from Frigyesi [2003]. Generally, hotspot removal is only applied to ventilation images.

$$HS = (x, y) \in C_v; \frac{V(x, y) - \hat{V}}{\hat{s}} \geq q \quad (2.2)$$

Where  $\hat{V}$  and  $\hat{s}$  are made the following:

$$\hat{V} = \text{median}\{V(x, y); (x, y) \in C_v\} \quad (2.3)$$

$$\hat{s} = \text{median}\{|V(x, y) - \hat{V}|; (x, y) \in C_v\} \quad (2.4)$$

and  $q = 6$ . This value was taken from Frigyesi [2003].

Figure 2.3 shows a right posterior oblique lung scan before and after hotspot removal has taken place. The hotspot is removed by interpolating the surrounding pixels from the outside in.

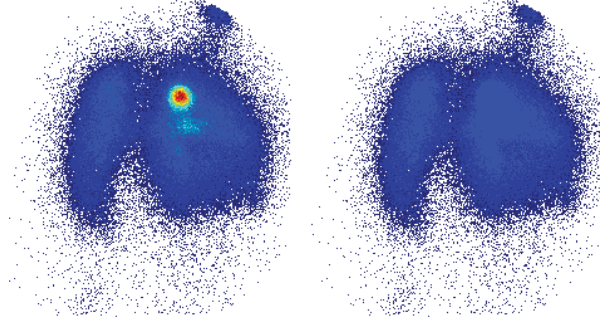


Figure 2.3: Image before (left) and after (right) hotspot removal

### Image Filtering and Contour Extraction

The image is then filtered using a  $[4 \times 4]$  averaging matrix to reduce noise in the image and provide smoother contours. Finally, iso-contours are extracted and the lung is segmented from the background. An example of an anterior lung image with overlaid iso-contours can be seen in Figure 2.4.



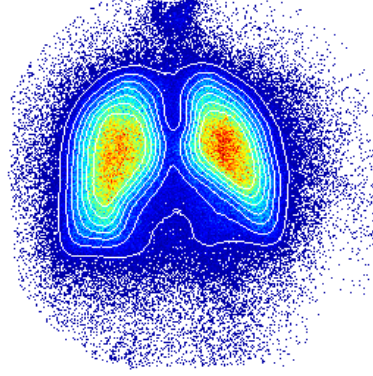


Figure 2.4: Anterior lung with iso-contours overlaid

### Dethroating and Destomaching

Dethroating (Frigyesi [2003]) involves the removal of throat artifacts from the ventilation images. Throat artifacts, caused by the radio pharmaceutical getting trapped in the patients trachea, leads to areas of high image intensity. Stomach artifacts are caused by the patient swallowing some radio pharmaceutical instead of inhaling it. The radio pharmaceutical then settles in the stomach and shows up on the image. These artifacts must be removed for the image alignment and classification to be most effective.

For the purposes of this study, these algorithms where developed based on a image segmentation technique described in Ericsson et al. [2003]. The procedure for dethroating and destomaching are very similar, however the destomaching procedure requires the image to be rotated so that the stomach artifact is pointing vertically down, as this is not always the case. First an image profile in the horizontal direction is obtained, which is then filtered using a  $2^{nd}$  order Savitzky-Golay filter with a frame size of 25. The peak value of the profile is then found. Using this peak value, the positive and negative gradient of the profile are approximated using a simple straight line of the form  $y = mx + c$ . The x-intercepts (zero-crossings) of the straight lines are found and these values are then used to determine which areas of the image must be removed. The entire procedure is show in Figure 2.5, the two zero-crossings are shown along with the straight line approximations (red dashed lines). The areas shaded in red indicate which areas are to be removed from the image.

Figure 2.6 show the lung images before (top two rows) and after segmentation (bottom two rows) respectively. The views, from left to right, are Anterior, Posterior, Left Lateral, Right Lateral, Left Posterior Oblique and Right Posterior Oblique.

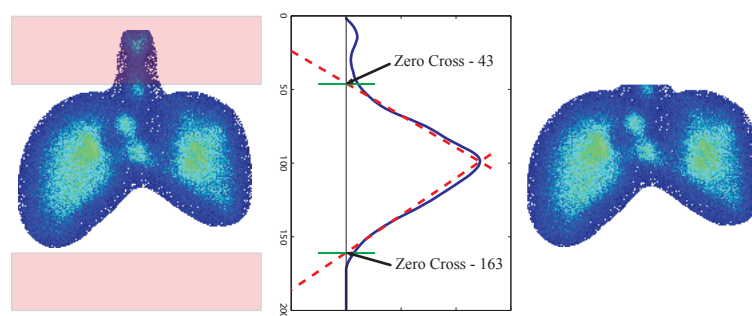


Figure 2.5: Image dethroating. The left hand image shows to lungs before dethroating. The right image shows the lungs after a dethroating is performed.

The image segmentation is able to remove unwanted information, while still retaining vital parts of the image which are required for reliable classification.

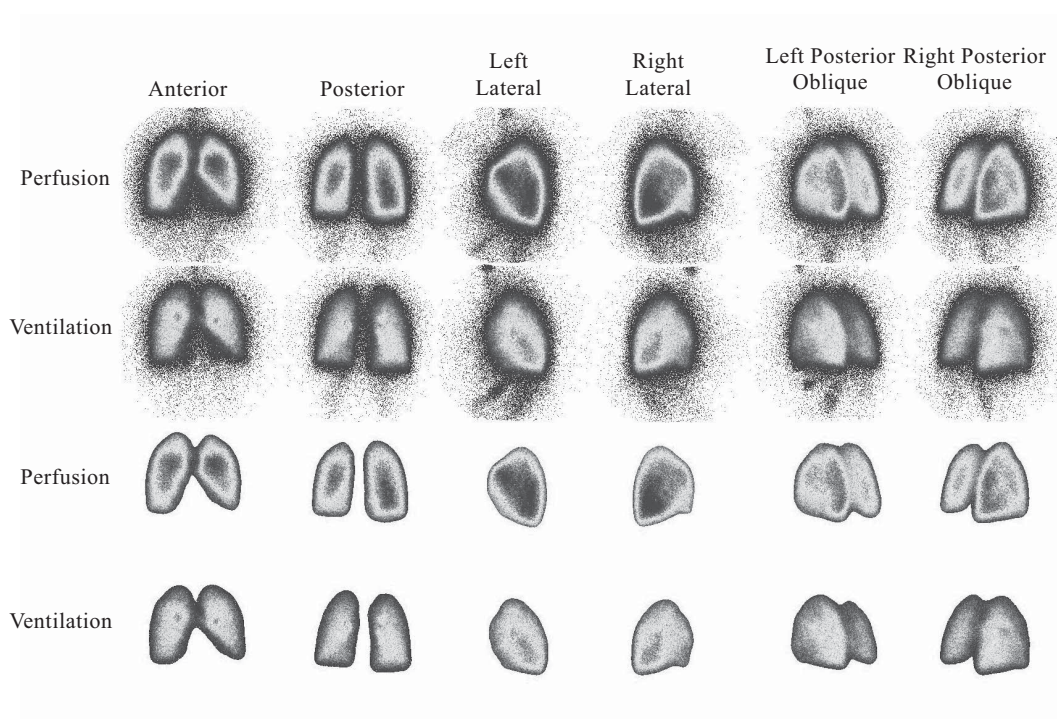


Figure 2.6: Images before (top two rows) and after (bottom two rows) segmentation

### 2.5.2 Image Alignment

As pulmonary embolism is identified by matched defects in perfusion and ventilation images, each V/Q pair (there are six pairs) must be aligned before subtraction. More specifically areas where ventilation is present and perfusion absent are regarded as probable pulmonary emboli. To reduce the effects of lung defects on the alignment,

the segmented images are converted to binary images, formally stated in Equation (2.5).

$$V_B(x, y) = \begin{cases} 1 & \text{if } V_B(x, y) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

$V_B(x, y)$  is the intensity value of the image at position  $(x, y)$ . The alignment is accomplished using multi-variable (scale, rotation, x-translation, y-translation) optimization algorithms. A disadvantage of the genetic algorithm is that it is computationally expensive and it takes longer to find the global minimum compared to the simplex method. However genetic algorithms are very well suited to multivariate optimisation due to the encoding of all variables onto a single chromosome. The principle of genetic algorithms is explained below. Another method presented in Ericsson et al. [2003] makes use of image profiles and 2D convolution for segmentation and alignment.

## Genetic Algorithms

The genetic algorithm is part of a group known as the *Population-Based optimisation methods* and is based on the principles of reproduction, natural selection and naturally occurring genetic operations including those of crossover and mutation Koza [1997]. The GA algorithm initially creates a population of chromosomes of specified length. The size of the population is determined by the user, which was chosen to be 1000. The initial estimates for the optimisation parameters are encoded with specified precision onto the chromosome. Each chromosome's fit is evaluated using a cost function, which in this case is a subtraction function given in Equations 2.6 and 2.7.

$$D(x, y) = \sum_{x=1}^{256} \sum_{y=1}^{256} |V(x, y) - P(x, y)| \quad (2.6)$$

where  $V(x, y)$  and  $P(x, y)$  are the intensities at position  $(x, y)$  of the ventilation and perfusion images respectively.

$$err = \left( \sum_{x=1}^{256} \sum_{y=1}^{256} D(x, y) \times 100 \right)^2 \quad (2.7)$$

Those chromosomes with the best fit are used to breed the next generation of chromosomes, while also undergoing processes of crossover and mutation. This process continues for as many generations as required until the maximum number of generations has been exceeded or the minimum error has been attained.

Figure 2.7 shows the results of an image alignment algorithm using the GA with Schepp-Logan digital phantom images. The reference image represents the image that needs to be optimised, in other words the origin of the optimisation problem. The target image represents the destination of the optimisation. The 4 parameters, namely scale, rotation, x-translation and y-translation provide a transformation between the reference image and the target image. The transformation image represents the reference image, after it has been transformed with the optimised parameters. Table 2.1 shows a summary of the parameters found using the GA.



Figure 2.7: Result of image alignment using Schepp-Logan digital phantoms

Table 2.1: Summary of GA optimisation

| Parameter     | Actual       | Found         | Error (%) |
|---------------|--------------|---------------|-----------|
| Scale         | 0.90         | 0.88          | 2.3       |
| Rotation      | $-6.5^\circ$ | $-7.15^\circ$ | 9.1       |
| X-translation | 25           | 27            | 7.4       |
| Y-translation | 15           | 16            | 6.25      |

### 2.5.3 Image Subtraction

After the images are all aligned using the genetic algorithm, the ventilation and perfusion images are subtracted. The algorithm subtracts the ventilation image from the perfusion image, therefore areas with intensity values less than 0 indicate that there is more ventilation than perfusion in that specific area. The severity of the defect can then be quantified by taking a magnitude of pixel intensity in the subtraction image.

### Quotient Images

An alternative method, developed for this study, to image subtraction is that of obtaining quotient images. These quotient images are obtained by dividing the ventilation image by the perfusion image resulting in an image which represents four different states. This is the main advantage of quotient images over subtraction images as subtraction images only represents a single state. The different states that are represented using quotient images are summarised in Table 2.2.

Table 2.2: Summary of quotient image states

|                                                                      | Ventilation | Perfusion | Quotient = $\frac{vent}{perf}$ | State Relevance                                  |
|----------------------------------------------------------------------|-------------|-----------|--------------------------------|--------------------------------------------------|
| I                                                                    | vent > 0    | perf > 0  | Ratio                          | Relevant depending on ratio                      |
| II                                                                   | vent > 0    | perf = 0  | $\infty$                       | The shows probable locations of pulmonary emboli |
| III                                                                  | vent = 0    | perf > 0  | 0                              | Not relevant for diagnosis of PE                 |
| IV                                                                   | vent = 0    | perf = 0  | NaN                            | Not relevant for diagnosis of PE                 |
| <b>Key:</b> perf - perfusion, vent - ventilation, NaN - Not a Number |             |           |                                |                                                  |

As can be seen in Table 2.2 the first state contains the same information that is obtained using image subtraction. If the ratio is quite high, it could be assumed that there are perfusion defects in that specific area of the lung. The second state, which is equally important in the determination of pulmonary embolism shows areas of ventilation without associated perfusion.

#### 2.5.4 Feature Extraction

In order for the neural network to classify input data correctly, features need to be extracted from the lung images. Feature extraction helps in the reduction of the input data space as well as removing data that are redundant in the classification process. There are many different feature extraction methods, and a few different ones have been tested in this research. The first method discussed is that of Fourier Coefficients.

##### Polar Co-ordinates and Fourier Coefficients

One method for feature extraction is to convert the desired areas of the lung into polar co-ordinates. A polar co-ordinate representation of a typical lung is shown in Figure 2.8.

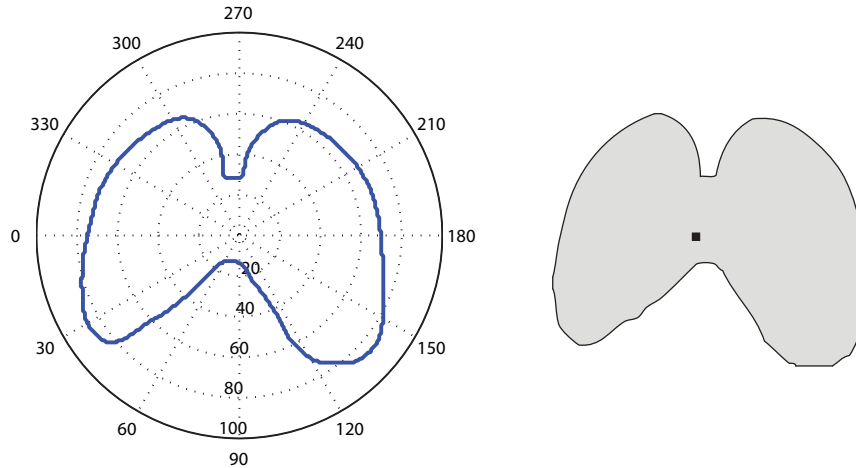


Figure 2.8: The left image shows the anterior view of lung represented in polar co-ordinates. The right image shows the anterior view, the black square represents the centroid of the image.

Polar co-ordinates need to be calculated from a reference point, which in this case is the centroid of the lung image, which is indicated by the black square in Figure 2.8. The conversion from cartesian to polar co-ordinates is taken on the outer contour of the selected area(s) of the lung. After the conversion, the radius vector is sorted according to increasing values of  $\theta$ . In order to make the defects independent of their position in the image, the Fourier coefficients of the signal are calculated. The Fourier coefficients indicate the strength of the fundamental as well as the harmonic frequencies contained in the signal, which was generated using the contour of the lung. Therefore any sudden changes in the contour of the lung would show up in

the higher frequency Fourier coefficients.

## Principal Component Analysis

Principal component analysis (PCA) (Jolliffe [2002]) is used to extract only the vital information from the image which reduces the number of inputs to the classification system. This allows for faster classification and training times. A summary of the concepts of PCA is given in the next paragraph, while more comprehensive coverage can be found in Jolliffe [2002].

The use of PCA allows the number of variables in a multivariate data set to be reduced, while retaining as much of the variation present in the data set. This reduction is achieved by taking  $n$  variables  $x_1, x_2, x_3, \dots, x_n$  and finding the combinations of these to produce principal components (PCs) which are uncorrelated. These PCs are also termed eigenvectors. The fact that the individual PCs are uncorrelated is an important and useful property as it shows that each PC is describing different "dimensions" in the data. When using PCA, the goal is to choose only the eigenvectors needed to describe almost all (say 90%) of the variance in the data. Hopefully the number of eigenvectors chosen is much less than the total number of eigenvectors calculated. Accordingly, some degree of data reduction is accomplished as the variation in the original number of variables can be described using a smaller number of new variables (PCs).

PCA requires a substantial amount of memory in order to compute the Hessian matrix, therefore instead of calculating one set of transformations for all six images, which would require 4.5GB of physical memory using 64x64 images, six separate transformations were calculated, one for each of the views described earlier.

