Performance Technical Report with SF-298

Feasibility Studies of Nearest Neighbor Residual Vector Quantizer Classifiers for a Collection of Signal and Sensor Waveforms

Prepared by

Christopher F. Barnes, PhD
Sensors and Electromagnetics Applications Laboratory
Georgia Tech Research Institute
Georgia Institute of Technology
Atlanta, GA 30332–0857

Prepared for

Douglas E. Lake, PhD, ONR 313
Office of Naval Research
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217–5660

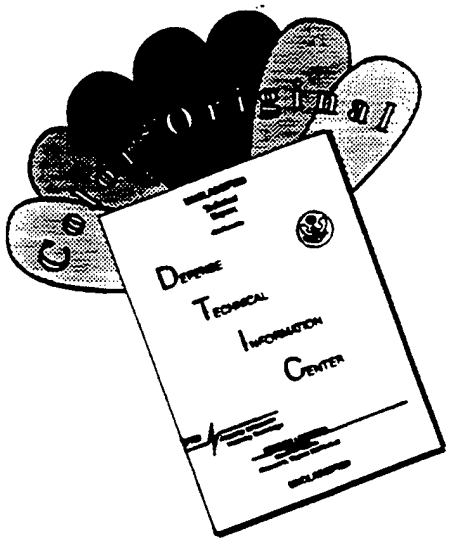
ONR Grant No: N00014–96–1–0308
GTRI Project No: A–5138–000

January 1, 1997

GEORGIA INSTITUTE OF TECHNOLOGY
A Unit of the University System of Georgia
Atlanta, Georgia 30332

Georgia Tech
RESEARCH INSTITUTE

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1 January 1997	Interim Report 1-1-96 through 1-1-97

4. TITLE AND SUBTITLE
Feasibility Studies of Nearest Neighbor Residual Vector Quantizer Classifiers for a Collection of Signal and Sensor Waveforms

5. FUNDING NUMBERS
ONR Grant No:
N00014-96-1-0308

6. AUTHOR(S)
Christopher F. Barnes

7. PERFORMING ORGANIZATION NAME (S) AND ADDRESS(ES)
Georgia Tech Research Institute
Georgia Institute of Technology
Atlanta, Georgia 30332

8. PERFORMING ORGANIZATION REPORT NUMBER
A-5138-000

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Douglas E. Lake ONR 313, ONR
Ballston Centre Tower One
800 N Quincy Street
Arlington, VA 22217-5600

10. SPONSORING/MONITORING AGENCY REPORT

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Preliminary results obtained from using direct sum successive approximations for joint compression and classification of several sensor types are reported.

14. SUBJECT TERMS
compression classification

15. NUMBER OF PAGES
107

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Acknowledgment

This report summarizes the current status of ongoing work in a Georgia Tech Research Institute (GTRI) research project entitled, *"Feasibility Studies of Nearest Neighbor Residual Vector Quantizer Classifiers for a Collection of Signal and Image Sensor Waveforms."* This research is supported by an Office of Naval Research (ONR) grant, number N00014–96–1–0308, and performed under GTRI project number A–5138–000. This interim Performance Technical Report covers the first year of this three year grant, and reports on work performed between 1 January 1996 and 31 December 1996.

The ONR technical project director is Dr. Douglas E. Lake, ONR 313. The principal investigator (PI) is Dr. Christopher F. Barnes of GTRI. The principal investigator expresses sincere appreciation to ONR for its support and, in particular, to Dr. Lake for his guidance and oversight in the execution of this research program.

Data Source Credits

The data used in this research were obtained from various sources. The following credits and acknowledgments apply:

The sidescan sonar data were provided courtesy of the Coastal Systems Station, Dahlgren Division, Naval Surface Warfare Center, Code R12, Panama City, Florida. These data were collected under funding from the Office of Naval Research (Dr. Wally Ching, ONR 321T).

The acoustical backscatter data were provided courtesy of the Coastal Systems Station, Dahlgren Division, Naval Surface Warfare Center, Panama City, Florida.

The NRaD pump data were obtained from Dr. Lake's home page on the world wide web. These data sets have been extensively analyzed in prior research by Jeff Allen of the Research, Development, Test & Evaluation Division (NRaD) at the Naval Command, Control, and Ocean Surveillance Center (NCCOSC) in San Diego.

The electro-optical images were provided courtesy of Rome Laboratories of the United States Air Force.

The digitized mammogram images were provided courtesy of the National Expert and Training Centre for Breast Cancer Screening and the Department of Radiology at the University of Nijmegen, the Netherlands.

Contents

List of Figures

List of Tables

Chapter 1

Research Project Overview

This Performance Technical Report is organized as follows.

Chapter 1 explains the problem addressed by this research project and provides an overview of the grant purpose, problem statement, proposed solution, research objectives and technical approach.

Chapter 2 describes the progress achieved to date for each of the different sensor types that are of interest in this research project.

Chapter 3 gives the research plans for the next year for each of the sensor types of interest.

Appendices A, B and C elaborates technical details in order for this document to serve as a self-contained reference. These appendices also contain example DSSA classification results and a glossary of abbreviations.

1.1 Grant Purpose

The purpose of this ONR grant is to support the evaluation of the performance of a particular joint compression/classification algorithm called *nearest neighbor residual vector quantizer (NN-RVQ) classification* on data obtained from a variety of sensors and applications. NN-RVQ is based on a new mathematical development called *direct sum successive approximations (DSSA)*. DSSA is a technology invented[1] by the principal investigator that can be used for *vector quantization (VQ)* and pattern recognition. DSSA uses a data decomposition process to construct structured *codevectors* and *templates* that can be efficiently searched (in terms of computation and memory) in *vector quantizers* used for data compression, and in nearest neighbor pattern recognition algorithms. The purpose of this grant is to assess the performance of NN-RVQs when they are used for joint compression and classification of various types of sensor data.

[1]Patent No. 5,250,949, held in part by the principal investigator, and assigned to Brigham Young University, with derived intellectual property held by the Georgia Institute of Technology.

1.2 Problem Statement

There are two underlying problems addressed by this research:

1. **The Data Compression Problem**

   The lack of sufficient bandwidth required to transmit data at a high rate from a remote sensor to the data user, or equivalently, the lack of sufficient computer memory required to store large volumes of measured sensor data.

2. **The Onboard Data Classification Problem**

   The excessive computational resources required for real time classification of data onboard sensor platforms. The need for real time classification is motivated by the requirements for an onboard prescreen capability, an onboard target recognition capability, or an onboard machine fault detection capability.

1.3 Proposed Solution

The proposed solution is to use a technique recently developed for the mathematical decomposition and synthesis of sensor data in a joint compression and classification algorithm. The new technique is called *direct sum successive approximation.* DSSA can be used in a type of data compression process called *residual vector quantization (RVQ)*, and in a type of data classification process called *nearest neighbor residual vector quantizer classification.*

1.4 Research Objectives

The primary object of this research is to determine the feasibility of performing joint compression and classification on a variety of image and signal sensors using the DSSA algorithm in nearest neighbor residual vector quantization classifiers. This contract has been structured by ONR to permit flexibility as to exactly what sensors, data, and applications are evaluated by GTRI. The use of DSSA for classification does not require feature extraction[2]. DSSA may be incorporated into NN-RVQ classifiers in such a way that direct classification of data samples is possible. DSSA classifiers can be easily designed for a wide variety of data types—all that is required is sample data of clutter and targets, or sample data of normal and faulty machine state measurements. Thus the ease at which DSSA classifiers can be designed and implemented allows a wide variety of sensors and associated data to be investigated in a cost efficient manner.

[2]This does not preclude the use of a DSSA system as a conventional discriminate that operates on a set of extracted data features.

The specific objectives and tasks of this research program are to obtain data and evaluate joint DSSA compression/classification on a variety of signal and image data sources. Candidate sensors include defense related imagery and signal data. Dual-use applications such as computer assisted diagnosis and compression of medical mammography image data are also candidates for study. GTRI will continue to be responsive to any directive from ONR as to which data sets should be tested and evaluated in this research project (to the extent that funding levels and data availability permit). The following list of candidate sensor types have been identified, and research into DSSA processing of each of these types of data has progressed to varying degrees at this point in this multi-year research program.