PCA was performed on both quotient and subtraction images of varying sizes, from 16x16 to 64x64. Figure 2.9 shows a graph reflecting variance retained (VR) to the number of eigenvectors required. Notice that as the image size gets smaller, for the same VR, the number of required eigenvectors decreases. Conversely, for the same number of eigenvectors, the VR increases by approximately 10% for every half reduction in image size. This trend is most likely caused by a certain amount of variance being lost when reducing the image size.

Tables 2.3 through 2.8 give a summary of the PCA reduction for both quotient and

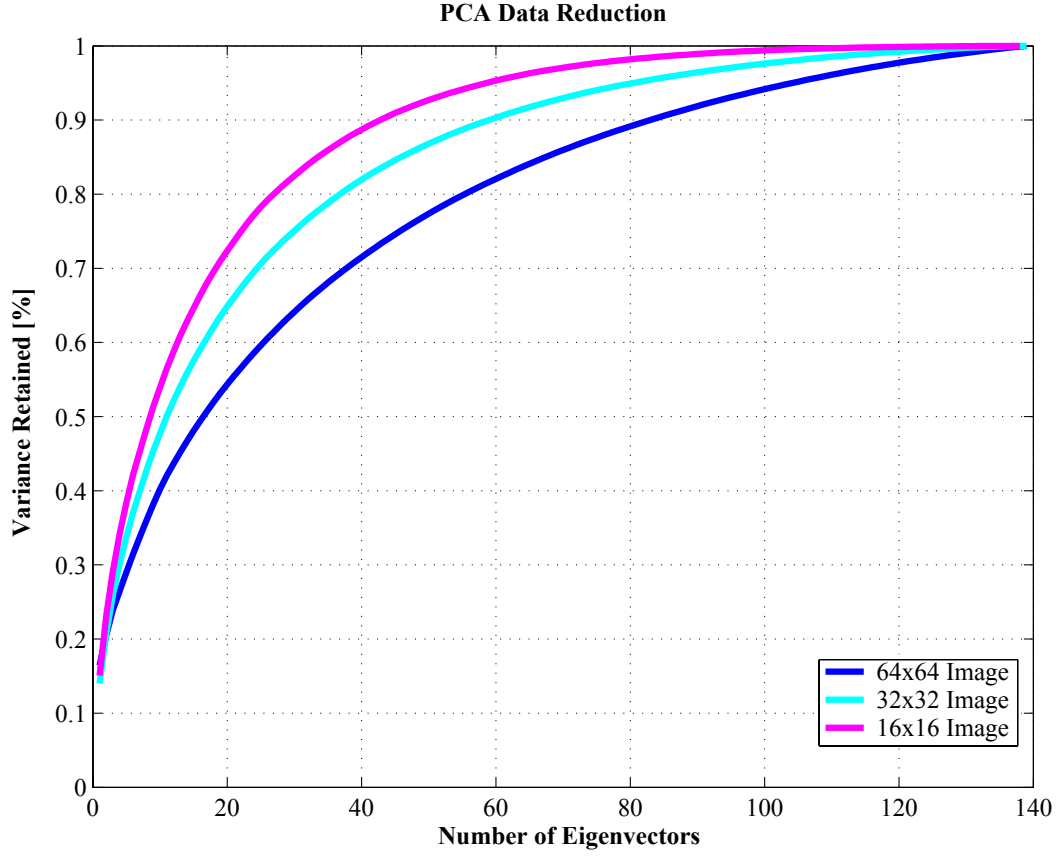


Figure 2.9: PCA data reduction for varying sizes of input images. All graphs are from an anterior quotient image.

subtraction images. The numbers in the tables represent the number of eigenvectors required to retain the specified variance. From the results it can be seen that subtraction images require fewer eigenvectors than quotient images for the same VR. This is more evident when the VR is low.

The question which needs to be asked is should subtraction or quotient images be used as inputs to the classifier? On the one hand, using subtraction images would require fewer inputs, however there is less variance in the dataset, which could imply that it would be more difficult for the classifier to distinguish between the different cases. Quotient images on the other hand, show more variance, but would require more inputs to the classifier.



Table 2.3: Anterior PCA Data Reduction

| Image Size | Type        | Variance Retained |      |      |      |      |      |
|------------|-------------|-------------------|------|------|------|------|------|
|            |             | 0.70              | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| 16x16      | Quotient    | 19                | 23   | 27   | 34   | 43   | 59   |
|            | Subtraction | 10                | 12   | 15   | 19   | 26   | 41   |
| 32x32      | Quotient    | 25                | 30   | 37   | 46   | 59   | 81   |
|            | Subtraction | 12                | 15   | 20   | 26   | 38   | 61   |
| 64x64      | Quotient    | 30                | 37   | 46   | 57   | 72   | 95   |
|            | Subtraction | 16                | 21   | 28   | 40   | 57   | 86   |

Table 2.4: Posterior PCA Data Reduction

| Image Size | Type        | Variance Retained |      |      |      |      |      |
|------------|-------------|-------------------|------|------|------|------|------|
|            |             | 0.70              | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| 16x16      | Quotient    | 19                | 23   | 28   | 34   | 43   | 59   |
|            | Subtraction | 10                | 12   | 15   | 20   | 27   | 42   |
| 32x32      | Quotient    | 26                | 32   | 38   | 48   | 60   | 81   |
|            | Subtraction | 12                | 16   | 21   | 28   | 40   | 63   |
| 64x64      | Quotient    | 33                | 39   | 48   | 59   | 74   | 96   |
|            | Subtraction | 17                | 23   | 31   | 43   | 61   | 90   |

Table 2.5: Left Lateral PCA Data Reduction

| Image Size | Type        | Variance Retained |      |      |      |      |      |
|------------|-------------|-------------------|------|------|------|------|------|
|            |             | 0.70              | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| 16x16      | Quotient    | 20                | 23   | 28   | 35   | 44   | 60   |
|            | Subtraction | 9                 | 11   | 14   | 18   | 25   | 39   |
| 32x32      | Quotient    | 26                | 31   | 38   | 47   | 60   | 81   |
|            | Subtraction | 11                | 15   | 19   | 26   | 37   | 60   |
| 64x64      | Quotient    | 33                | 39   | 48   | 59   | 74   | 96   |
|            | Subtraction | 17                | 23   | 31   | 43   | 61   | 90   |

Table 2.6: Right Lateral PCA Data Reduction

| Image Size | Type        | Variance Retained |      |      |      |      |      |
|------------|-------------|-------------------|------|------|------|------|------|
|            |             | 0.70              | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| 16x16      | Quotient    | 18                | 22   | 29   | 33   | 41   | 56   |
|            | Subtraction | 10                | 12   | 15   | 19   | 26   | 40   |
| 32x32      | Quotient    | 23                | 28   | 35   | 44   | 56   | 77   |
|            | Subtraction | 12                | 15   | 20   | 27   | 38   | 62   |
| 64x64      | Quotient    | 30                | 37   | 45   | 56   | 72   | 96   |
|            | Subtraction | 18                | 24   | 33   | 46   | 65   | 93   |

Table 2.7: Left Posterior Oblique PCA Data Reduction

| Image Size | Type        | Variance Retained |      |      |      |      |      |
|------------|-------------|-------------------|------|------|------|------|------|
|            |             | 0.70              | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| 16x16      | Quotient    | 20                | 24   | 29   | 35   | 44   | 60   |
|            | Subtraction | 10                | 13   | 16   | 20   | 27   | 41   |
| 32x32      | Quotient    | 26                | 32   | 39   | 48   | 60   | 81   |
|            | Subtraction | 13                | 16   | 20   | 27   | 38   | 60   |
| 64x64      | Quotient    | 32                | 38   | 46   | 57   | 72   | 94   |
|            | Subtraction | 17                | 22   | 29   | 40   | 58   | 87   |

Table 2.8: Right Posterior Oblique PCA Data Reduction

| Image Size | Type        | Variance Retained |      |      |      |      |      |
|------------|-------------|-------------------|------|------|------|------|------|
|            |             | 0.70              | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| 16x16      | Quotient    | 21                | 25   | 30   | 36   | 46   | 62   |
|            | Subtraction | 10                | 13   | 16   | 21   | 28   | 42   |
| 32x32      | Quotient    | 28                | 33   | 40   | 50   | 63   | 83   |
|            | Subtraction | 13                | 16   | 21   | 27   | 38   | 60   |
| 64x64      | Quotient    | 33                | 40   | 49   | 60   | 74   | 95   |
|            | Subtraction | 17                | 22   | 29   | 40   | 58   | 88   |

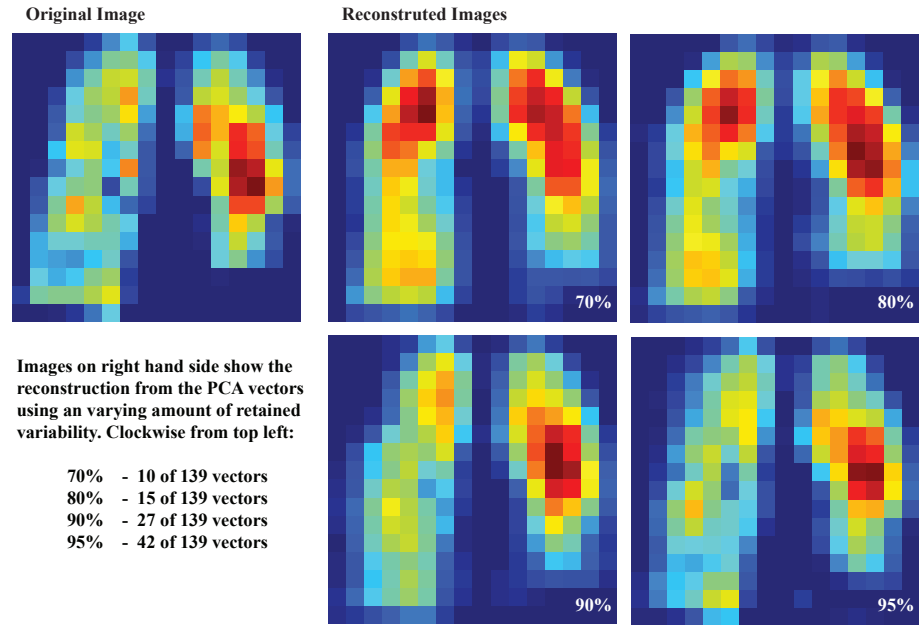


Figure 2.10: Reconstruction of posterior lung image [16x16] using varying amounts of VR

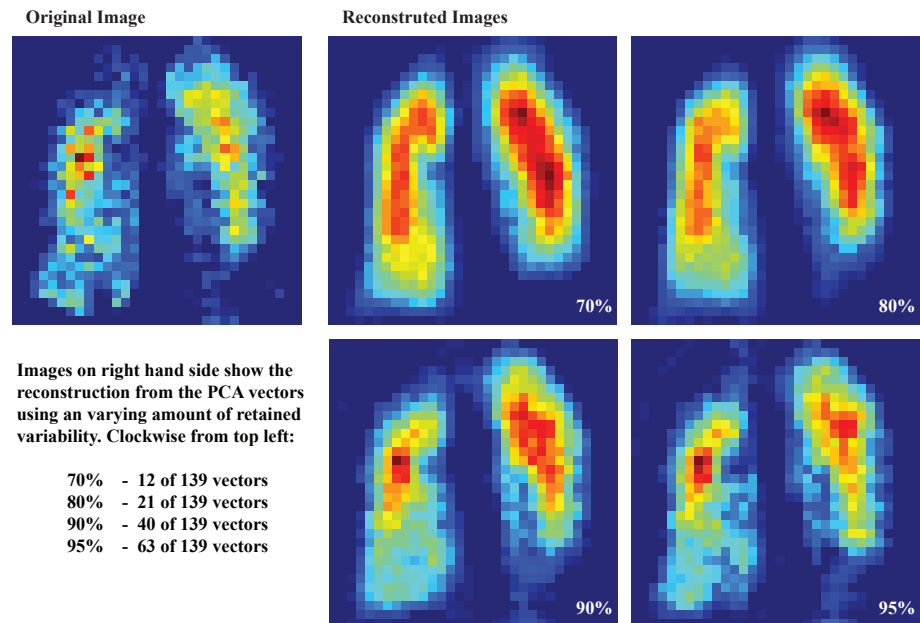


Figure 2.11: Reconstruction of posterior lung image [32x32] using varying amounts of VR

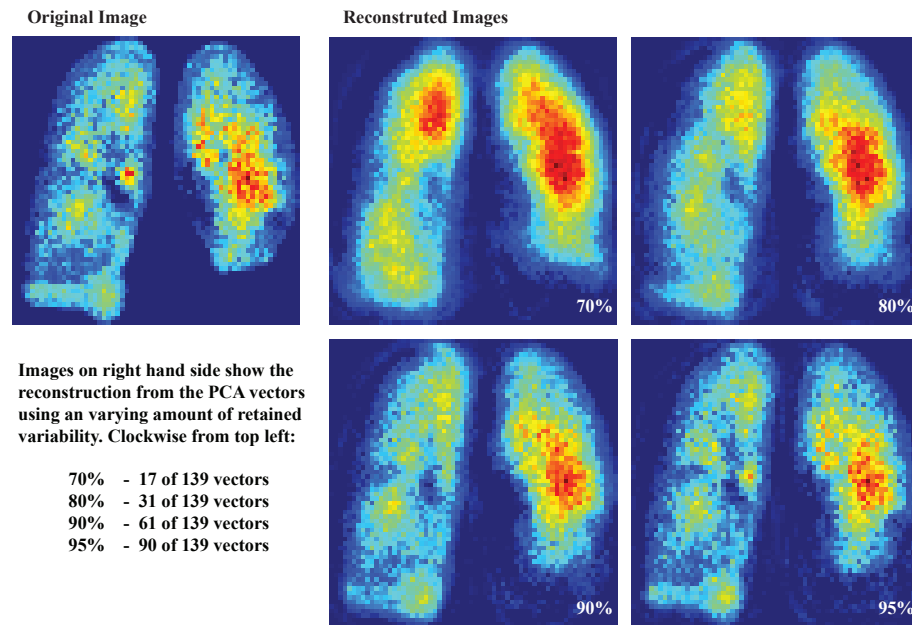


Figure 2.12: Reconstruction of posterior lung image [64x64] using varying amounts of VR

## Grid Images

To use each pixel as input to a neural network classifier, would required 65535 inputs which is impractical and would require vast amounts of training data. So in order to keep the dimensionality of the data to a minimum, each lung image is converted into a grid image. This also allows for the removal of most of the black background in the original 256x256 image as these areas do not contain any information about the probability of pulmonary embolism.

The first stage in the conversion is to find the bounding rectangle of the lung. This region is then converted into a grid of specified size, regardless of the size and aspect ratio of the bounding rectangle. Features are then extracted from these grid images using the methods described above. Grid sizes of 16x16, 32x32 and 64x64 were used for comparative purposes.

### 2.5.5 Input Selection

After the number of inputs has been reduced, it can be further minimised by using input selection methods. For this the statistical overlay function (SoF) is employed, taken from Marwala [2001]. The SoF equation is given in Equation 2.8.

$$\delta = \left\| \frac{\mu_1 - \mu_2}{\frac{\sigma_1 + \sigma_2}{2}} \right\| \quad (2.8)$$

Each input to the system can be said to have a distribution of possible values. The goal of input selection is to choose those inputs whose distributions show the greatest amount of separation across the different classes of output. This concept is illustrated in Figure 2.13. The greater the value of  $\delta$ , the more separation there is between the two inputs. So  $\delta$  between  $\mu_1, \sigma_1$  and  $\mu_3, \sigma_3$  would be greater than the  $\delta$  between  $\mu_1, \sigma_1$  and  $\mu_2, \sigma_2$ . The number of selected inputs was varied between 10, 20 and 30 for comparative purposes.

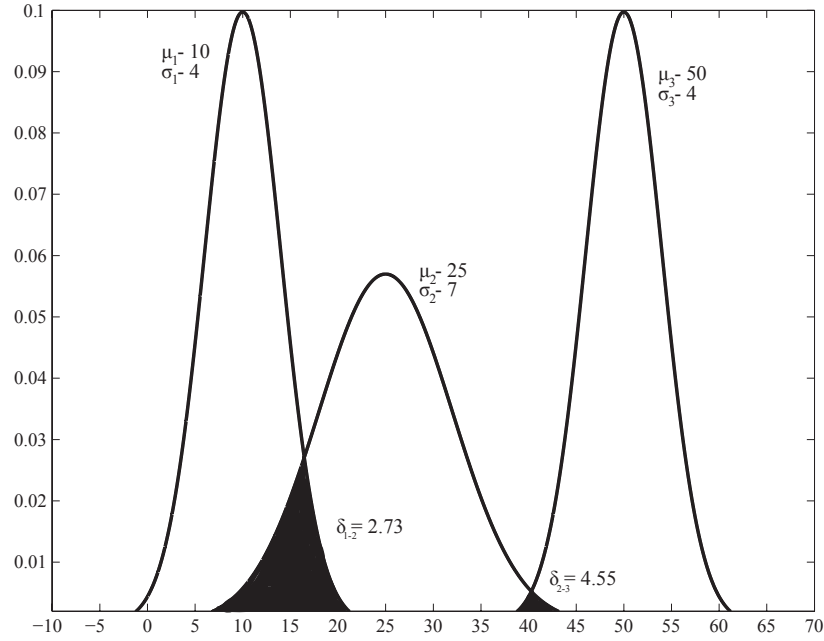


Figure 2.13: Illustration of the Statistical Overlay Function ( $\delta$ ).

### 2.5.6 Classification

#### Bayesian Decision Theory

Bayesian Neural Networks are based on Bayesian statistics. The objective of a Bayesian Neural Network is to maximise the a posteriori class probabilities. Assume that  $\vec{x}$  represents an n-dimensional feature vector obtained from the image that needs to be classified. Next assume that  $w_i$  represent the classes to which the image can belong. The Bayes rule (2.9) can be stated as follows (Koutroumbas and Theodoridis [1999]):

$$P(w_i|\vec{x}) = \frac{p(\vec{x}|w_i)P(w_i)}{p(\vec{x})} \quad (2.9)$$

where  $p(\vec{x}|w_i)$  is the likelihood function (also known as the class-conditional probability), and  $P(w_i)$  is the priori probabilities for the different classes and  $p(\vec{x})$  is the normalisation factor, otherwise known as the evidence. The Bayes Classification Rule then becomes (Koutroumbas and Theodoridis [1999]):

$$\text{If } P(w_i|\vec{x}) > P(w_k|\vec{x}); k = 1...n; i \neq k, \text{ is classified to } w_i \quad (2.10)$$

This classification rule is chosen as it always minimizes the probability of error for the system (Duda and Hart [1973]). In some cases, a more appropriate method would be to associate a loss function with a particular action to be taken. Let this loss term be defined by  $\lambda(\alpha_i, w_j)$  (Duda and Hart [1973]). This loss function computes the loss associated with taking a decision  $\alpha_i$  when the true class is  $w_j$ . Because it is already known that the probability  $\vec{x}$  belongs to  $w_j$  is given by  $P(w_j|\vec{x})$ , therefore, the expected loss associated with taking a decision  $\alpha_i$  becomes,

$$R(\alpha_i, \vec{x}) = \sum \lambda(\alpha_i|w_j)P(w_j|\vec{x}) \quad (2.11)$$

This is known as the *conditional risk*. The problem now becomes a search for a decision rule that minimises the *overall risk*, given below in Eq. (2.12) .

$$R_{overall} = \int R(\alpha(\vec{x})|\vec{x})p(\vec{x})d\vec{x} \quad (2.12)$$

A decision rule  $\alpha(\vec{x})$ , also known as a *discriminant function* is any function that can be used to determine a decision boundary. Clearly, if  $\alpha(\vec{x})$  is chosen to minimize the conditional risk for all  $\vec{x}$ , the overall risk, will then be minimised. Therefore, to minimise the overall risk compute the conditional risk for  $i = 1, \dots, n$  and make the decision  $\alpha_i$  for which the conditional risk  $R(\alpha_i, \vec{x})$  is a minimum. The resulting minimum overall risk is known as the *Bayes Risk* and ensures optimal performance for the system.

## The Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is a widely used architecture for the construction of neural networks. The topology of an MLP is shown in Figure 2.14.

The hidden and output layers both have activation functions, which relate to the relationship between the input and output of each node. Because the data to be classified is highly non-linear, hyperbolic tangent functions are used in the hidden layers. To keep the output between 0 and 1 a logistic function is used on the output node.

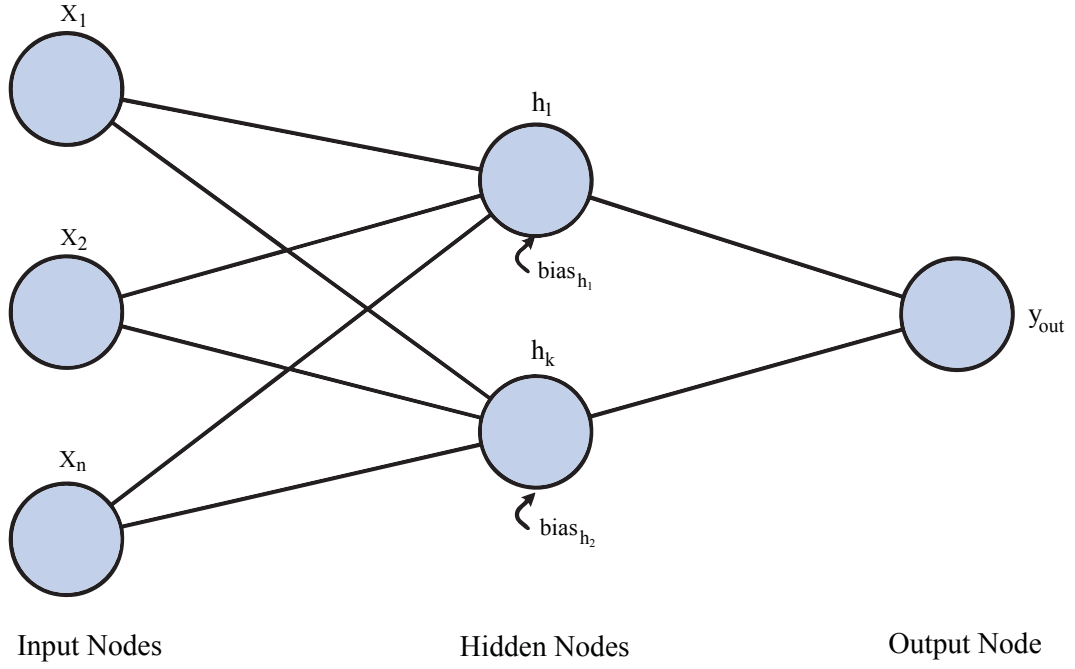


Figure 2.14: Multi-Layer perceptron architecture

### Bayesian MLP

A Bayesian MLP uses the standard MLP structure. The difference is in the training method for the network. The training uses the Hybrid-Monte-Carlo method (HMC). Samples for the weights of the neural network are generated using a Markov-Chain. These samples are then either accepted or rejected depending on a threshold value.

A Markov-Chain is a stochastic process whereby the probability that a new state (sample weights) is accepted depends solely upon the current state and not on any of the previous states (Norris [1998]). Thus a Markov-Chain can be said to be a memoryless system. These changes in state are known as transitions and can be formally stated in Equation 2.13.

$$Pr(X_{x+1} = x | X_0 = x_0, X_1 = x_1, \dots, X_n = x_n) = Pr(X_{n+1} = x | X_n = x_n) \quad (2.13)$$

We can see that the probability of a state transition, given the current and past states is a function of the current state only. This is the Markov property.



## 2.6 System Training

The Bayesian MLP was trained in Matlab© using the NETLAB toolbox (Nabney [2002]). The network consisted of a single input, hidden and output layer. The number of input nodes was varied between 10, 20 and 30. The hidden layer contained 5 nodes while the output layer contained 1 node.

Of the 179 cases, 125 (70%) were used for training (49 negative, 56 intermediate, 20 high), while 54 (30%) were used for validation (27 negative, 20 intermediate, 7 high). The network was trained using the Hybrid-Monte-Carlo method, resulting in a committee of 250 neural networks.

A very important aspect of the neural network training is the scaling of the input data. It is essential that the input data is scaled to be in the range  $[0 \ 1]$ . The scaling was achieved using the formula given in Equation 2.14.

$$X = \left( \frac{X - X_{min}}{X_{max} - X_{min}} \right) \quad (2.14)$$

The scaled PCA inputs were fed into each of the 250 networks and the final classification was based upon the mean value of the committee output. Because each member of the committee gives an individual output, 95% confidence levels can be calculated by taking into account the standard deviations ( $\sigma$ ) of the output distributions. So outputs that exhibit a small  $\sigma$  imply a high confidence, while outputs exhibiting a large  $\sigma$  imply a low confidence.

Table 2.9 gives a summary of the network parameters.

Table 2.9: Summary of Hybrid Monte-Carlo network parameters

| Parameter                          | Value |
|------------------------------------|-------|
| Number of retained samples         | 250   |
| Samples rejected at start of chain | 200   |
| Number of steps in trajectory      | 100   |
| Step Size                          | 0.001 |

The results from the training and validation data sets are in Chapter 4.

## Chapter 3

# Software

### 3.1 PEDiag

PEDiag is a software program developed in MATLAB© for this project. The program takes the 12 lung images as inputs and determines if there is a possibility of pulmonary embolism. The software performs all the image processing, alignment and classification. This section contains a functional breakdown of all code (.m files) used for this project including inputs, outputs and flow diagrams.

Figure 3.1 shows a screen shot of PEDiag. The top two images show the anterior perfusion (left) and left lateral ventilation (right) before image segmentation. The bottom 6 images represent the subtraction images, clockwise from the left: anterior, left lateral, left posterior oblique, posterior, right lateral and right posterior oblique. The buttons down the right hand side provide a means of segmenting and classifying the image set.

The network selection provides the user with the ability to select which neural network they wish to use. The selection of the network is based on the image type, variance retained and the number of inputs to the network.

The images can also be exported to portable network graphics (PNG) format. Two images are exported, the first contains all the images before segmentation while the second image shows the segmented images.

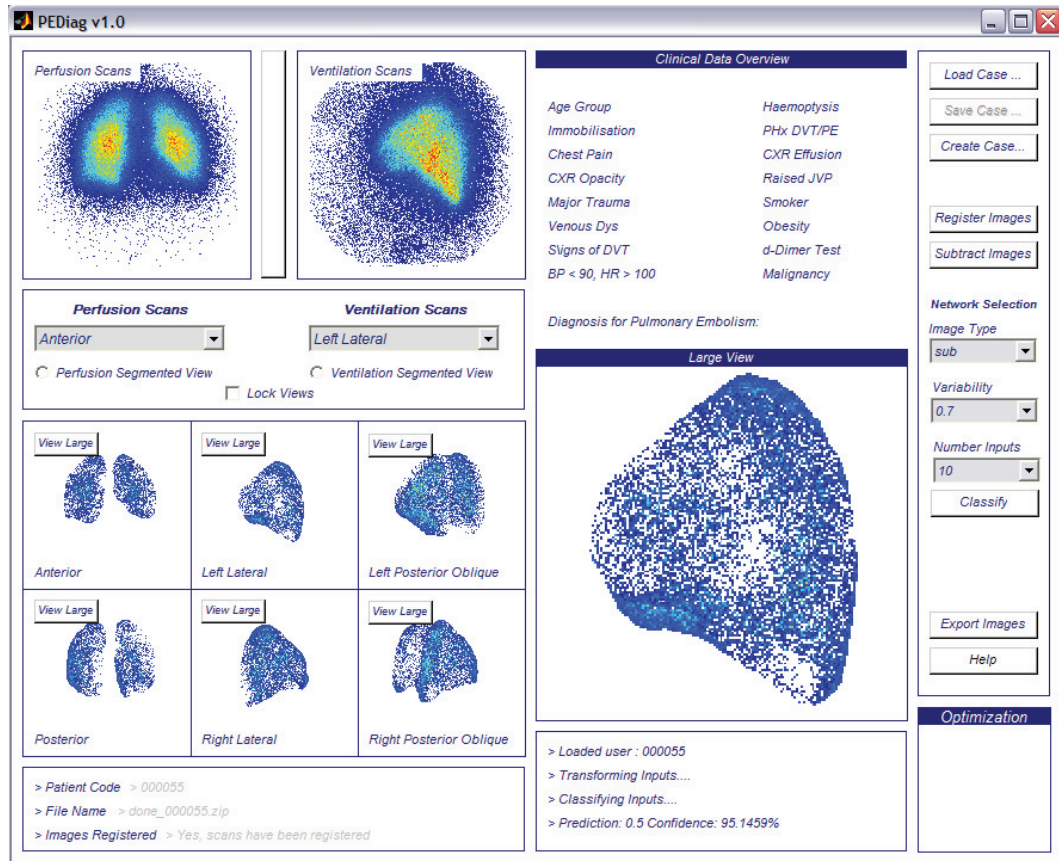


Figure 3.1: Screen shot of PEDiag.

### 3.2 Medical Image Viewer

The Medical Image Viewer was developed in *Microsoft© Visual C# Express Edition*. The viewer allows all cases to be reviewed by different nuclear medicine physicians. Each nuclear medicine physician can then submit their findings to the project.

This image viewing software is intended to enable future development of this work, and was not used in this project. More information on the viewer can be found in Appendix B.

## Chapter 4

# Results and Discussion

The results are split into two sections: training performance (Section 4.1) and validation performance (Section 4.2). The performance is measured using confusion matrices and is measured in the following way:

Table 4.1: Example of a confusion matrix showing how the performance is calculated

|                     | Actual | Predicated |              |       |
|---------------------|--------|------------|--------------|-------|
|                     |        | Negative   | Intermediate | High  |
| <b>Negative</b>     | $a_1$  | $p_1$      |              |       |
| <b>Intermediate</b> | $a_2$  |            | $p_2$        |       |
| <b>High</b>         | $a_3$  |            |              | $p_3$ |

The values  $a_1, \dots, a_3$  represent the actual amount of each class based on the gold standard diagnosis <sup>1</sup>. The values  $p_1, \dots, p_3$  show the predictions of the neural network. The performance is then calculated using Equation 4.1.

$$Performance = \sqrt{\frac{p_1 \cdot p_2 \cdot p_3}{a_1 \cdot a_2 \cdot a_3}} \cdot 100 \quad (4.1)$$

Equation 4.1 is very sensitive to misclassifications. If a single class is completely misclassified the resulting Performance would be 0%, even if the other classes had a high rate of classification. Only the best results for each different image size using 10, 20 and 30 inputs are shown.

<sup>1</sup>The Gold Standard was obtained from the actual clinical report in each case

## 4.1 Training Performance

Tables 4.2 through 4.4 reveal the VR that yields the best network training performance for image sizes of 16x16, 32x32 and 64x64. The number of inputs to the classifier is varied between 10, 20 and 30 for each image size. The implications of these tables is discussed further in Section 4.4.