Sidescan Sonar Images

Sidescan sonar images are used for sea mine detection. A previous project[3] established the feasibility of using DSSA for sea mine detection in sonar imagery. An objective of this research is to investigate the feasibility of joint compression and classification of sidescan sonar images communicated from remote *unmanned underwater vehicles (UUV)* to a receiving surveillance base station.

Acoustical Backscatter Signals

The use of acoustical backscatter has been proposed as a simple means of sea mine detection. Another concurrent research program[4] is investigating the feasibility of using DSSA for classifying backscatter data. An objective of this research is to explore the possibility of using DSSA for joint compression and classification of acoustical time series data.

Machine State Status Signals

The use of machine state status signals has been proposed for determining the need for condition-based maintenance (as opposed to scheduled maintenance). Examples of such data include vibration and motor current time series data from sources such as helicopter gear boxes, fire control and condensate pumps. An objective of this research is to investigate the feasibility of classifying machine state time series data and explore DSSA data compression possibilities for such signals.

---

[3]Contract N61331-93-K-0005, *"An Initial Evaluation of a New Approach to High Resolution Sonar Imagery,"* issued to GTRI by the Coastal Systems Station, Dahlgren Division of the Naval Surface Warfare Center.

[4]Contract N61331-96-C-0027, *"Continued Development of Unique Mathematical and Statistical Signal Processing Algorithms for Detecting and Classifying Mines in Acoustic Backscatter,"* issued to GTRI by the Coastal Systems Station, Dahlgren Division of the Naval Surface Warfare Center.

Synthetic Aperture Radar Images

Synthetic aperture radar (SAR) images provide ground and sea surface surveillance regardless of day, night and weather conditions. An objective of this research is to investigate the joint classification/compression of SAR data for target detection and recognition. The results of this research would be most applicable to remote sensing from unmanned aerial vehicles (UAV) launched from either ground or ship base stations.

Electro-Optical Images

Possible classification tasks associated with electro-optical images include target detection and land-use classification. An objective of this research is to investigate the joint compression/classification of electro-optical (EO) images for one or more defense or dual-use applications.

Infrared Images

Possible classification tasks associated with infrared images include target detection. An objective of this research is to investigate the joint compression/classification of infrared (IR) images for one or more applications.

Multispectral Images

Possible classification tasks associated with multispectral images include target detection and land-use classification. An objective of this research is to investigate the joint compression/classification of multispectral (MS) images.

X-ray Images

A key to surviving breast cancer is early detection. Mammography X-rays are used to detect microcalcifications, which are small (sometimes as small as 1/10th of a millimeter) calcium deposits that may be an early indicator of malignant tumor growth. However, not all microcalcifications are detected by radiologists because of imperfections and limitations of the human visual system. The proposed solution is to use a computer and a microcalcification detection algorithm based on DSSA to cue radiologists to regions of the mammogram that likely contain microcalcifications[5]. Since DSSA can also provide compression, an objective of this research is to determine the feasibility of joint compression/classification of mammograms for storage in medical picture archival systems (PACs).

[5]This research was supported in part by GTRI internal research and development funds.

1.5 Technical Approach

The following technical steps are being used to achieve the research objectives for each sensor type selected for testing and evaluation.

Task 1: Data Acquisition

The first step is to acquire sample data—this presents a challenge in some cases. The availability of sufficient amounts of sensor data and associated truth data (e.g., target locations, machine state, etc.) has, and will, continue to influence the choice of which sensors are tested in this research program.

Task 2: Test Plan Definition

The second step is to develop an acceptable test plan for the selected sensor data type for evaluating NN-RVQ performance. These test plans are being developed using procedures acceptable to the Navy. Possible methods include 1) the use of separate training and testing subsets of the sensor data, and 2) the use of the entire sensor data set for training, and then the use of alternative means in order to estimate NN-RVQ performance. These alternatives include analytical estimates, bootstrapping, test sets where simulated noise has been added to the sensor data, etc. Care should be taken to not over train the classifiers or vector quantizers, since over training can bias measured performance results.

Task 3: DSSA Implementation

The third step is the adaptation of existing DSSA software developed by GTRI for implementation of compression and classification algorithms for each sensor of interest. This adaptation is rather straightforward, all that is required is modification of the software input/output structures. This illustrates an advantage of the DSSA approach to pattern recognition—feature definition and extraction are not required[6] in the classifier design process. What is required, however, is a training process that uses data that has been preclassified (by human experts) for the generation of structured DSSA templates. Thus, DSSA has two modes of operation: an *off-line training mode* for template generation and an *on-line classification mode* for nearest neighbor pattern recognition.

Task 4: Template/Codevector Design

The fourth step is to select the key DSSA classifier parameters that are necessary for DSSA implementation. These parameters include template (codevector) block size, the number of

[6]DSSA may be used with classification systems that do use extracted features.

DSSA stages, and the number of templates (codevectors) at each stage. Once reasonable (not necessarily optimal) choices have been made for each these parameters, the data acquired in Task are is used to form training data in accordance with the test plan of Task 2. This training data are then used to generate vector quantizer codevectors and NN-RVQ classifier templates.

Task 5: Performance Evaluation

The fifth step is to evaluate performance of the DSSA system in accordance with the test plan defined in Task 2.

Task 6: Conclusions and Feasibility Assessment

The sixth step is to assess the performance results within the context of implementation complexity. Implementation complexity is loosely quantified in this report as the number of multiply-adds and memory locations required for the DSSA implementation. Feasibility is judged if acceptable probability of detection rates and sufficient compression ratios are obtained at "reasonable" implementation costs.

Chapter 2

Current Research Results

GTRI has obtained, to date, various amounts of six different types of sensor data: EO image data, X-ray image data, side scan sonar image data, SAR image data, acoustical backscatter signal data, and machine state signal data. However, some of these data sets are subject to certain use restrictions that GTRI is in the process of clarifying and resolving. GTRI has not obtained, to date, any data for infrared or multispectral image sensors.

The current progress, status and experimental results are summarized for each of these sensors in the following sections.

2.1 Electro-Optical Images

Task 1: Data Acquisition—Initiated

In the data classification problem, several distinct classes or states of nature are known to exist. Data influenced by these states are measured in an imprecise, noisy manner. The problem is to correctly estimate the underlying state. DSSA provides a novel data representation process that generates a successive approximation structure for sets of image data grouped together into small blocks of pixels [1]. This research seeks to integrate DSSA classification systems with DSSA compression systems. One type of sensor data of interest is EO-imagery.

Figure 2.1 is an original aerial EO image of the Moffet Naval Air Station close to San Francisco. Figures 2.2–2.4 show a DSSA generated representation sequence for this EO image. These three figures show DSSA approximations to the original at various compression rates. Note that it is reasonable to expect that the different airplanes on the airfield would be recognized by a human observer at various compression ratios due to differences in the corresponding levels of fidelity.