### 4.1.1 16x16 Image Size

Table 4.2: Training confusion matrices for 10, 20 and 30 inputs. 16x16 image size.

|                                                                           | Actual | Predicated |              |      |
|---------------------------------------------------------------------------|--------|------------|--------------|------|
|                                                                           |        | Negative   | Intermediate | High |
| <b>Negative</b>                                                           | 49     | 39         | 10           | 0    |
| <b>Intermediate</b>                                                       | 56     | 3          | 53           | 0    |
| <b>High</b>                                                               | 20     | 0          | 1            | 19   |
| <i>10 Inputs: Training Performance : 84.60% — Variance Retained : 90%</i> |        |            |              |      |
| <b>Negative</b>                                                           | 49     | 48         | 1            | 0    |
| <b>Intermediate</b>                                                       | 56     | 0          | 56           | 0    |
| <b>High</b>                                                               | 20     | 0          | 0            | 20   |
| <i>20 Inputs: Training Performance : 98.97% — Variance Retained: 70%</i>  |        |            |              |      |
| <b>Negative</b>                                                           | 49     | 49         | 0            | 0    |
| <b>Intermediate</b>                                                       | 56     | 0          | 56           | 0    |
| <b>High</b>                                                               | 20     | 0          | 0            | 20   |
| <i>30 Inputs: Training Performance : 100.00% — Variance Retained: 90%</i> |        |            |              |      |

#### 4.1.2 32x32 Image Size

Table 4.3: Training confusion matrix for 10, 20 and 30 inputs. 32x32 image size.

|                                                                            | Actual | Predicated |              |      |
|----------------------------------------------------------------------------|--------|------------|--------------|------|
|                                                                            |        | Negative   | Intermediate | High |
| <b>Negative</b>                                                            | 49     | 40         | 9            | 0    |
| <b>Intermediate</b>                                                        | 56     | 3          | 53           | 0    |
| <b>High</b>                                                                | 20     | 0          | 1            | 19   |
| <i>10 Inputs: Training Performance : 85.67% — Variance Retained : 90%</i>  |        |            |              |      |
| <b>Negative</b>                                                            | 49     | 49         | 0            | 0    |
| <b>Intermediate</b>                                                        | 56     | 0          | 56           | 0    |
| <b>High</b>                                                                | 20     | 0          | 0            | 20   |
| <i>20 Inputs: Training Performance : 100.00% — Variance Retained : 75%</i> |        |            |              |      |
| <b>Negative</b>                                                            | 49     | 49         | 0            | 0    |
| <b>Intermediate</b>                                                        | 56     | 0          | 56           | 0    |
| <b>High</b>                                                                | 20     | 0          | 0            | 20   |
| <i>30 Inputs: Training Performance : 100.00% — Variance Retained : 95%</i> |        |            |              |      |

### 4.1.3 64x64 Image Size

Table 4.4: Training confusion matrix for 10 inputs. 64x64 image size.

|                                                                            | Actual | Predicated |              |      |
|----------------------------------------------------------------------------|--------|------------|--------------|------|
|                                                                            |        | Negative   | Intermediate | High |
| <b>Negative</b>                                                            | 49     | 41         | 8            | 0    |
| <b>Intermediate</b>                                                        | 56     | 8          | 48           | 0    |
| <b>High</b>                                                                | 20     | 0          | 0            | 20   |
| <i>10 Inputs: Training Performance : 84.68% — Variance Retained : 75%</i>  |        |            |              |      |
| <b>Negative</b>                                                            | 49     | 48         | 1            | 0    |
| <b>Intermediate</b>                                                        | 56     | 0          | 56           | 0    |
| <b>High</b>                                                                | 20     | 0          | 1            | 19   |
| <i>20 Inputs: Training Performance : 96.46% — Variance Retained : 95%</i>  |        |            |              |      |
| <b>Negative</b>                                                            | 49     | 49         | 0            | 0    |
| <b>Intermediate</b>                                                        | 56     | 0          | 56           | 0    |
| <b>High</b>                                                                | 20     | 0          | 0            | 20   |
| <i>30 Inputs: Training Performance : 100.00% — Variance Retained : 90%</i> |        |            |              |      |

## 4.2 Validation Performance

Tables 4.5 through 4.7 reveal the VR that yields the best network validation performance for image sizes of 16x16, 32x32 and 64x64. The number of inputs to the classifier is varied between 10, 20 and 30 for each image size. The implications of these tables is discussed further in Section 4.4.

### 4.2.1 16x16 Image Size

Table 4.5: Validation confusion matrix for 10, 20 and 30 inputs. 16x16 image size.

|                                                                             | Actual | Predicated |              |      |
|-----------------------------------------------------------------------------|--------|------------|--------------|------|
|                                                                             |        | Negative   | Intermediate | High |
| <b>Negative</b>                                                             | 27     | 12         | 10           | 5    |
| <b>Intermediate</b>                                                         | 20     | 4          | 14           | 2    |
| <b>High</b>                                                                 | 7      | 1          | 2            | 4    |
| <i>10 Inputs: Validation Performance : 42.16% — Variance Retained : 90%</i> |        |            |              |      |
| <b>Negative</b>                                                             | 27     | 16         | 6            | 5    |
| <b>Intermediate</b>                                                         | 20     | 8          | 8            | 4    |
| <b>High</b>                                                                 | 7      | 2          | 1            | 4    |
| <i>20 Inputs: Validation Performance : 36.80% — Variance Retained : 70%</i> |        |            |              |      |
| <b>Negative</b>                                                             | 27     | 17         | 8            | 2    |
| <b>Intermediate</b>                                                         | 20     | 9          | 8            | 3    |
| <b>High</b>                                                                 | 7      | 1          | 2            | 4    |
| <i>30 Inputs: Validation Performance : 38.00% — Variance Retained : 90%</i> |        |            |              |      |



#### 4.2.2 32x32 Image Size

Table 4.6: Validation confusion matrix for 10, 20 and 30 inputs. 32x32 image size.

|                                                                             | Actual | Predicated |              |      |
|-----------------------------------------------------------------------------|--------|------------|--------------|------|
|                                                                             |        | Negative   | Intermediate | High |
| <b>Negative</b>                                                             | 27     | 13         | 11           | 3    |
| <b>Intermediate</b>                                                         | 20     | 4          | 12           | 4    |
| <b>High</b>                                                                 | 7      | 1          | 2            | 4    |
| <i>10 Inputs: Validation Performance : 40.63% — Variance Retained : 90%</i> |        |            |              |      |
| <b>Negative</b>                                                             | 27     | 18         | 7            | 2    |
| <b>Intermediate</b>                                                         | 20     | 10         | 8            | 2    |
| <b>High</b>                                                                 | 7      | 1          | 1            | 5    |
| <i>20 Inputs: Validation Performance : 43.64% — Variance Retained : 75%</i> |        |            |              |      |
| <b>Negative</b>                                                             | 27     | 14         | 11           | 2    |
| <b>Intermediate</b>                                                         | 20     | 4          | 10           | 6    |
| <b>High</b>                                                                 | 7      | 2          | 2            | 3    |
| <i>30 Inputs: Validation Performance : 33.33% — Variance Retained : 95%</i> |        |            |              |      |

### 4.2.3 64x64 Image Size

Table 4.7: Validation confusion matrix for 10, 20 and 30 inputs. 64x64 image size.

|                                                                             | Actual | Predicated |              |      |
|-----------------------------------------------------------------------------|--------|------------|--------------|------|
|                                                                             |        | Negative   | Intermediate | High |
| <b>Negative</b>                                                             | 27     | 14         | 6            | 7    |
| <b>Intermediate</b>                                                         | 20     | 5          | 11           | 4    |
| <b>High</b>                                                                 | 7      | 1          | 1            | 5    |
| <i>10 Inputs: Validation Performance : 45.13% — Variance Retained : 75%</i> |        |            |              |      |
| <b>Negative</b>                                                             | 27     | 17         | 5            | 5    |
| <b>Intermediate</b>                                                         | 20     | 9          | 9            | 2    |
| <b>High</b>                                                                 | 7      | 2          | 0            | 5    |
| <i>20 Inputs: Validation Performance : 44.98% — Variance Retained : 95%</i> |        |            |              |      |
| <b>Negative</b>                                                             | 27     | 17         | 9            | 1    |
| <b>Intermediate</b>                                                         | 20     | 10         | 9            | 1    |
| <b>High</b>                                                                 | 7      | 2          | 0            | 5    |
| <i>30 Inputs: Validation Performance : 45.00% — Variance Retained : 90%</i> |        |            |              |      |

### 4.3 Individual Class Performance : All Image Sizes

Figure 4.3 shows the mean classification error of the individual classes with varying amounts of VR.

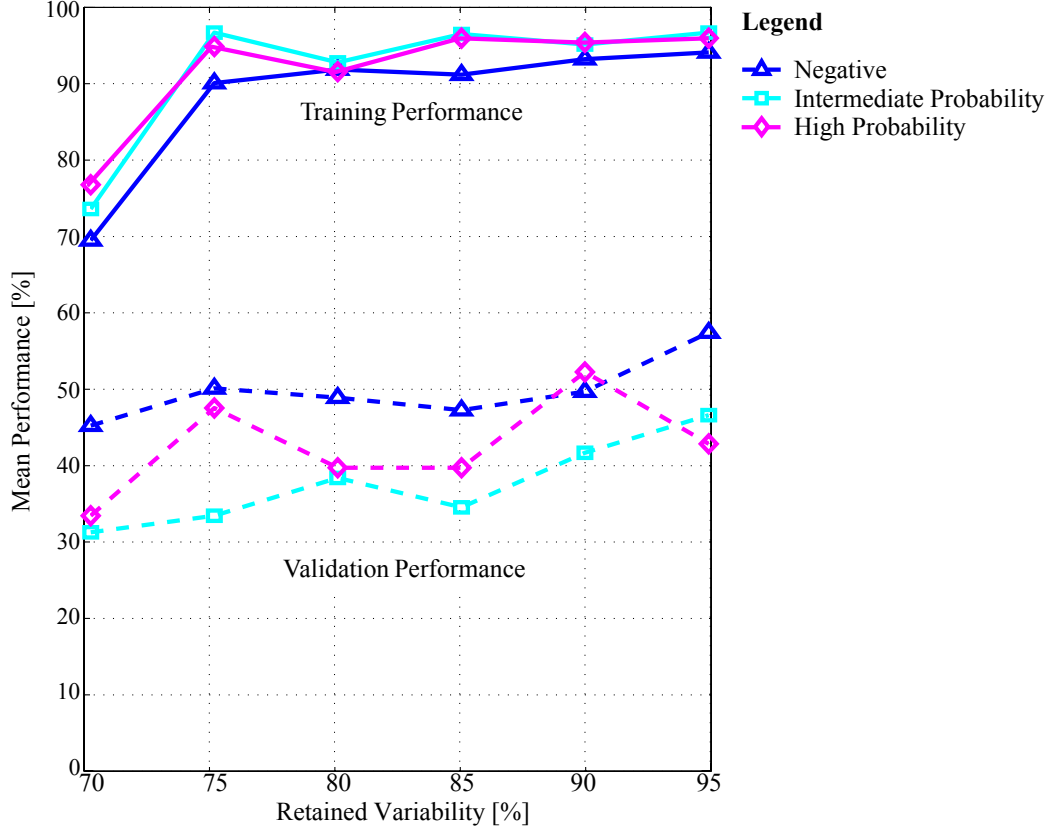


Figure 4.1: Mean Individual class training and validation Performance for different values of VR. All the image sizes are grouped together.

## 4.4 Discussion

### 4.4.1 Performance

From Sections 4.1 and 4.2 it can be seen that the training performance is higher than the validation performance. With a Bayesian network, it has been shown, that with a sufficient training population, the training performance will be a good indication of the error that can be expected on unseen data. This is not the case with these results, so we could assume that the training population is not of a sufficient size.

Referring to the above figure, it is noticeable that the mean performance for the

intermediate probability class is lower than the mean performance for the other two classes, the only exception being the case of a VR of 95%. Another interesting point to note is the mean training and validation performance are reversed for the individual classes, in other words, the negative class has the lowest average training performance, but shows the best mean validation performance, while the intermediate probability class shows the best mean training performance but the lowest mean validation performance.

Overall the negative cases are shown to be diagnosed with the highest degree of accuracy, followed by the high probability cases and lastly the intermediate probability cases. This would confirm what was mentioned earlier in that the intermediate cases are the most difficult to diagnose.

#### **4.4.2 Variance Retained**

The VR, chosen during the PCA analysis is another parameter which was varied. Again from Figure 4.1 a steep increase in training performance is gained between a VR of 70% and 75%. There also appears to be a gradual increase in validation performance with increasing VR.

#### **4.4.3 Image Size & Number of Inputs**

Figures 4.2, 4.3 and 4.4 show the system performance when the number of inputs is varied between 10, 20 and 30.

The performance when using 10 inputs is quite poor, and the training performance is below that of the 20 and 30 input networks. It is interesting to note that while the training performance of 16x16 and 32x32 images is very high, the 64x64 images show extremely poor training performance for a VR of 70% across all input ranges.

Validation performance increases with image size, and the number of inputs to the neural network. The training performance remains fairly constant for input sizes above 10, except in the case mentioned above.

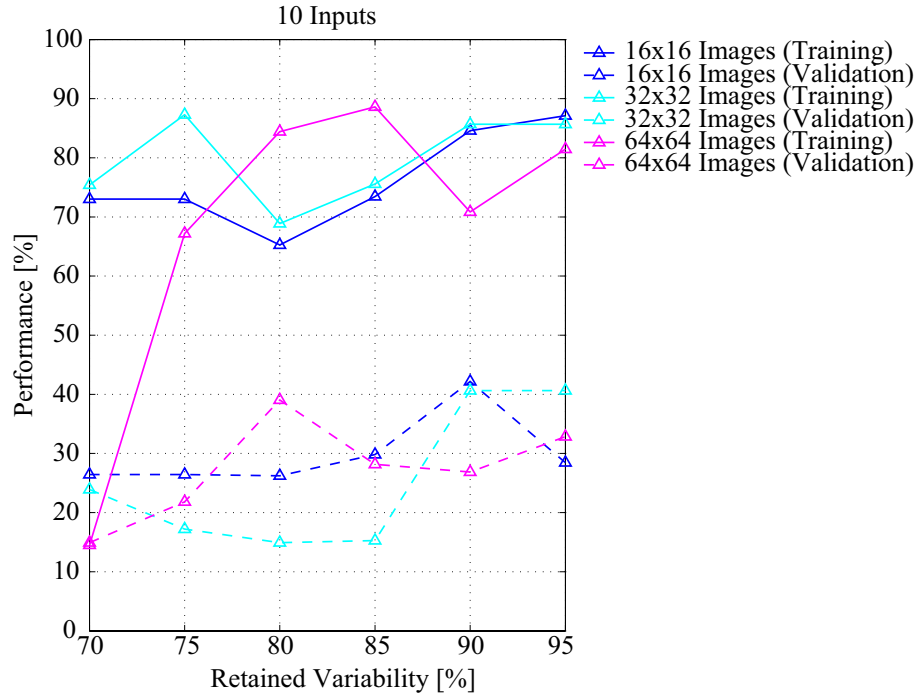


Figure 4.2: Performance vs. VR for 10 inputs. Image size is 16x16, 32x32 and 64x64.

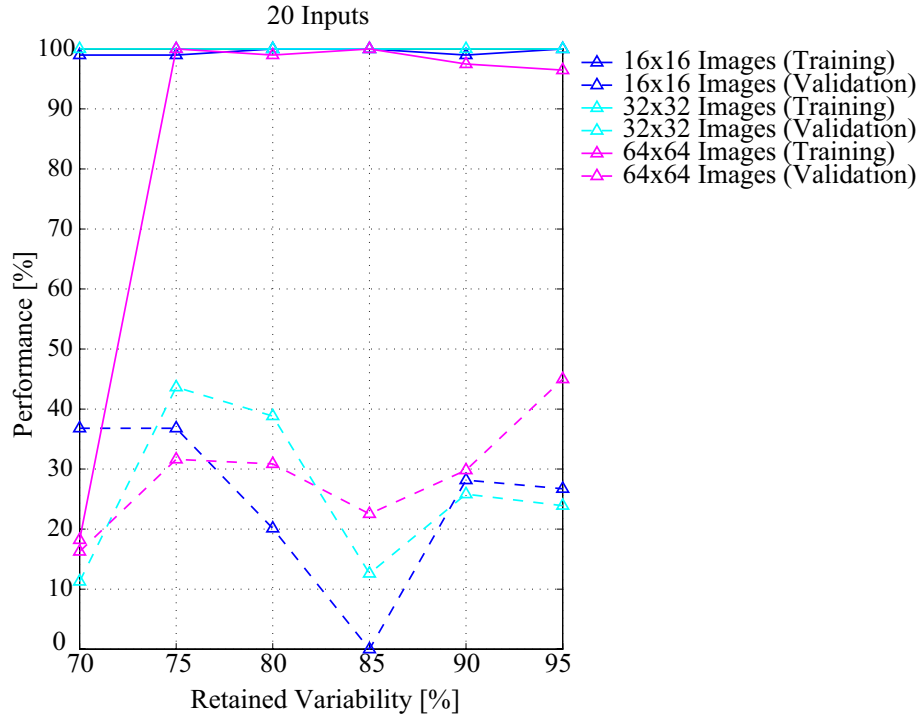


Figure 4.3: Performance vs. VR for 20 inputs. Image size is 16x16, 32x32 and 64x64.