The possibility of using successive approximation data representations in a automated recog-

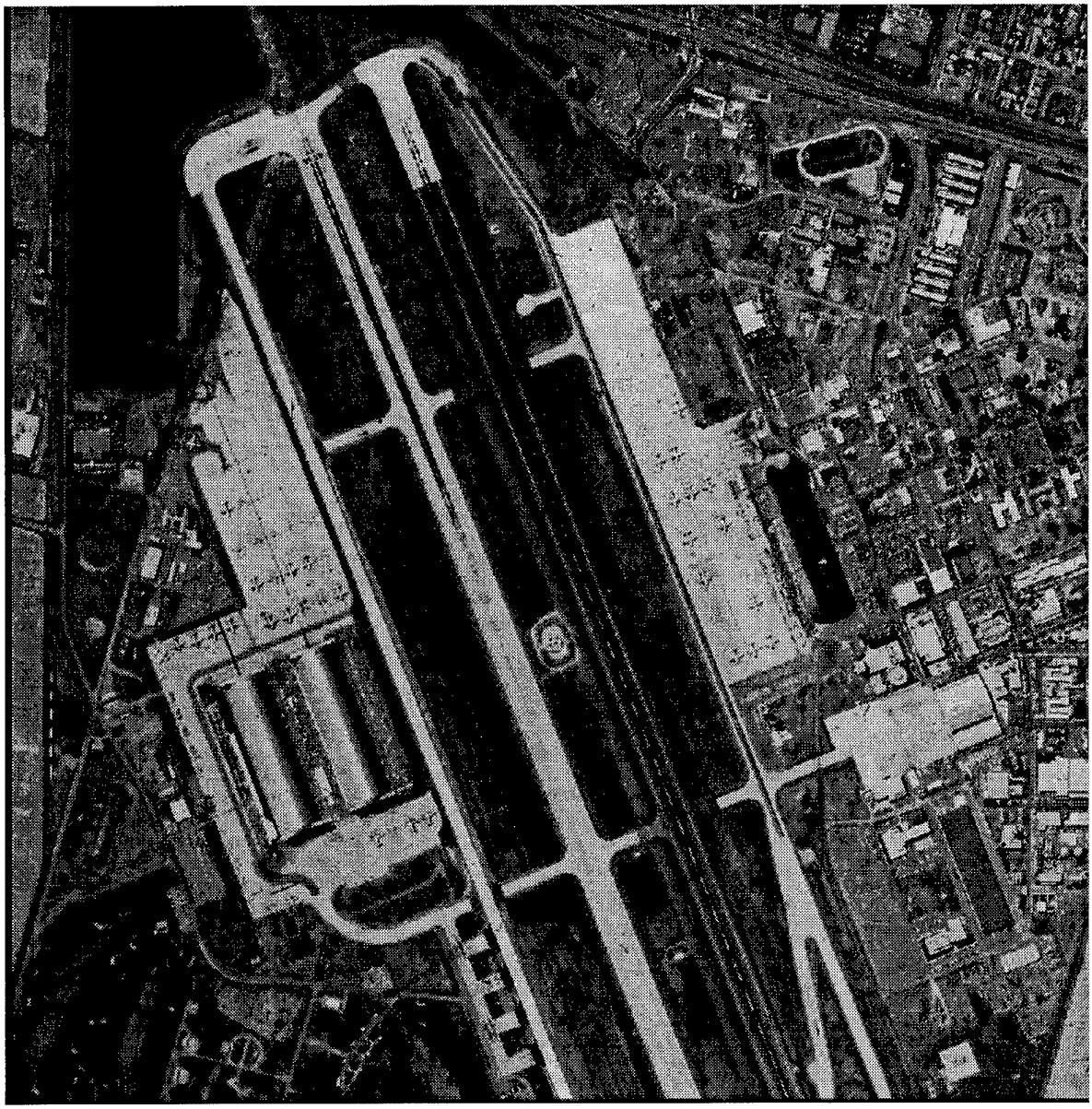


Figure 2.1: Original Moffet Naval Air Station EO image.

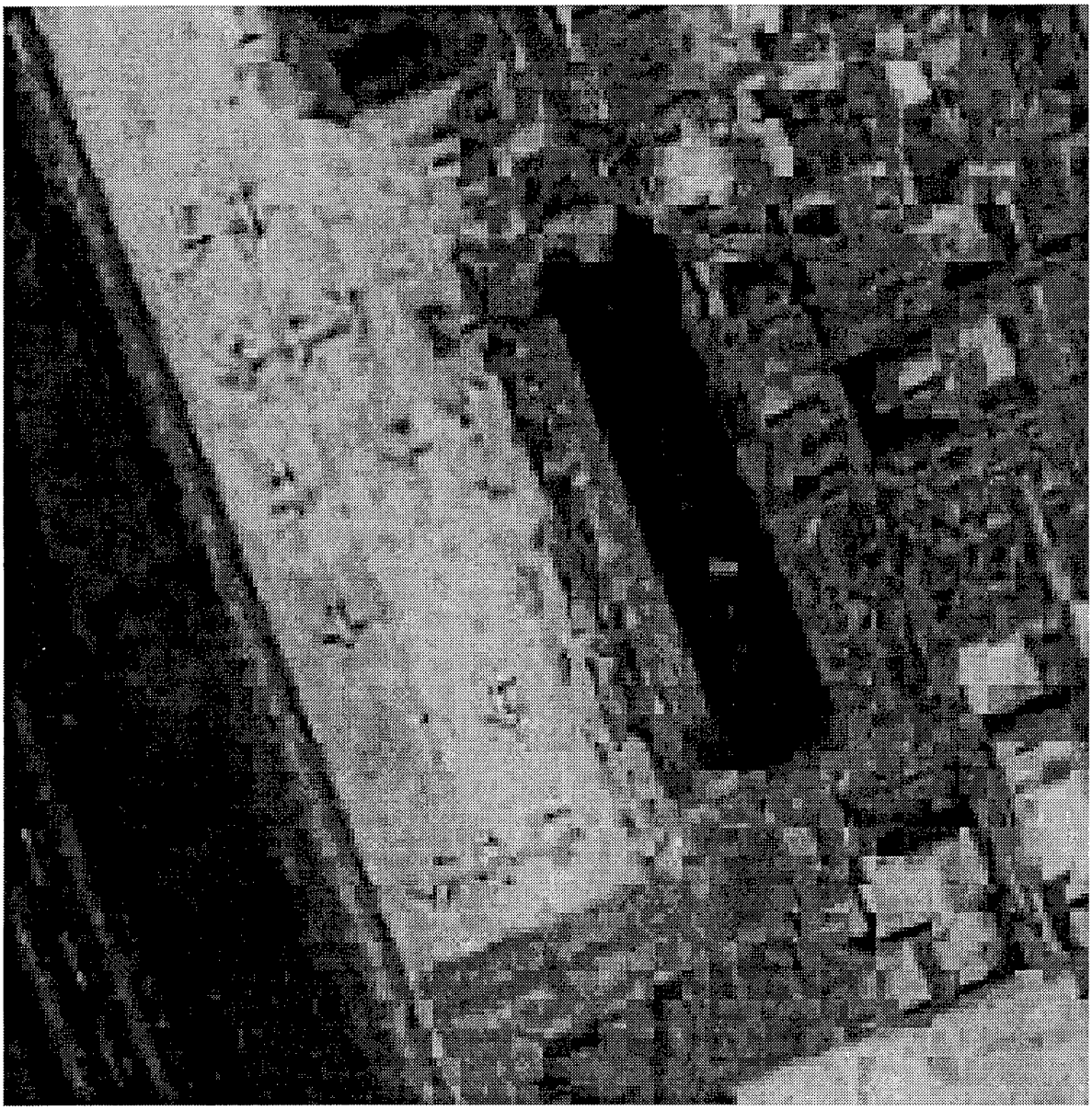


Figure 2.2: Portion of Moffet Naval Air Station EO image severely compressed with DSSA to a low data rate.

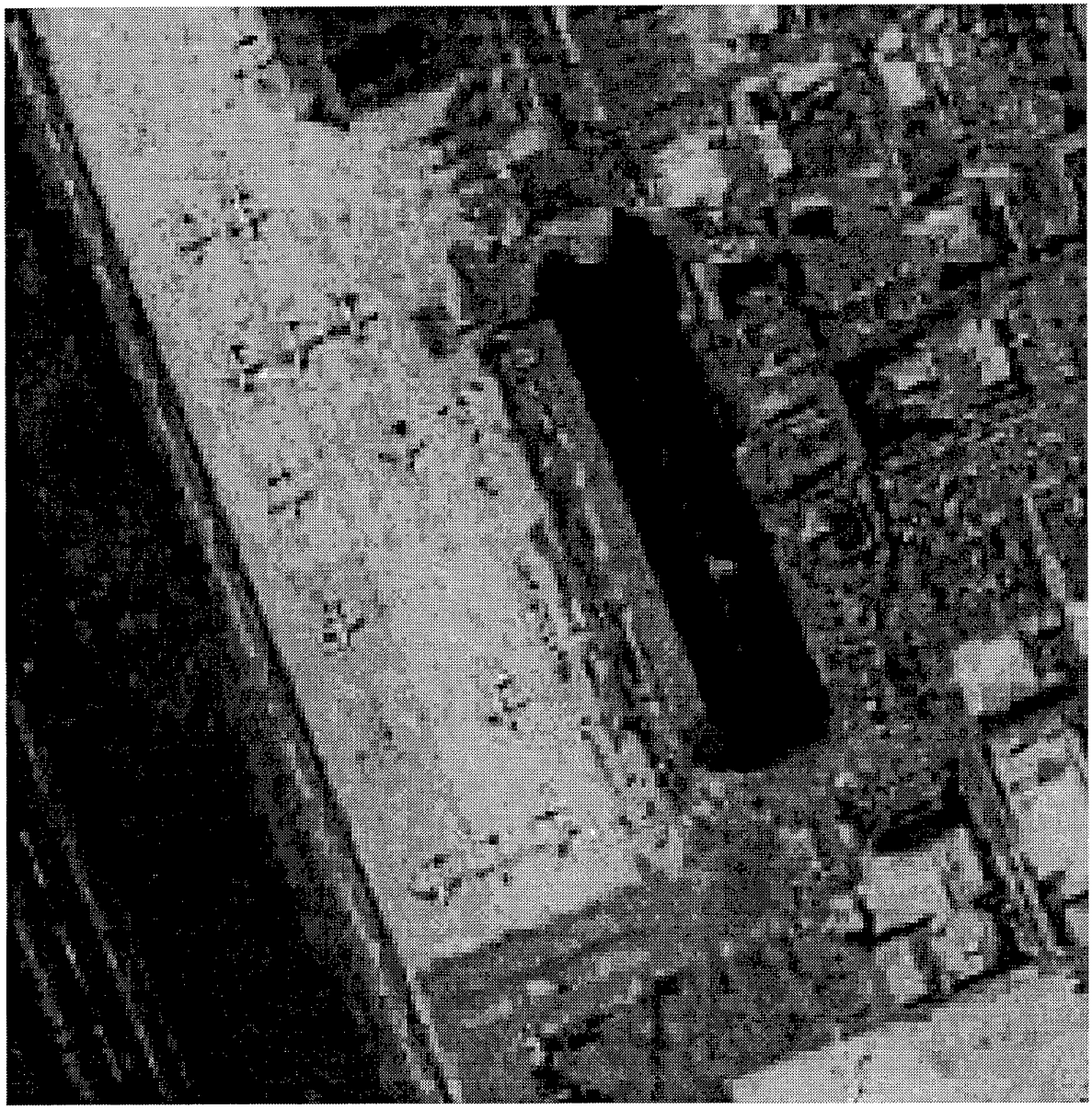


Figure 2.3: Portion of Moffet Naval Air Station EO image moderately compressed with DSSA to a moderate data rate.

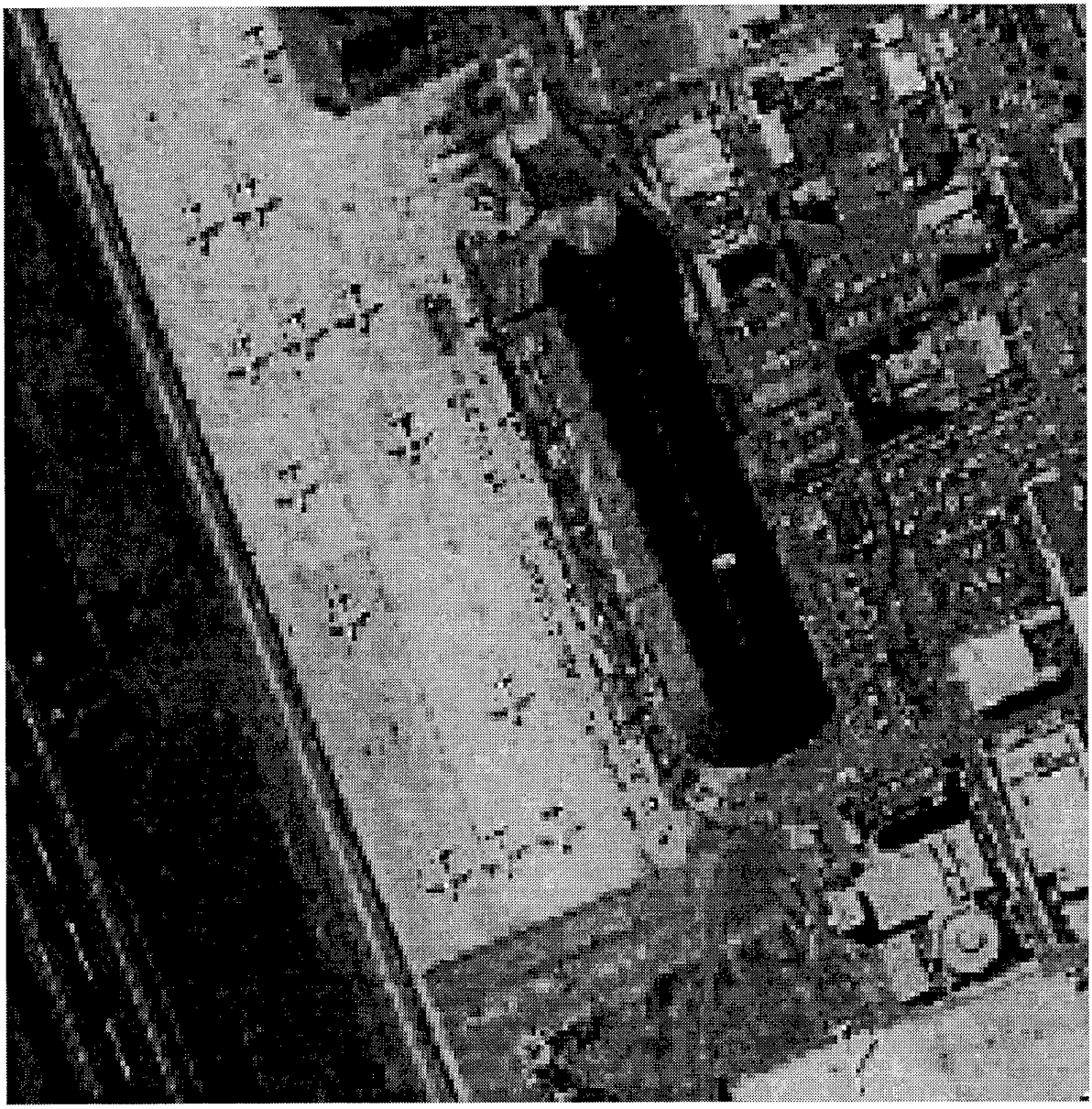


Figure 2.4: Portion of Moffet Naval Air Station EO image mildly compressed with DSSA to a high data rate.

nition process is a conceptually interesting problem. For instance, a decision system could first attempt to classify data represented by a coarse approximation. If the classification does not succeed with a high level of confidence, additional details are then added to the data representation such that a more accurate representation is obtained. Then the decision system tries once again to reach a classification decision with an acceptable level of confidence. This process is repeated until the data are confidently classified. These figures of the Moffet airfield show DSSA systems designed with the intent of data compression. GTRI desires to integrate DSSA classification and compression into a single demonstration system for EO-image processing, but GTRI does not possess an extensive set of EO imagery with targets of interest to the Navy identified by ground truth data. However, GTRI does have plans to generate a simple EO image demonstration using the Moffet Airfield image. Details plans for completing Tasks 2–6 in relation to this EO image are given in the next chapter.

2.2 X-ray Images

Experiments were conducted on digitized mammogram images, which contain microcalcifications, an early indicator of breast cancer. The NN-RVQ was applied to this dual-use data set as an algorithm for computer assisted detection (CAD) of breast cancer. Although, not a primary interest of the Navy, this data set was readily available, and provided a synergistic opportunity for exercising recent refinements to DSSA, and hence received the most attention in this initial phase of this ONR grant. Furthermore, the extensive set of experiments conducted with mammograms serves to illustrate the possibilities resulting from applying DSSA to EO, IR, MS and SAR imagery.