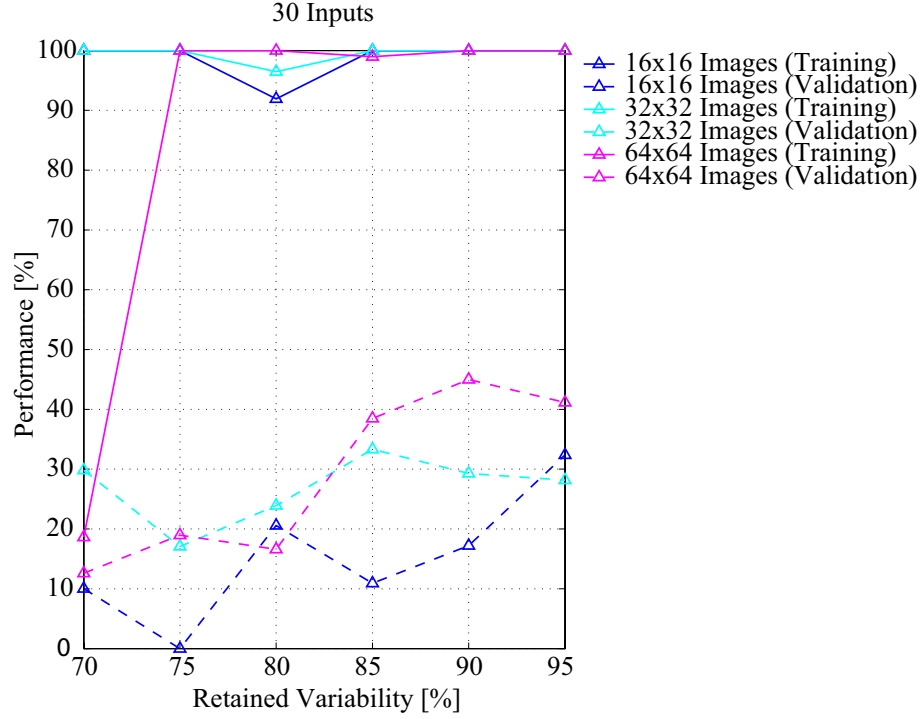


Figure 4.4: Performance vs. VR for 30 inputs. Image size is 16x16, 32x32 and 64x64.

#### 4.4.4 Sensitivity, Specificity, Positive and Negative predictive values

Table 4.8 shows the sensitivities, specificities, positive and negative predictive values for the different class outputs. These values are calculated using the validation performance, not the training performance. The specificity and NPV of the high and intermediate probability classes exhibit high means, indicating that PE can be reasonably excluded should a high or intermediate probability of PE not be predicted.

The negative cases show the highest sensitivity, while the intermediate probability cases show the lowest sensitivity. This is as expected, as in previous research the intermediate cases have been shown to be the most difficult to diagnose accurately.

In order to compare these results with those of previous authors, an receiver operating characteristic (ROC) curve is needed. As most of the previous works only considered a 2-class classifier, it was decided that for comparative purposes the intermediate probability cases would be grouped together with the high probability cases to form a new "positive" class. The ROC curve is shown in Figure 4.5. The ROC was generated from a network using an image size of 64x64, 30 inputs and a VR of 95%.

Table 4.8: Table showing sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV) using the validation dataset.

|                                                    |             | Output Class |              |              |
|----------------------------------------------------|-------------|--------------|--------------|--------------|
|                                                    |             | Negative     | Intermediate | High         |
| <b>Sensitivity [%]</b>                             | max         | 70.37        | 55.00        | 71.43        |
|                                                    | min         | 18.52        | 0.00         | 14.29        |
|                                                    | <b>mean</b> | <b>48.35</b> | <b>30.84</b> | <b>46.83</b> |
| <b>Specificity [%]</b>                             | max         | 77.14        | 94.44        | 95.92        |
|                                                    | min         | 61.37        | 68.00        | 68.11        |
|                                                    | <b>mean</b> | <b>69.01</b> | <b>82.01</b> | <b>81.03</b> |
| <b>PPV [%]</b>                                     | max         | 61.54        | 62.50        | 71.43        |
|                                                    | min         | 33.33        | 0            | 6.25         |
|                                                    | <b>mean</b> | <b>50.97</b> | <b>41.24</b> | <b>25.95</b> |
| <b>NPV [%]</b>                                     | max         | 77.14        | 79.07        | 95.92        |
|                                                    | min         | 55.10        | 62.96        | 88.68        |
|                                                    | <b>mean</b> | <b>66.42</b> | <b>71.44</b> | <b>92.71</b> |
| Refer to Appendix C for definitions of parameters. |             |              |              |              |

The area under curve (AUC) is 0.64. This compares to a AUC of 0.86 achieved in Ericsson et al. [2003]. An interesting comparison is to the work done in Frigyesi [2003] where an AUC of 0.85 was achieved when the gold standard was angiography, while an AUC of 0.67 was achieved when the gold standard was the consensus opinion of nuclear medicine physicians. This compares favourably with the AUC of 0.64 in this study, where the gold standard was also the opinion of a nuclear medicine physician.

## 4.5 Conclusion

Although performance is below that of previous works, it has been shown that it is feasible to diagnose PE automatically using a Bayesian Neural Network. The most critical areas in the process are those of image segmentation, feature extraction and input normalisation. Without correct scaling of the input data, the network simply does not "work". The image segmentation routines proved to be very effective in most circumstances. There were however a few instances where manual touch-up was needed. This was mostly done on ventilation images where the scans contained large amounts of noise.

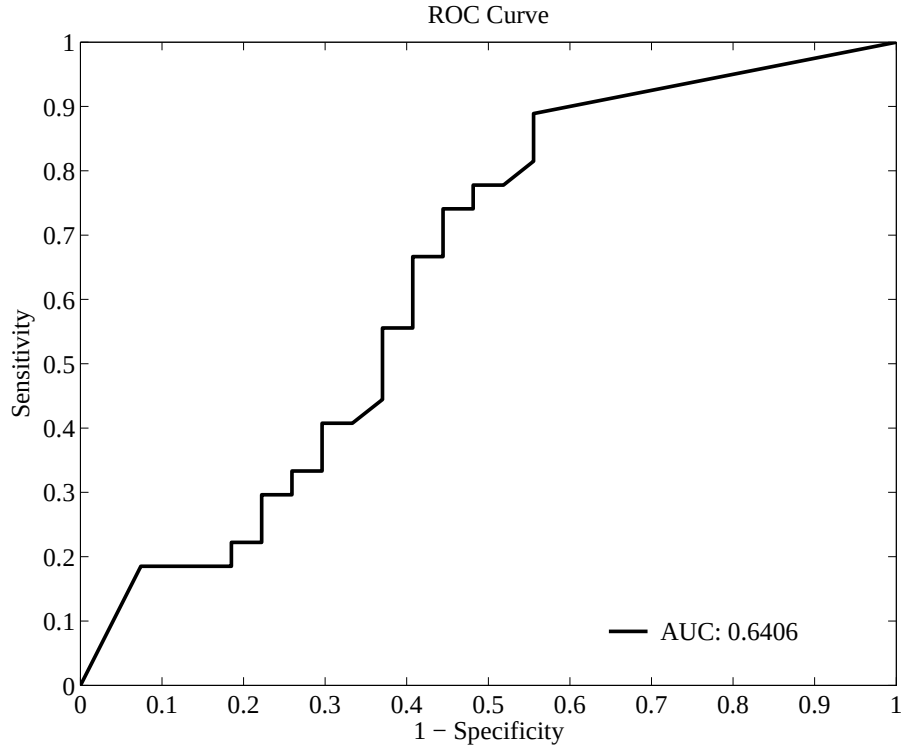


Figure 4.5: ROC curve for Bayesian Classifier. 64x64 image size, using 30 inputs and a VR of 95%. The AUC is 0.64. Intermediate and high probability cases are grouped together.

From the results, it would appear that the best system performance is achieved by having a network with 30 inputs using an image size of 64x64 and a VR of 90-95%. Intermediate cases were shown to be the most difficult to diagnose correctly, while specificity calculations show that PE can reasonably be excluded if the prediction is not positive.

This research could be furthered by the collection of more cases to provide a more balanced data set including equal numbers of each class for both training and validation. It is likely that a large increase in available cases will greatly improve system performance.

If possible, using a different *Gold Standard*, angiography for instance, may improve the performance of the Bayesian classifier. The software shown in Appendix B could also be used to obtain opinions from more physicians.



## References

- H Bounameaux, P de Moerloose, A Perrier, and MJ Miron. D-dimer testing in suspected venous thromboembolism: an update. *Quarterly Journal of Medicine*, 90:437–442, 1997.
- C.H Dagli, A.L Buczak, J Ghosh, M.J Embrechts, O Ersoy, and S Kerckel. *Intelligent Engineering Systems Through Artificial Neural Networks: Volume 10, Fuzzy Logic and Evolutionary Programming*. Amer Society of Mechanical Engineers, November 2000. 753-759. ISBN: 0791801616.
- E Drucker, M Rivitz, and J Shepard. Acute pulmonary embolism: assessment of helical ct for diagnosis. *Radiology*, 209:235–241, 1998.
- R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc., 1973.
- A Ericsson, A Huart, A Ekefj rd, K  str m, H Holst, E Evander, P Wollmer, and L Edenbrandt. Automated interpretation of ventilation-perfusion lung scintigrams for the diagnosis of pulmonary embolism using Support Vector Machines. In *13th Scandinavian Conference on Image Analysis*. SCIA, June 2003.
- M Fiszman, P.J Haug, and P.R Frederick. Automatic extraction of PIOPED interpretations from Ventilation/Perfusion (V/Q) lung scan reports. In *AMIA 98 Symposium*. American Medical Informatics Association, November 1998.
- A. Frigyesi. An automated method for the detection of pulmonary embolism in V/Q scans. *Medical Image Analysis*, 7(7):341–349, 2003.
- K Garg, C Welsh, and A Feyerabend. Pulmonary embolism: diagnosis with spiral ct and ventilation-perfusion scanning - correlation with pulmonary angiographic results or clinical outcome. *Radiology*, 208:201–208, 1998.
- R.C. Gonzalez and P. Wintz. *Digital image processing*. Addison Wesley, Reading, Massachusetts, 2nd ed edition, 1987.

- R.D Hull, J Hirsh, C.J Carter, R.M Jay, P.E Dodd, P.A Ockelford, G Coates, G.J Gill, A.G.G Turpie, D.J Doyle, H.R Buller, and G.E Raskob. Pulmonary angiography, ventilation lung scanning, and veno-graphy for clinically suspected pulmonary embolism with abnormal perfusion lung scan. *Annals of Internal Medicine*, 98(98): 891–899, 1983.
- Christopher W. Humphreys, Lisa K. Moores, and Andrew F. Shorr. Cost-minimization of two algorithms for diagnosing acute pulmonary embolism. *Thrombosis Research*, 113:275–282, 2004.
- Dr Hustinx. *Procedure Guidelines for Lung Scintigraphy*. Belgian Society of Nuclear Medicine, 2002. URL <http://www.belnuc.be/pages/docs/index.htm>.
- T.M Hyers, R.D Hull, and J.C Weg. Antithrombotic therapy for venous thromboembolic disease. *Chest*, 119(108):176–193, 2001.
- I.T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, October 1 2002.
- K. Koutroumbas and S. Theodoridis. *Pattern Recognition*. Academic Press, 1999.
- J Koza. Genetic programming. Technical report, Stanford University, 1997.
- D Lindahl, J Lanke, A Lundin, J Palmer, and L Edenbrandt. Improved classifications of myocardial bull’s eye scintigrams with computer-based decision support systems. *Journal of Nuclear Medicine*, 40:96–101, 1999.
- L.J Mango. Computer-assisted cervical cancer screening using neural networks. *Cancer Letters*, 77:155–162, 1994.
- T. Marwala. *Fault identification using neural networks and vibration data*. PhD thesis, University of Cambridge, 2001.
- J.R Mayo, M Remy-Jardin, and N.L Müller. Pulmonary embolism: prospective comparison of spiral ct with ventilation/perfusion scintigraphy. *Radiology*, 205: 447–452, 1997.
- PIOPED Investigators. Value of the ventilation/perfusion scan in acute pulmonary embolism. Results of the Prospective Investigation of Pulmonary Embolism Diagnosis study (PIOPED). *The Journal of the American Medical Association (JAMA)*, 1990(20):2753–2759, 1990.
- L.K. Moore, J.F Collen, et al. Practical utility of clinical prediction rules for suspected acute pulmonary embolism in a large academic institution. *Thrombosis Research*, 113(113):1–6, 2004.
- Ian T Nabney. NETLAB. www, June 2002.

- James R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.
- M Remy-Jardin, J Remy, and F Deschildre. Diagnosis of pulmonary embolism with spiral ct: comparison with pulmonary angiography and scintigraphy. *Radiology*, 200:699–706, 1996.
- M Remy-Jardin, J Remy, L Wattinne, and F Giraud. Central pulmonary thromboembolism: diagnosis with spiral volumetric ct with single-breath-hold technique comparison with pulmonary angiography. *Radiology*, 185:381–387, 1992.
- M Rodger and P.S Wells. Diagnosis of pulmonary embolism. *Thrombosis Research*, 103:224–238, 2001.
- A.B Van Rossum, P.M.T Pattynama, and A Tjin. Pulmonary embolism: Validation of spiral ct-angiography in 124 patients. *Radiology*, 201:467–470, 1996.
- M Sadik, D Jakobsson, F Olofsson, M Ohlsson, Madis Suurkula, and L Edenbrandt. A new computer-based decision-support system for the interpretation of bone scans. *Nuclear Medicine Communications*, 27:417–423, 2006.
- N Schluger, C Henschke, T King, R Russo, B Binkert, M Rackson, and D Hayk. Diagnosis of pulmonary embolism at a large teaching hospital. *Journal of Thoracic Imaging*, (9):180–184, 1994.
- J.A Scott and E.L Palmer. Neural network analysis of ventilation-perfusion lung scans. *Radiology*, 186:661–664, 1993.
- G Serpen, R Iyer, H.M Elsamaloty, and EI Parsai. Automated lung outline reconstruction in ventilation-perfusion scans using principle component analysis techniques. *Computers in Biology and Medicine*, 33:119–142, 2003.
- C Stöllberger, J Finsterer, W Lutz, C Stöberl, A Kroiss, A Valentin, and J Slany. Multivariate analysis-based prediction rule for pulmonary embolism. *Thrombosis Research*, 97(97):267–273, 2000.
- C.L Teigen, T.P Maus, P.F Sheedy II, C.M Johnson, A.W Stanson, and T.J Welch. Pulmonary embolism: diagnosis with electron-beam ct. *Radiology*, 188:839–845, 1993.
- G.D Tourassi, C.E Floyd, H.D Sostman, and R.E Coleman. Acute pulmonary embolism: Artificial neural networks approach for diagnosis. *Radiology*, 189:555–558, 1993.

- P.S Wells, D.R Anderson, and M Rodger et al. Derivation of a simple clinical model to categorize patients probability of pulmonary embolism: increasing the models utility with the SimpliRED D-dimer. *Thromb Haemost*, (83):416–420, 2000.
- PS Wells, J Ginsberg, DR Anderson, J Hirsh, AGG Turpie, J Bormanis, and C Kearon. The value of bilateral venous compression ultrasonography in patients with suspected pulmonary embolism (PE). *Blood*, (90):1303, 1997.
- P.S Wells, J.S Ginsberg, D.R Anderson, C Kearon, M Gent, A.G.G Turpie, J Bormanis, J Weitz, M Chamberlain, D Bowie, D Barnes, and J Hirsh. Use of a clinical model for safe management of patients with suspected pulmonary embolism. *Annals of Internal Medicine*, 129(129):997–1005, 1998.
- J Wicki, T.V Perneger, A.F Junod, H Bounameaux, and A Perrier. Assessing clinical probability of pulmonary embolism in the emergency ward. *Archives of Internal Medicine*, 161(1)(161):92–97, 2001.

# APPENDIX A

## Data Structure Formats

### A.1 Layout of User Structure

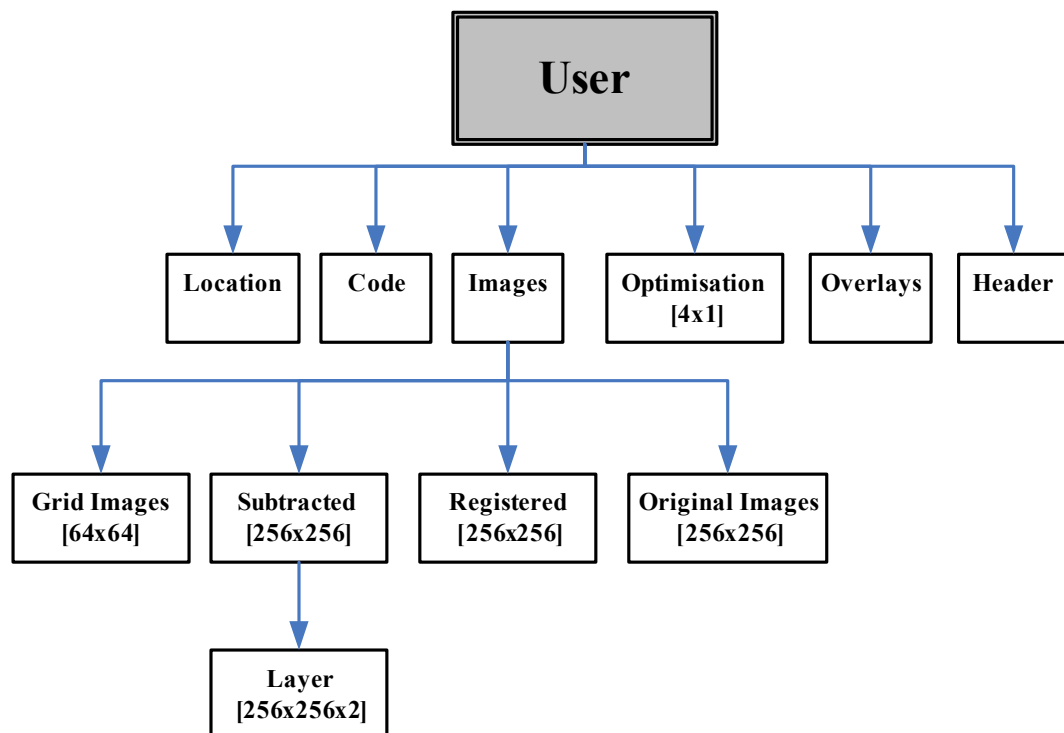


Figure A.1: Structure of User

## APPENDIX B

# Medical Image Viewer User Manual

This image viewing software is intended to enable future development of this work, and was not used in this project.

### B.1 Installation

Insert the CD-ROM into the appropriate drive and the setup program should start automatically. If the setup does not start, run the file *x:\setup.exe*, where *x:\* should be replaced with the correct drive letter for your CD-ROM.

Once the setup has started, it may install the *Microsoft .NET framework* depending on whether it is installed already. After that, just follow the instructions in the setup program.

### B.2 Initial Startup

When the program is run for the first time, the image database is extracted. The extraction could take from 1-5 minutes depending on the speed of your computer. The extraction is only run once.

## B.3 Basic Controls

Figure B.1 shows the layout of the *Medical Image Viewer*. Each numbered section will be explained below.

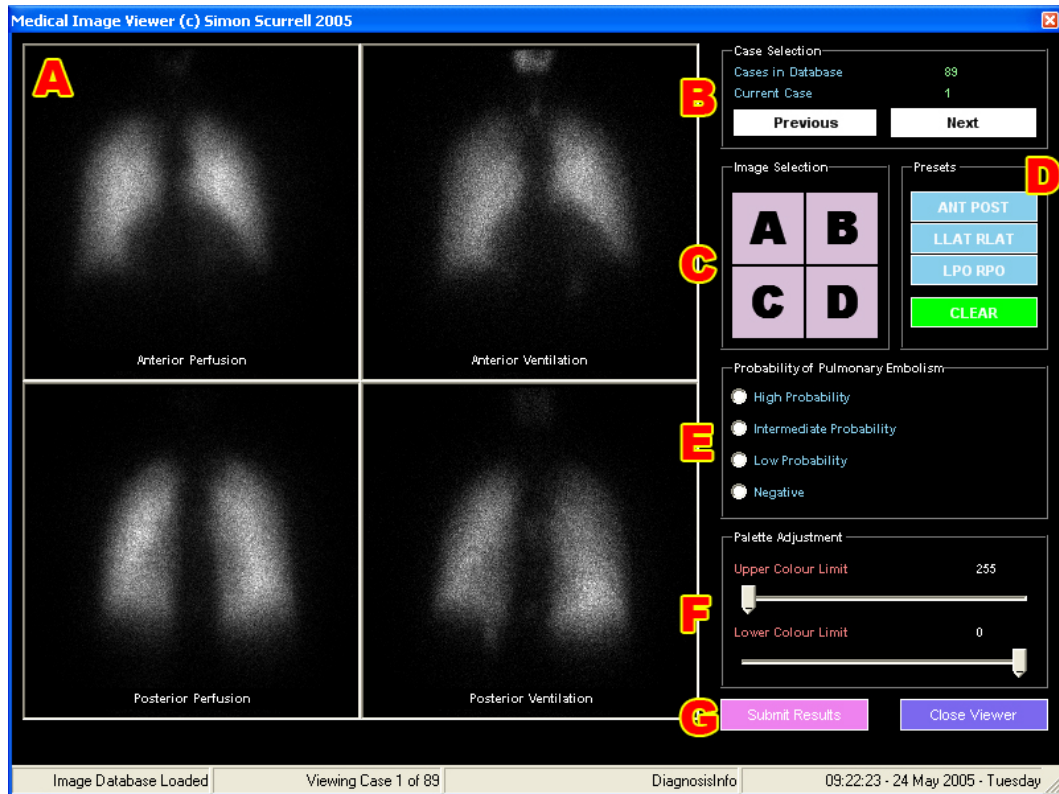


Figure B.1: Medical Image Viewer Screenshot

### B.3.1 Lung Viewing Windows

These four windows can show any of the 12 images contained in each case. When a case is first loaded the default view, moving clockwise from the top left is:

- Anterior Perfusion
- Anterior Ventilation
- Posterior Ventilation
- Posterior Perfusion

The tag at the bottom of each window indicates which image is being viewed in that window.

### **B.3.2 Case Selection**

The section provides information on how many cases are in the database, as well as which case is currently being examined. The *Previous* and *Next* buttons allow you to navigate through the cases quickly.

### **B.3.3 Image Selection**

The four buttons marked *A,B,C,D* represent the four lung viewing windows. Clicking any of these buttons allows you to choose which image of the case is shown in its respective viewing window.

### **B.3.4 Presets**

These buttons provide preset configurations for the images displayed in the viewing windows. Clicking for example, the *LPO RPO* button will quickly bring up the perfusion and ventilation oblique views. If you require more flexibility for image selection you can use the *Image Selection* buttons described in section B.3.3.

### **B.3.5 Probability of Pulmonary Embolism**

This is where the diagnosis is made. You can select one of the four options for each case. The diagnosis for each case is stored in memory until saved to the computer's hard drive.



### B.3.6 Palette Adjustment

This section allows the image palette to be adjusted. The lower colour limit can never be more than the upper colour limit. Usually, using only the upper colour limit is sufficient.

### B.3.7 Submit Results

## B.4 Saving Progress

This program has two methods for saving data. The first one is an autosave feature, which saves the data every few seconds after the first change is made. The second method is used when the user closes the program. If you have made any changes since the last save point, or a save point has not been created yet, you will be prompted to save your current progress. You can respond either yes or no. This prompt is shown in Figure B.2

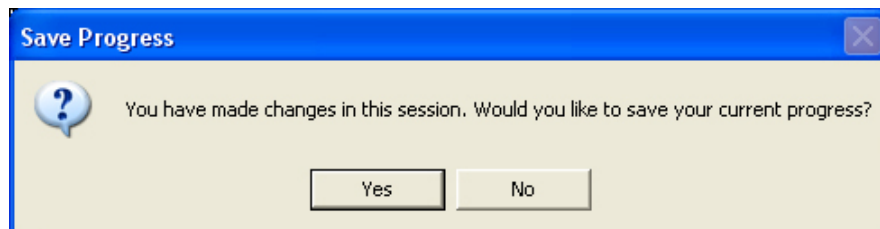


Figure B.2: Save progress

### B.4.1 Loading Progress

If your progress was saved previously, you will be asked if you would like to load your previously saved data.

### B.4.2 Recovery Using the *Autosave* Function

If for some reason, the program crashed or your current progress was not saved when the program exited, you can restore your data using the *autosave* feature. When

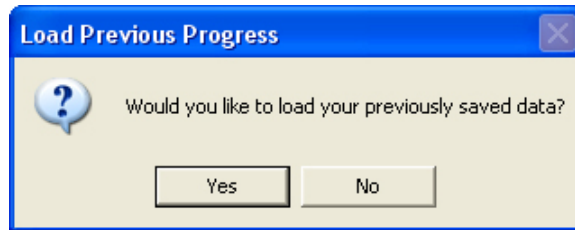


Figure B.3: Loading progress

the program is started you will see a prompt like the one given in Figure B.4.

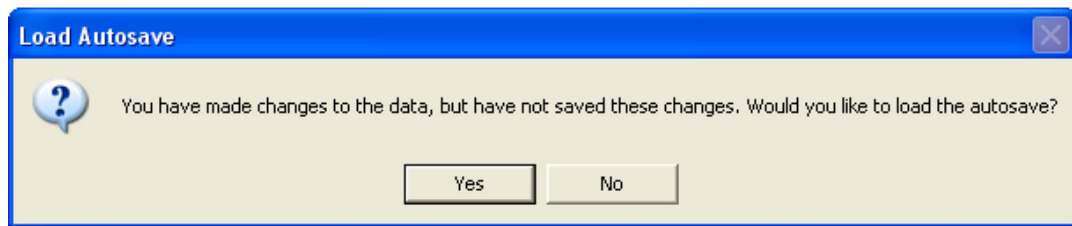


Figure B.4: Recover using autosave

The *autosave* feature only comes into effect when there is no other saved data or when the *autosave* data is more recent than the saved data.

## B.5 Submitting Results

After all the cases have been diagnosed, the results need to be submitted. To submit results click on the **Submit Results** button on the bottom left of the screen. You will then be prompted to enter your details as shown in Figure B.5.

A Windows-style dialog box titled "Submit Reporter Information" with a blue header bar and a close button in the top right. The main area has a dark background. At the top, it says "Please enter the following information". Below this are two text input fields: "Your Name" with the text "John Doe" and "Your E-mail Address" with the text "jdoe@mail.co.za". To the right of these fields are two buttons: "OK" and "Cancel". At the bottom, there is a "Notice" section with a text area containing the following text: "All information entered will be kept private. All information is contained with a password protected file. Your e-mail address will only be used for correspondence between yourself and myself should any problems arise."

Figure B.5: Entering user details

After your details have been entered, press the **OK** button. This will save the results to a file and you see something like Figure B.6.

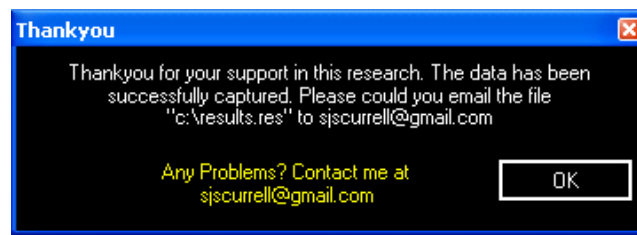


Figure B.6: Results success

The results are stored in a file called *results.res* which by default is stored in the root directory of the hard drive, usually *c:\*. This file, *c:\results.res* should be emailed to *sjscurrell@gmail.com*.

## APPENDIX C

### Statistical Formula Review

What follows is the definitions of a few important statistical formula used to determine the effectiveness of a specific prediction rule.

**TP** True Positive.

**TN** True Negative.

**FP** False Positive.

**FN** False Negative.

#### C.1 Sensitivity

Sensitivity is the probability the the test is positive when given to a group of patients with the condition.

$$S_n = \frac{TP}{(TP + FN)} \quad (\text{C.1})$$

Conditional probability form:

$$S_n = P(\text{Test Positive} | \text{Patient has condition}) \quad (\text{C.2})$$

## C.2 Specificity

Specificity is the probability the the test is negative when given to a group of patients without the condition.

$$S_p = \frac{TN}{(TN + FP)} \quad (C.3)$$

Conditional probability form:

$$S_p = P(\text{Test Negative} | \text{Patient does not have condition}) \quad (C.4)$$

## C.3 Positive Predictive Value

Positive Predictive Vale (PPV) is the probability the the patient has the condition when restricted to those patients who tested positive.

$$PPV = \frac{TP}{(TP + FP)} \quad (C.5)$$

Conditional probability form:

$$PPV = P(\text{Patient has condition} | \text{Test Positive}) \quad (C.6)$$

## C.4 Negative Predictive Value

Negative Predictive Vale (NPV) is the probability the the patient does not have the condition when restricted to those patients who tested negative.

$$PPV = \frac{TN}{(TN + FN)} \quad (C.7)$$

Conditional probability form:

$$PPV = P(\text{Test Negative} | \text{Patient does not have condition}) \quad (C.8)$$

## APPENDIX D

### PEDiag Flow Charts

Detailed flow charts showing the implementation of the critical functions are presented below.