X-ray Task 1: Database Acquisition—Completed

Forty digitized mammogram images were obtained from an Internet site (via anonymous FTP from figment.csee.usf.edu) established with data acquired by Dr. Karssemeijer[1]. Information that identified the locations of known microcalcification clusters contained in the mammograms (truth data) was also obtained. Each of the mammograms contained various numbers of clustered microcalcifications. A total of 96 microcalcification clusters identified by radiologists are specified in the truth data.

X-ray Task 2: Test Plan Definition—Completed

Twenty-one of the mammogram images were used for generating training sets, and nineteen mammograms were reserved for testing purposes to evaluate the performance of the DSSA classifier.

[1]Department of Radiology, University Hospital Nijmegen, PO Box 9101, The Netherlands, nicomb-fys.kun.nl

X-ray Task 3: DSSA Implementation—Completed

A mammogram block size of 5×5 pixels was selected for DSSA processing. Each 5×5 block contains a total of 25 pixels, where the 8-bit intensity value of each pixel may be viewed as providing a feature value in a composite 25-dimensional feature space.

A total of 902 microcalcification training blocks were extracted from the known 41 microcalcification clusters contained in the training images. Each training block was reflected and rotated in eight ways (the eight symmetries of the square) to expand the size of the training set. Thus, the final microcalcification training set contained $8 \times 902 = 7,216$ blocks.

X-ray Task 4: Template/Codevector Design—Completed

The microcalcification-block training set was used to design a 37-stage DSSA classifier, with each stage consisting of 64 templates. The DSSA templates where then inserted into a nearest neighbor DSSA classifier.

X-ray Task 5: DSSA Performance Evaluation—Completed

The performance of the DSSA classifier was evaluated by estimating the probability of detections on the 19 mammograms retained for testing purposes. The output of the detector was a colored *detection map*. Regions of the detection map without detections were colored a light brown. Circles of various colors were drawn around each DSSA declared detection. Both the size and color of the circle indicate the degree of confidence with which the DSSA detection was declared: the larger the circle—the more confident the declaration; the cooler the color—the more confident the declaration (see the printed results in Appendix A where yellow, green, and blue indicate progressively more confident detections). The detection maps also indicate the locations of the radiologist-identified microcalcifications with black circles that enclose the human-detected locality of the microcalcifications.

The test images contained a total of 45 human-detected microcalcification clusters. Each of the mammograms and resulting overlays are shown in Appendix A. A total of 39 out of the 45 events were detected with the DSSA algorithm, giving an estimated probability of detection of 87%. Table 2.1 summarizes the detection results for each of the tested mammograms.

The false alarm rate varied greatly; some mammograms contained only a few false alarms and other mammograms contained many (see the colored detection overlays). However, all detections declared by the DSSA algorithm were displayed in these test results regardless of the associated confidence level. Improved false alarm rates can be easily obtained by not displaying the weaker detections, but at this stage of research, displaying all events that the DSSA classifier found suspect is of interest. Furthermore, since there were no images in the Nijmegen data base that did not contain microcalcifications, there are as yet no estimated false alarm rates determined for mammograms that are known to be clear of microcalcifications.

Mammogram Index	Number of Human-Detected Events	Number of DSSA-Detected Events	Number of DSSA-Missed Events	Detection Percentage Percent
c11c	1	1	0	100%
c11o	1	1	0	100%
c12o	13	11	2	85%
c13c	1	1	0	100%
c13o	1	1	0	100%
c14c	2	2	0	100%
c14o	2	2	0	100%
c15c	1	1	0	100%
c15o	1	1	0	100%
c16c	1	1	0	100%
c16o	1	1	0	100%
c17c	9	8	1	89%
c18c	2	2	0	100%
c18o	1	1	0	100%
c18e	1	1	0	100%
c19c	2	1	1	50%
c19o	3	1	2	33%
c20c	1	1	0	100%
c21o	1	1	0	100%

Table 2.1: X-ray Image DSSA Classification Results

X-ray Task 6: DSSA Feasibility and Practicality Assessment—Completed

Good detection rates where obtained on the mammogram images. False alarm rates were sometimes excessive. All of the missed detections occurred in four of the mammograms: c12o, c17o, c19c, and c19o. In all of the missed detections the microcalcifications were in regions with low X-ray density (dark mammograms). Thus, X-ray density may be a factor related to DSSA detection performance.

DSSA classification processing required on the order of 5-10 minutes on a SUN workstation for each processed mammogram (each mammogram contained $2,048 \times 2,048 = 4,194,304$ pixels). At most, 3,200 multiply-adds are required to classify each pixel of the mammogram. The amount of memory required to store the DSSA templates is 241K of memory.

The experimental results of this research established that the use of a DSSA data representation for direct classification of mammogram pixel data is feasible.

2.3 Sidescan Sonar Images

Sonar Task 1: Data Acquisition—Completed

The U.S. Navy must be able to prepare potential amphibious assault areas (AAA) by detecting and destroying enemy sea mines with autonomous underwater vehicles (AUV) and tethered unmanned underwater vehicles. This includes the ability to search and secure proud mines (mines which lie on the sea floor), buried mines, tethered mines, and drifting sea mines which pose a threat to Navy craft and personnel. A technical goal of the Navy is to develop the ability to conduct covert surveillance for classification and identification of all types of enemy sea mines that may exist in the very shallow water (VSW) environment typical of AAAs. The basic technical strategy adopted by the Navy to achieve these objectives and capabilities is to employ a suite of long range magnetic, acoustic backscatter, and sonar sensors. The overall research goal of the Navy is to develop autonomous systems that are robust in a real underwater environment typical of VSW. The systems must deal with expected signal-to-noise ratios, specular scattering, and multipath propagation effects. Most of all, the systems must provide sufficient performance levels to support human confidence in the results of an autonomous and unmanned surveillance and preparation of a potential AAA before Navy assets and personnel forces are permitted to enter the area to execute Navy missions and assignments.

In order to reach a confidence level that is high enough to justify risk to human life, future AUV systems may be required to relay data back to command stations for human verification of autonomous detection decisions. GTRI proposes the use of DSSA for the integration of both onboard detection processing, and data compression for bandwidth efficient communications to the base station.

GTRI has obtained side scan sonar images from CSS. Although this data has been evaluated previously for DSSA classification performance, the DSSA algorithm has been refined under GTRI internal research and development dollars. Thus GTRI plans to revisit this data set within the context of classification reevaluation, and plans a new evaluation to quantify joint DSSA classification/compression performance.

Sonar Task 2: Test Plan Definition—Completed

The Coastal Systems Station has previously suggested that the CSS sonar data be partitioned into training and testing data subsets. GTRI will conform to the CSS recommendation to allow comparisons with results obtained by other researchers.

Sonar Task 3: DSSA Implementation—Initiated

A sonar block size of 9×9 pixels was been tentatively selected for DSSA processing. Each 9×9 block contains a total of 81 pixels, where the 8-bit intensity value of each pixel may be

viewed as providing a feature value in a composite 81-dimensional feature space.

Training blocks are currently being extracted from known target locations. Each training block is being reflected along a horizontal axis to expand the size of the training set. The training set currently contains 824 blocks.

Sonar Task 4: Template/Codevector Design—Initiated

Preliminary DSSA template/codevector sets have been designed to obtain a qualitative assessment of performance. Currently, a 25-stage DSSA classifier, with each stage consisting of 32 templates has been designed.

Sonar Task 5: DSSA Performance Evaluation—Initiated

Qualitative assessments have been performed to judge the suitability of the DSSA block size selected in Sonar Task 1. Figures 2.5 and 2.6 at the end of this chapter show detection overlays of processed CSS sidescan sonar images that are completely analogous to the mammogram overlays in Appendix A. The corresponding CSS image is not shown in this report to prevent document distribution restrictions. Plans for completing Sonar Tasks 3–6 are given in the next chapter.