#### D.1 Image Processing Functions

##### D.1.1 imhotspots.m

**Description:** The function *imhotspots.m* is used to remove local deposition of aerosol.

**Usage:** [output\_image] = *imhotspots*(image,size)

**Inputs:** *image*: input image requiring hotspot removal  
*size*: size of area used for interpolation [size x size]

**Outputs:** *image*: returns image

**Flow Chart:** Figure. D.1

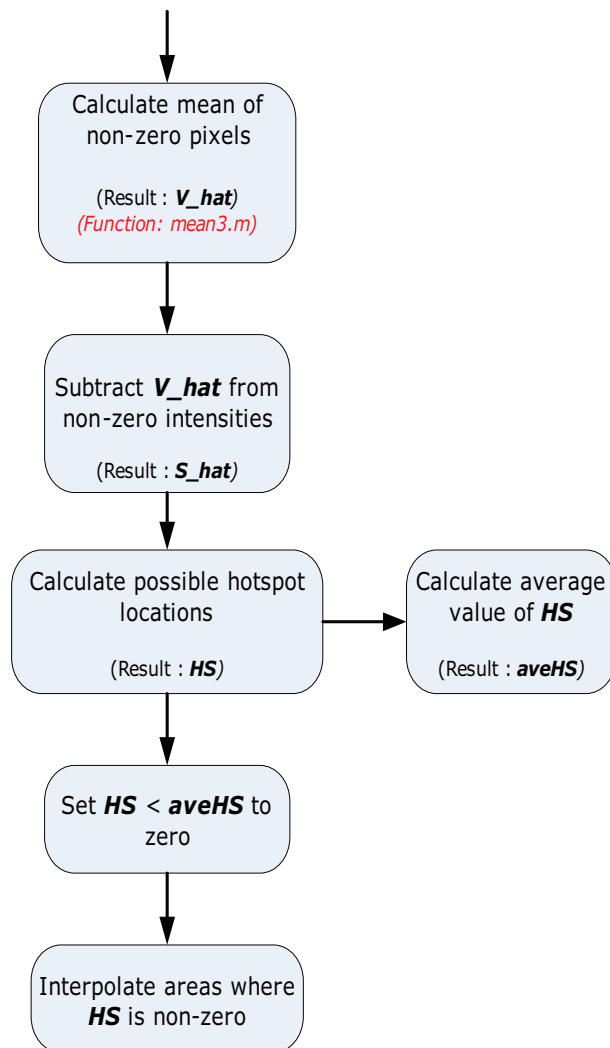


Figure D.1: imhotspots.m - Flow chart

### D.1.2 imfshs.m

**Description:** The function *imfshs.m* is used to normalize image intensity by stretching the images histogram.

**Usage:** output\_image = *imfshs*(input\_image)

**Inputs:** *image*: image requiring histogram stretching

**Outputs:** *image*: histogram stretched image

**Flow chart:** Figure. D.2

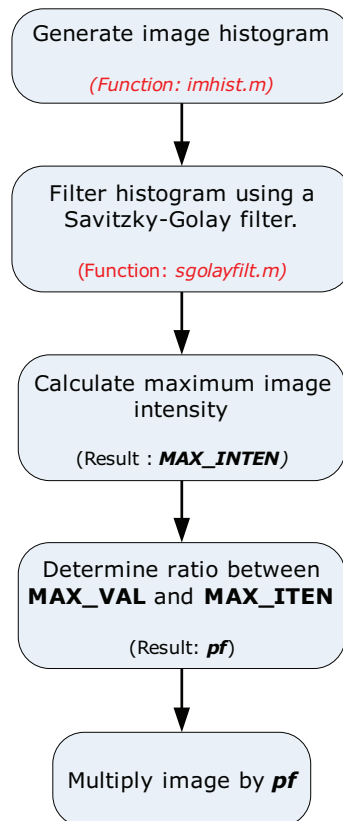


Figure D.2: imfshs.m - Flow chart



### D.1.3 imsegment.m

**Description:** The function *imsegment.m* is used to remove unwanted information from the V/Q scan.

**Usage:** output\_image = *imsegment*(image,method,thresh,art,vent)

**Inputs:** *image*: input image requiring segmentation  
*method*: selects which segmentation method to use  
*thresh*: choose threshold  
*art*: remove artifacts from image  
*vent*: choose if the image is a ventilation image

**Outputs:** *image*: returns segmented image

**Flow Chart:** Figure. D.3

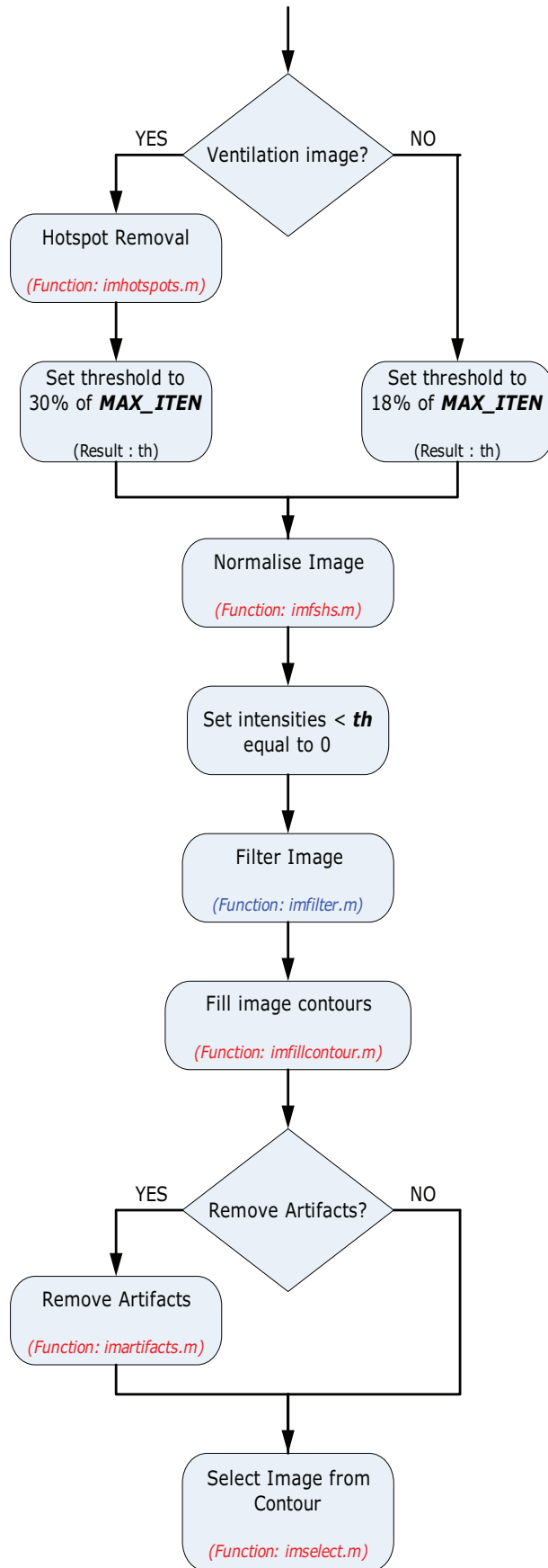


Figure D.3: `imsegment.m` - Flow chart

#### D.1.4 imartifacts.m

**Description:** The function *imartifacts.m* is used remove image artifacts after segmentation

**Usage:** [output\_image] = *imartifacts*(work\_image,thresh)

**Inputs:** *work\_image*: image that requires artifact removal  
*thresh*: the threshold ratio for artifact classification

**Outputs:** *output\_image*: returns image with artifacts removed

**Flow chart:** Figure. D.4

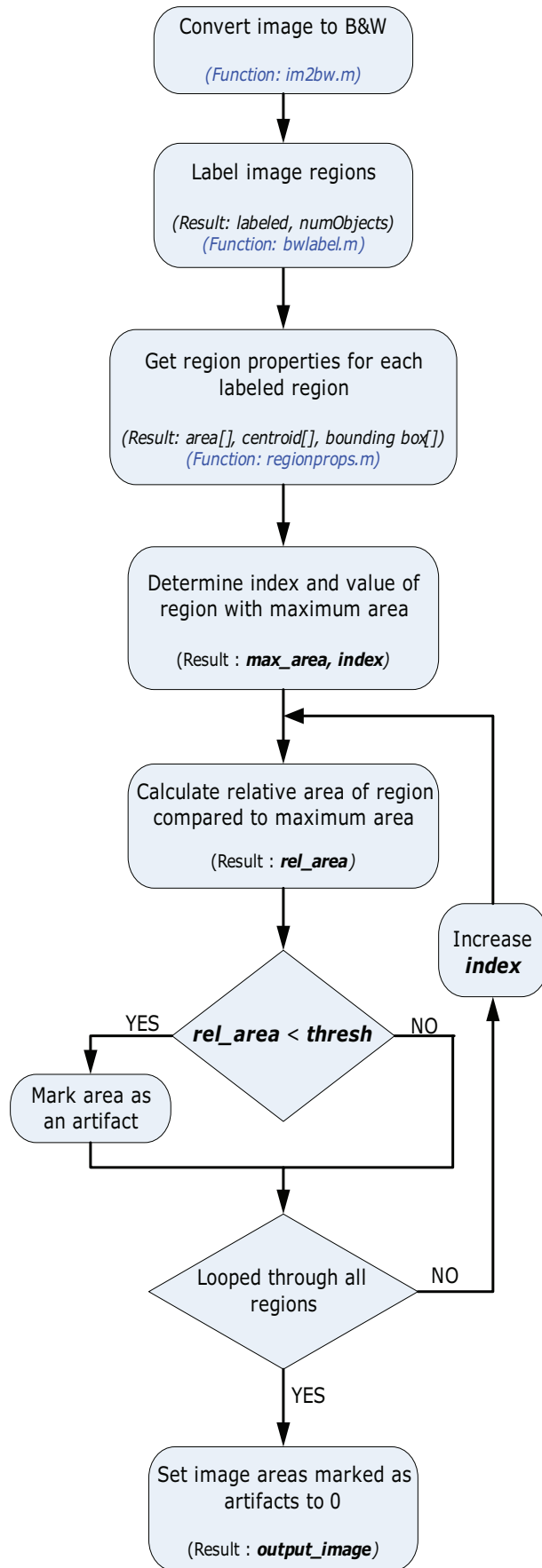


Figure D.4: imartifacts.m - Flow chart

### D.1.5 imfillcontour.m

**Description:** The function *imfillcontour.m* performs contour extraction

**Usage:** `output_image] = imfillcontour(work_image,fill);`

**Inputs:** *work\_image*: image on which contour extraction is performed  
*fill*: option specifying if "holes" should be filled

**Outputs:** *output\_image*: image after contour extraction.

**Flow chart:** Figure. D.5

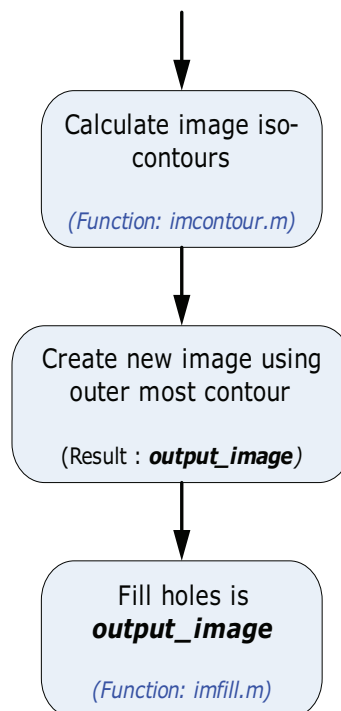


Figure D.5: imfillcontour.m - Flow chart

### D.1.6 imgenerategrid.m

**Description:** The function *imgenerategrid.m* converts an image into and grid of specified size

**Usage:** [grid] = *imgenerategrid*(work\_image,border,size)

**Inputs:** *work\_image*: image to convert into a grid image  
*border*: structure which defines bounding rectangle of image  
*size*: specifies the size of the grid image

**Outputs:** *grid\_image*: transformed image

**Flow chart:** Figure. D.6

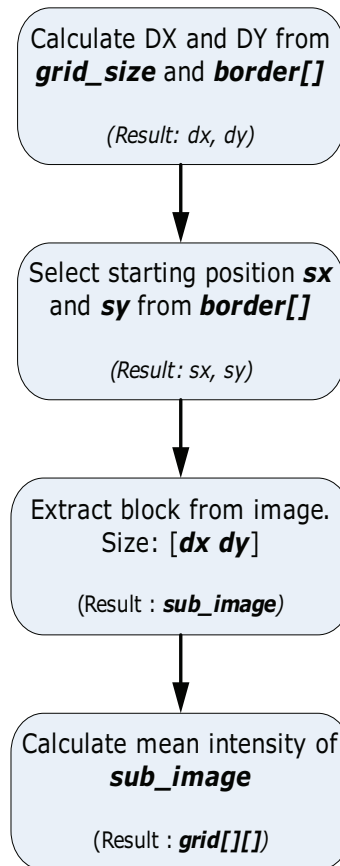


Figure D.6: imgenerategrid.m - Flow chart

### D.1.7 imgrid.m

**Description:** The function *imgrid.m* converts all 6 subtracted images into grid images

**Usage:** [newUser] = *imgrid*(User,grid\_size)

**Inputs:** *User*: user structure  
*grid\_size*: specifies the size of the grid

**Outputs:** *newUser*: new user structure containing grid images

**Flow chart:** Figure. D.7

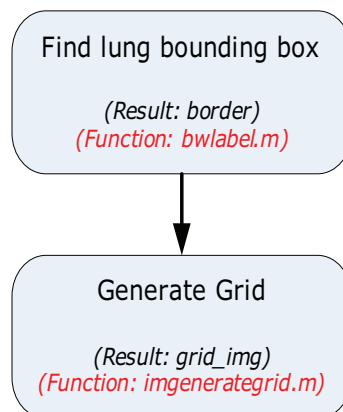


Figure D.7: imgrid.m - Flow chart

### D.1.8 imquotient.m

**Description:** The function *imquotient.m* generates quotient images

**Usage:** [newUser] = *imquotient*(User)

**Inputs:** *User*: user structure

**Outputs:** *newUser*: new user containing quotient images

**Flow chart:** Figure. D.8

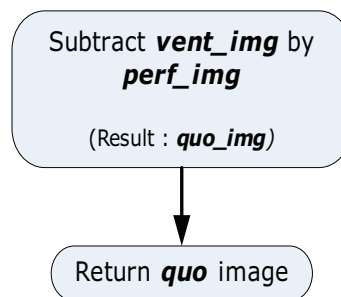


Figure D.8: imquotient.m - Flow chart



### D.1.9 imselect.m

**Description:** The function *imselect.m* selects a specific region of an image based on a binary selection mask

**Usage:** [output\_image] = *imselect*(work\_image,selection\_mask)

**Inputs:** *work\_image*: image from which selection is made  
*selection\_mask*: binary mask

**Outputs:** *output\_image*: selected region of image

**Flow chart:** Figure. D.9

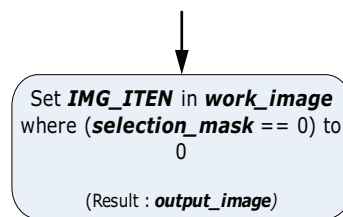


Figure D.9: imselect.m - Flow chart

## D.2 Image Alignment Functions

### D.2.1 `imalignsub_ga.m`

**Description:** The function *imalignsub\_ga.m* is used to align the ventilation and perfusion images.

**Usage:** `[output_image, params] = imalignsub_ga(target_image,ref_image,display)`

**Inputs:** *target\_image*: image that *ref\_image* aligns itself to  
*ref\_image*: image that needs to be transformed  
*display*: option to display alignment graphically

**Outputs:** *output\_image*: returns the aligned and subtracted image  
*params*: returns the optimised scale and rotation parameters

**Flow chart:** Figure. D.10

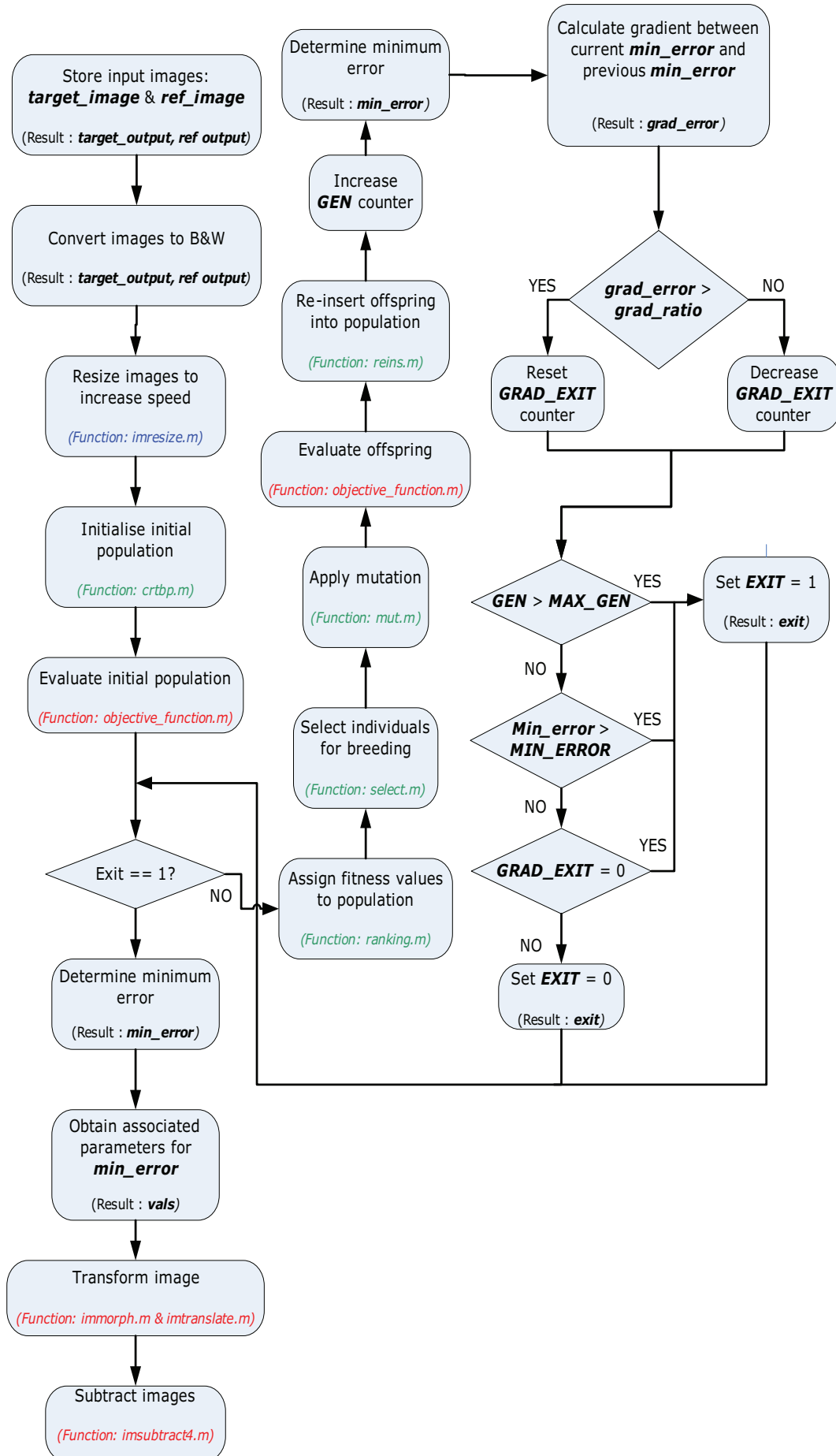


Figure D.10: imalignsub-ga.m - Flow chart

### D.2.2 `objective_function.m`

- Description:** The function *objective\_function.m* is used to calculate the error during optimisation
- Usage:** `[ObjValue] = objective_function(PhenData,tar_img,ref_img,display)`
- Inputs:**
- PhenData*: structure containing possible solutions
  - tar\_image*: target image
  - ref\_image*: reference image
  - display*: option to show alignment graphically
- Outputs:** *ObjValue*: array containing error values associated with possible solutions
- Flow chart:** Figure. D.11

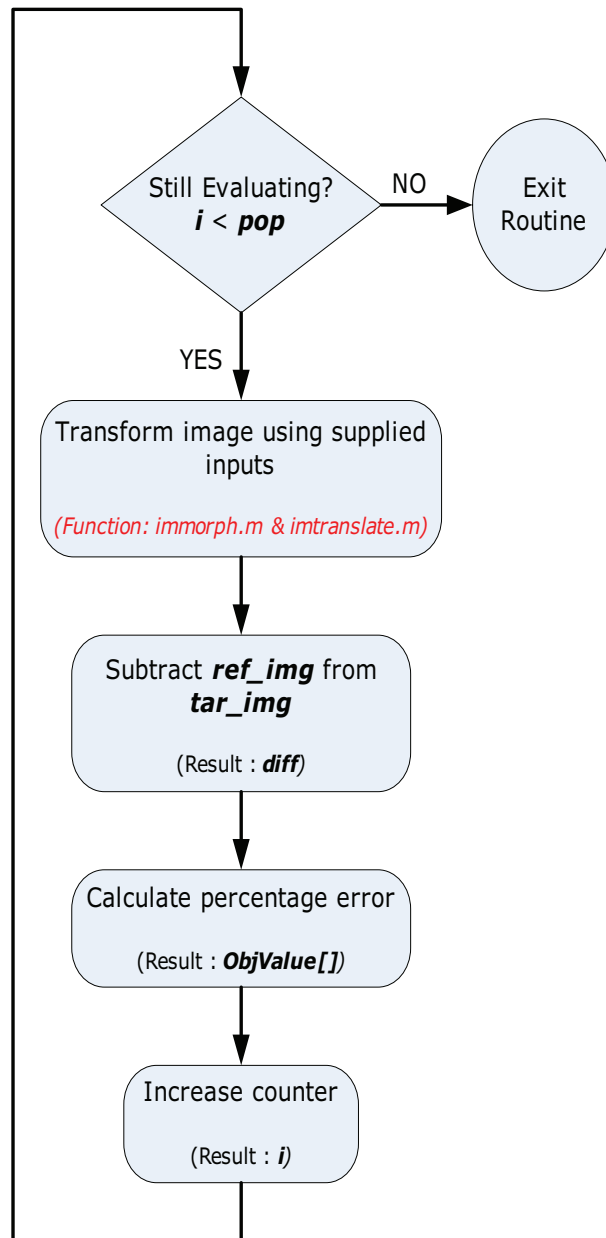


Figure D.11: `objective_function.m` - Flow chart

## D.3 Feature Extraction and Classification Functions

### D.3.1 transform\_inputs.m

**Description:** The function *transform\_inputs.m* applies a PCA transformation matrix to the inputs. The function also selects the best inputs based on the training data set.

**Usage:** [outputs] = *transform\_inputs*(User,var,type,nin)

**Inputs:** *User*: user structure  
*var*: the amount of retained variability (from PCA analysis).  
*type*: the type of image. Either 'sub' or 'quo'.  
*nin*: the number of inputs to select. 10, 20 or 30.

**Outputs:** *outputs*: the transformed inputs

**Flow chart:** Figure. D.12

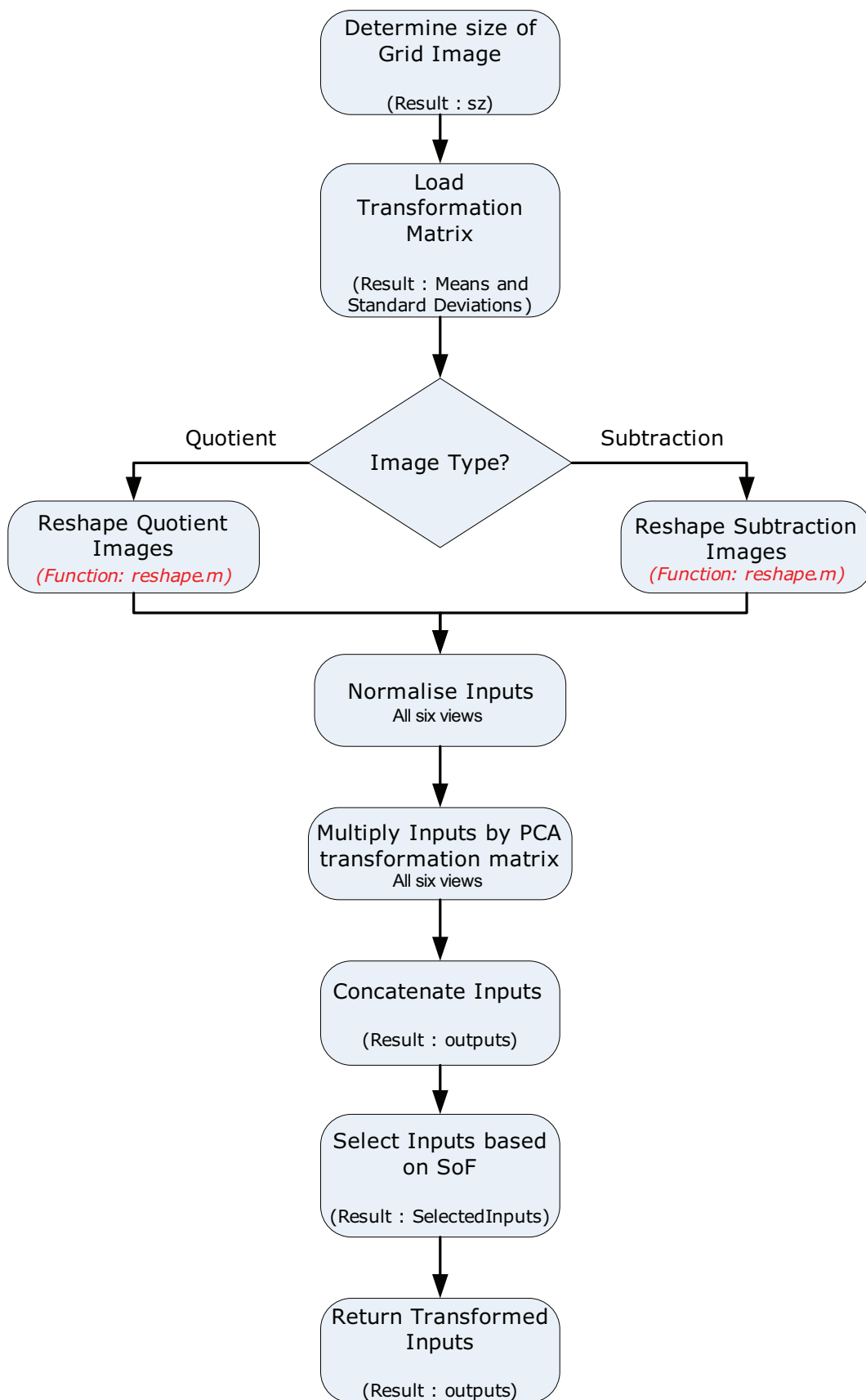


Figure D.12: transform\_inputs.m - Flow chart

### D.3.2 `classify_inputs.m`

|                     |                                                                                                                                                                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description:</b> | The function <i>classify_inputs.m</i> takes a set of transformed inputs and classifies them using the selected network. The neural network is selected based on the input parameters.                                                                             |
| <b>Usage:</b>       | <code>[pred,up,low,conf] = <b>classify_inputs</b>(inputs,var,type,sz)</code>                                                                                                                                                                                      |
| <b>Inputs:</b>      | <i>inputs</i> : the transformed inputs from <i>transform_inputs</i> .<br><i>var</i> : the amount of retained variability (from PCA analysis).<br><i>type</i> : the type of image. Either 'sub' or 'quo'.<br><i>sz</i> : the size of the grid image that was used. |
| <b>Outputs:</b>     | <i>pred</i> : the prediction of PE.<br><i>up</i> : the upper limit of the 95% confidence interval.<br><i>low</i> : the lower limit of the 95% confidence interval.<br><i>conf</i> : the confidence measure of the prediction.                                     |
| <b>Flow chart:</b>  | Figure. D.13                                                                                                                                                                                                                                                      |



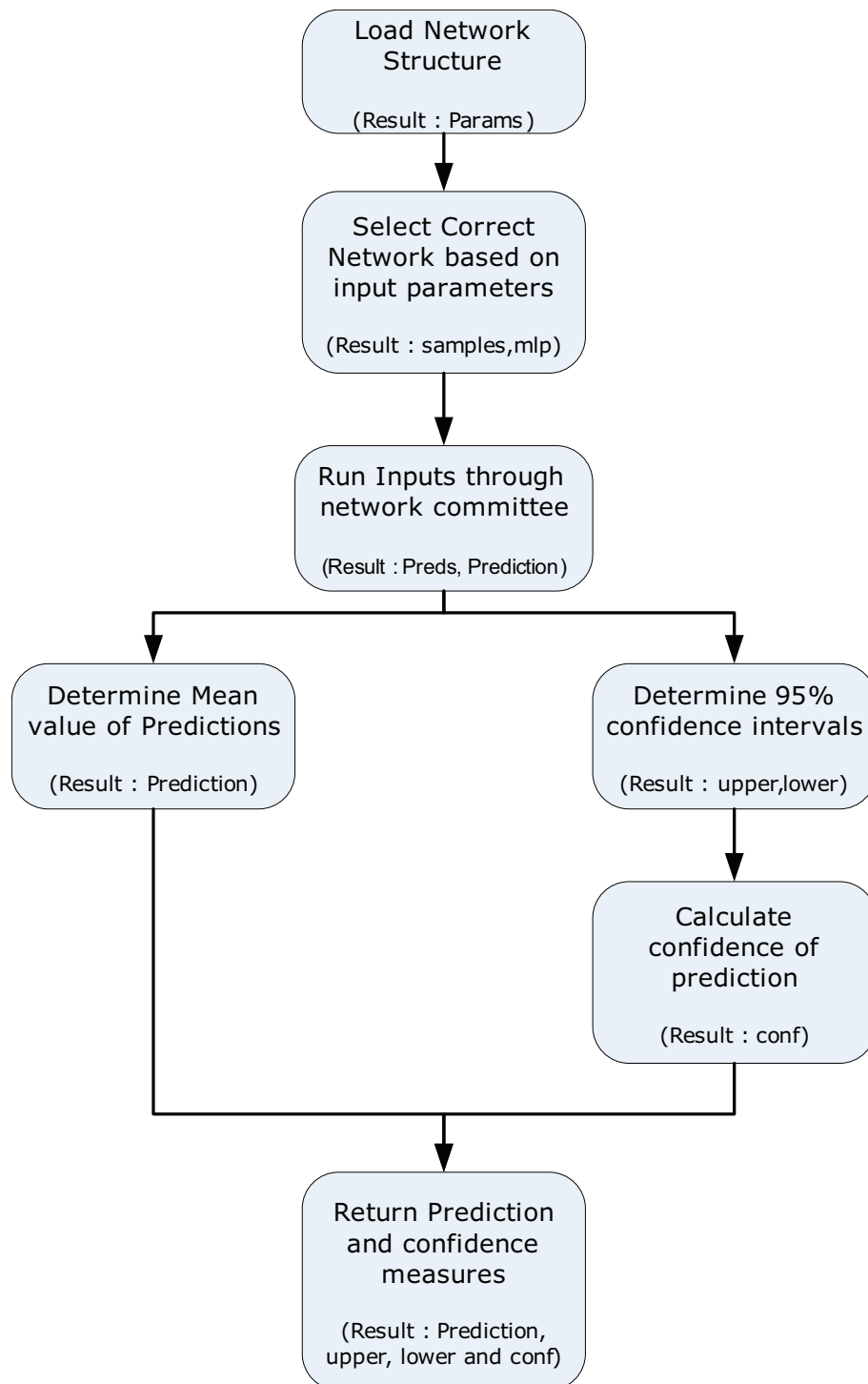


Figure D.13: classify\_inputs.m - Flow chart

## D.4 Miscellaneous Functions

### D.4.1 writecase.m

**Description:** The function *writecase.m* stores a user structure

**Usage:** `[] = writecase(User,status)`

**Inputs:** *User*: user structure  
*status*: specifies if case is a new case

**Outputs:** none

**Flow chart:** Figure. D.14

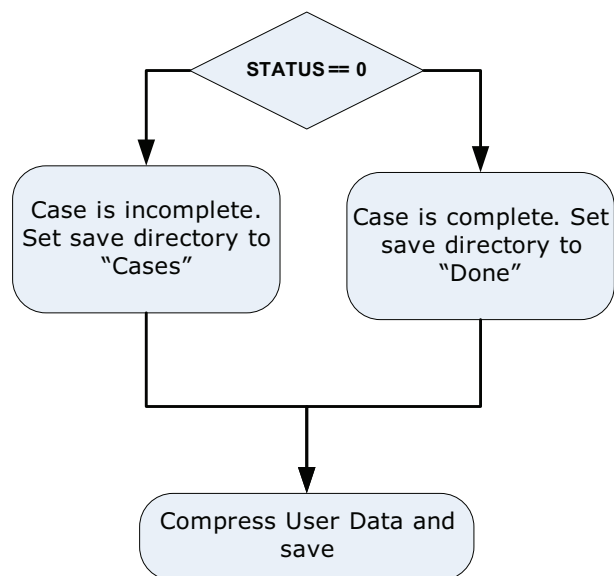


Figure D.14: writecase.m - Flow chart