2.4 Synthetic Aperture Radar Images

SAR Task 1: Data Acquisition—Initiated

GTRI has identified and obtained an extensive set of SAR data with ground military targets. However, GTRI has not yet received permission from the Army source to use these data in this study. Plans for completing SAR Tasks 1–6 are given in the next chapter.

2.5 Acoustical Backscatter Signals

Backscatter Task 1: Data Acquisition—Completed

GTRI has obtained the CSS acoustic backscatter data set.

Backscatter Task 2: Test Plan Definition—Completed

GTRI has adopted the CSS acoustic backscatter test plan for classification purposes. This test plan calls for the generation of synthetic clutter with synthesized SNR ratios for evalu-

ation purposes. Plans for completing Tasks 3–6 are given in the next chapter.

2.6 Machine State Status Signals

Task 1: Data Acquisition—Completed

GTRI has obtained the NRaD condensate pump data.

Task 2: Test Plan Definition—Initiated

GTRI has developed a preliminary plan to process these data as a time series sequence, but has not yet fully completed the plan. Suggestions given by Dr. Lake for processing these data will be followed. Plans for completing Tasks 2–6 are given in the next chapter.

2.7 Infrared Images and Multispectral Images

GTRI has not yet obtained any infrared or multispectral images.

EXAMPLE DSSA CLASSIFICATION RESULT

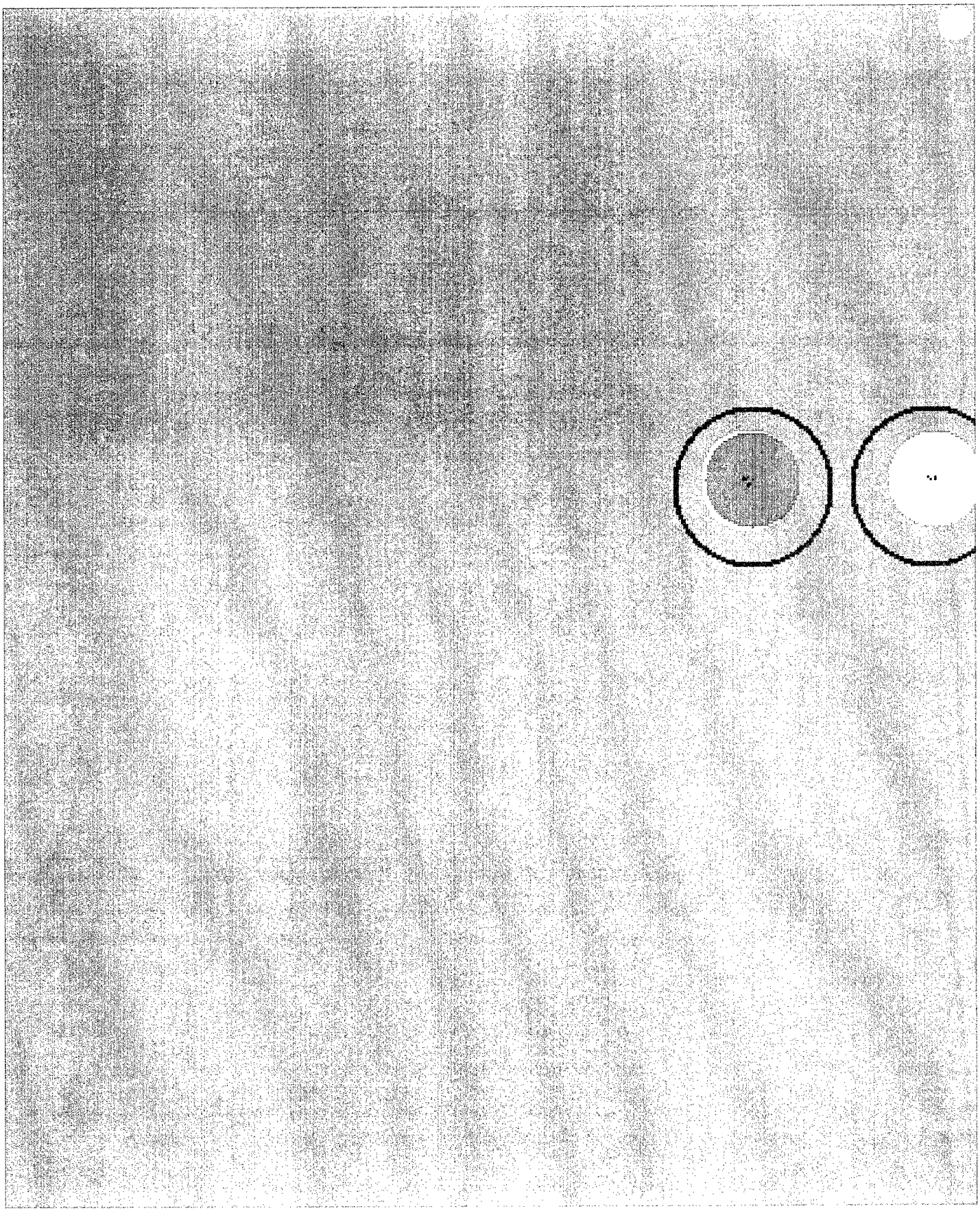
SIDESCAN SONAR IMAGE #si020074

DETECTION OVERLAY SHOWN ON NEXT PAGE

DSSA Classification Results		
Number of Known Sea Mines	=	2
Number of Computer-Detected Mines	=	2
Number of Computer-Missed Mines	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	25
Number of DSSA Templates per Stage	=	32
DSSA Template Size	=	9×9

Figure 2.5: Classified sidescan sonar image overlay #si020074.



EXAMPLE DSSA CLASSIFICATION RESULT

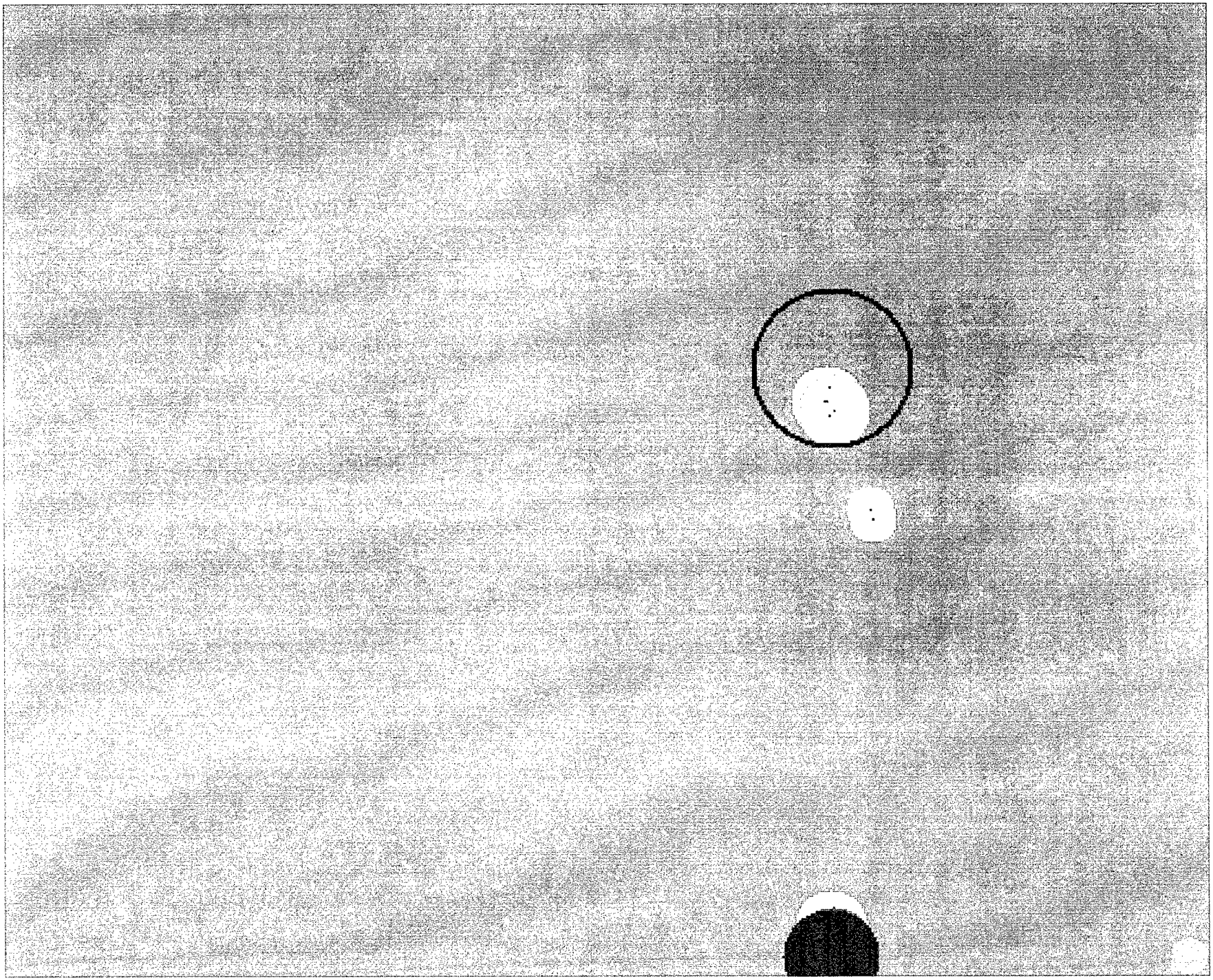
SIDESCAN SONAR IMAGE #si020079

DETECTION OVERLAY SHOWN ON NEXT PAGE

DSSA Classification Results		
Number of Known Sea Mines	=	1
Number of Computer-Detected Mines	=	1
Number of Computer-Missed Mines	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	25
Number of DSSA Templates per Stage	=	32
DSSA Template Size	=	9×9

Figure 2.6: Classified sidescan sonar image overlay #si020079.



Chapter 3

Conclusions and Future Research Plans

3.1 Preliminary Conclusions

A novel feature of DSSA is that NN-RVQs can be designed such that the first stages operate primarily on the easy-to-discriminate data and the latter stages deal with the more challenging target-like clutter data. This approach has computational and theoretical advantages. The computational advantage is that little signal processing effort is expended when classification decisions are easily reached. The theoretical advantage is that latter NN-RVQ stages can attempt to distinguish targets from target-like clutter and provide the necessarily complicated decision surfaces in these challenging areas of the decision space. This fact suggests that DSSA systems result in relatively simple classifiers and data compressors that can be implemented onboard sensor platforms. Although these initial experimental results support this conclusion, additional research will be performed for various sensors to further investigated these possibilities.

The following sections outline research plans for the coming year for the different sensors of interest.

3.2 Electro-Optical Images

GTRI would like to identify an EO image data base with a particular target of interest to the Navy; however, as a fall back position, GTRI plans to use the Moffet image to demonstrate that small targets such as airplanes on an airfield can be identified using the DSSA approach. GTRI will use one half of the Moffet image as training data, and the other half as testing data to generate this rather simple demonstration that can be used for illustration purposes.

3.3 X-ray Images

GTRI plans no further work with X-ray images.

3.4 Sidescan Sonar Images

GTRI will complete the training set extraction process and generate a full set of templates for the DSSA system. The performance of the joint classifier and compressor will be quantified and extensive sample results will be included in the next interim performance report.

3.5 Synthetic Aperture Radar Images

GTRI will seek permission from the Army SAR data source for use of their data on this project. GTRI is also interested in identifying other potential Navy sources for SAR data.

3.6 Acoustical Backscatter Signals

GTRI will continue acoustical backscatter processing during the coming year. A key issue is whether the backscatter should be processed as a high resolution range profile of underwater targets, or processed with the intent of identifying resonances and other target-related phenomenology.

3.7 Machine State Status Signals

GTRI will actively pursue machine state data processing during the coming year.

3.8 Infrared Images and Multispectral Images

GTRI has contacted researchers at the Naval Research Labs (NRL) about possible collaboration on NN-RVQ classification of infrared and multispectral image data. Initial discussions indicate that there would be significant synergism between GTRI's 6.1 and NRL's 6.2 research programs, this collaboration opportunity will be pursued during FY97.

Appendix A

Mammogram Images and DSSA Detection Overlays

EXAMPLE DSSA CLASSIFICATION RESULT

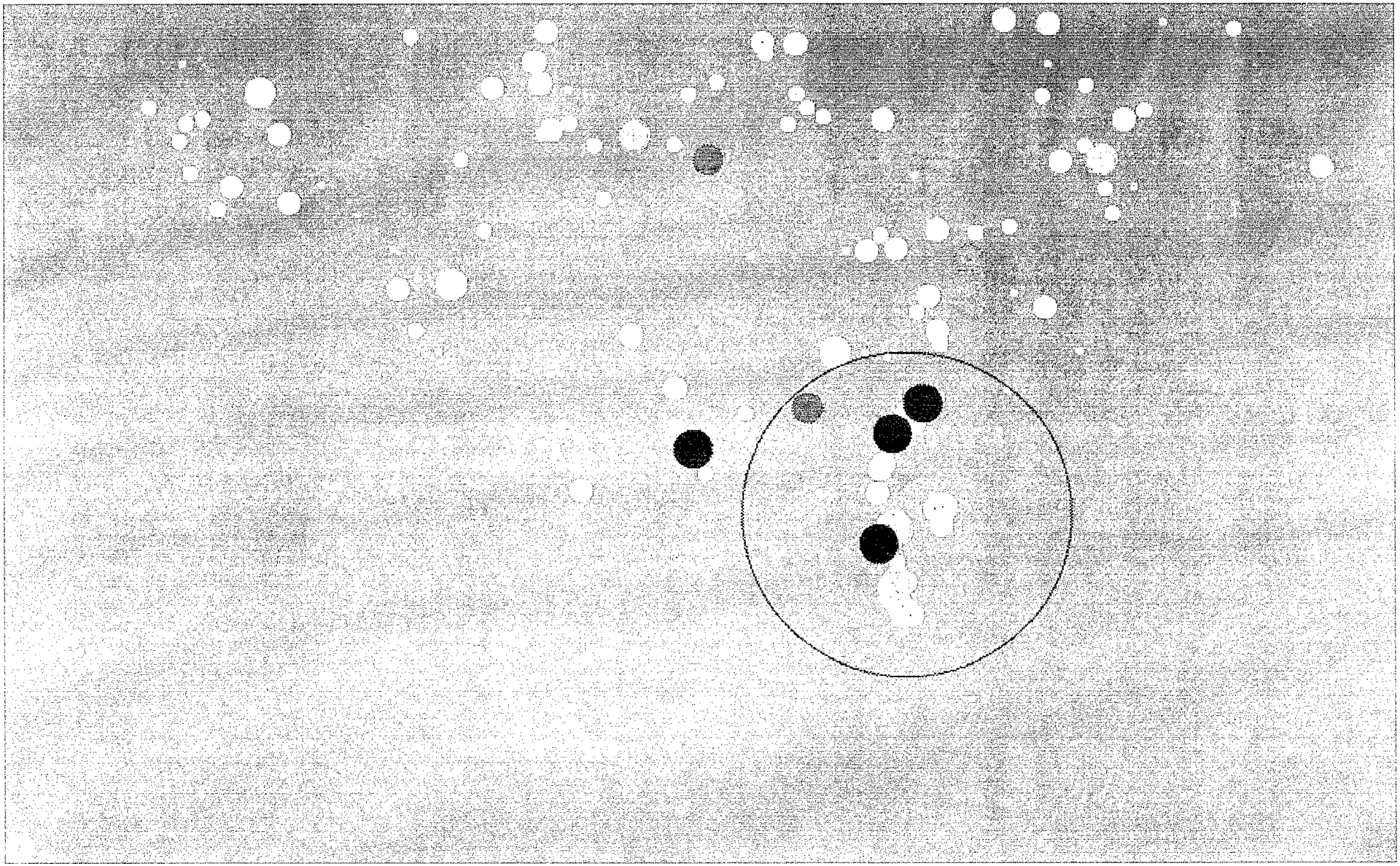
MAMMOGRAM #c11c

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.1: Classified mammogram #c11c.



EXAMPLE DSSA CLASSIFICATION RESULT

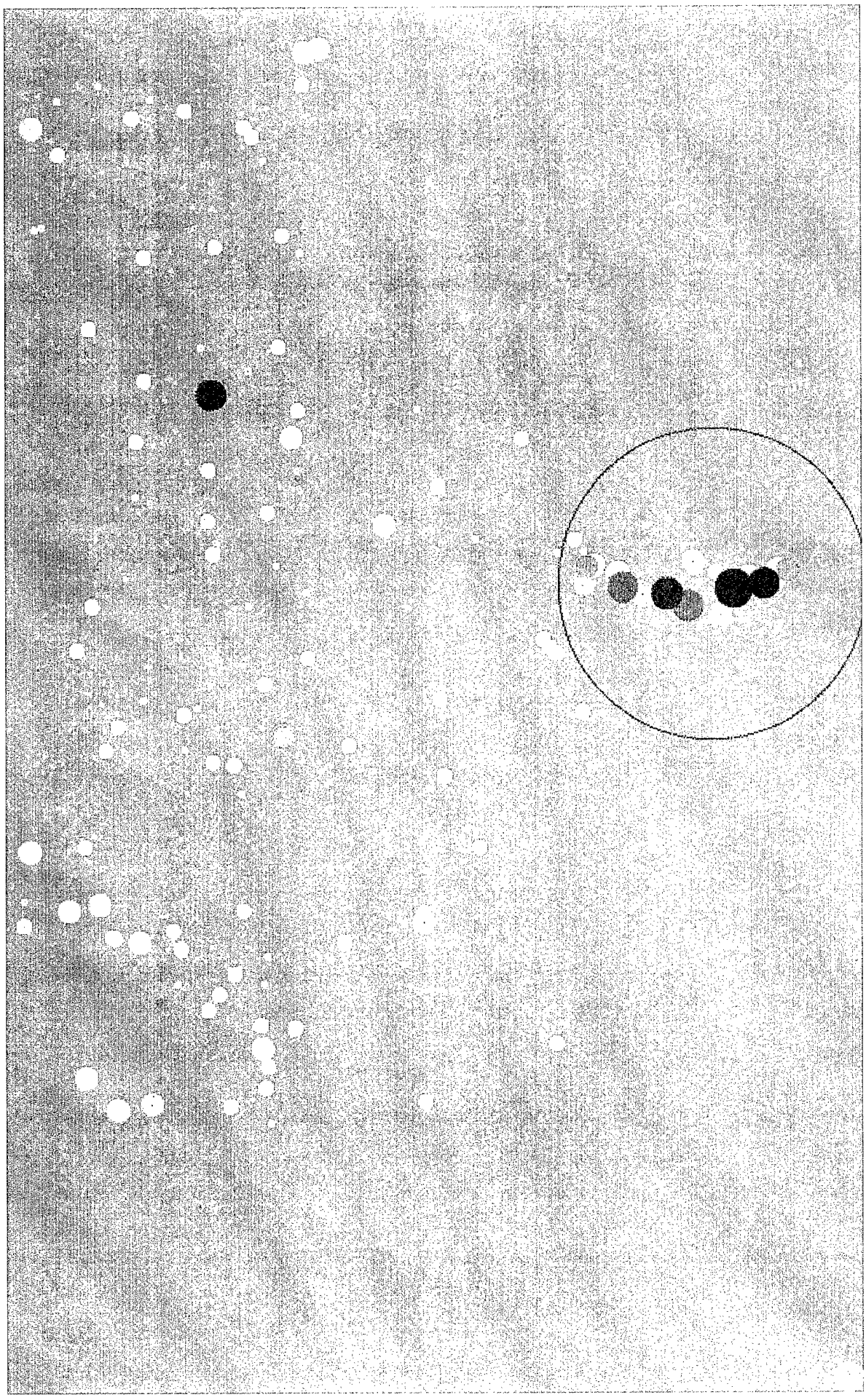
MAMMOGRAM #c11o

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.2: Classified mammogram #c11o.



EXAMPLE DSSA CLASSIFICATION RESULT

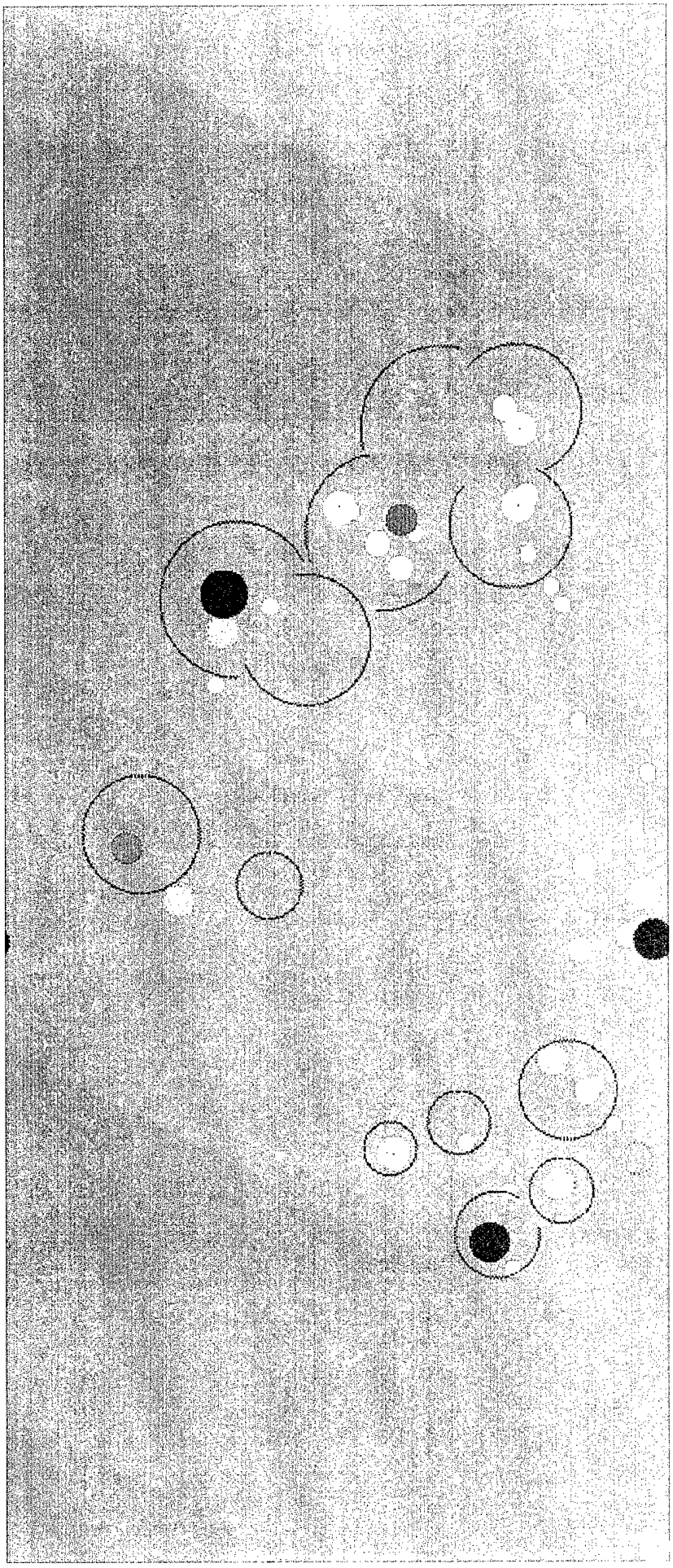
MAMMOGRAM #c12o

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	13
Number of Computer-Detected Clusters	=	11
Number of Computer-Missed Clusters	=	2

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.3: Classified mammogram #c12o.



EXAMPLE DSSA CLASSIFICATION RESULT

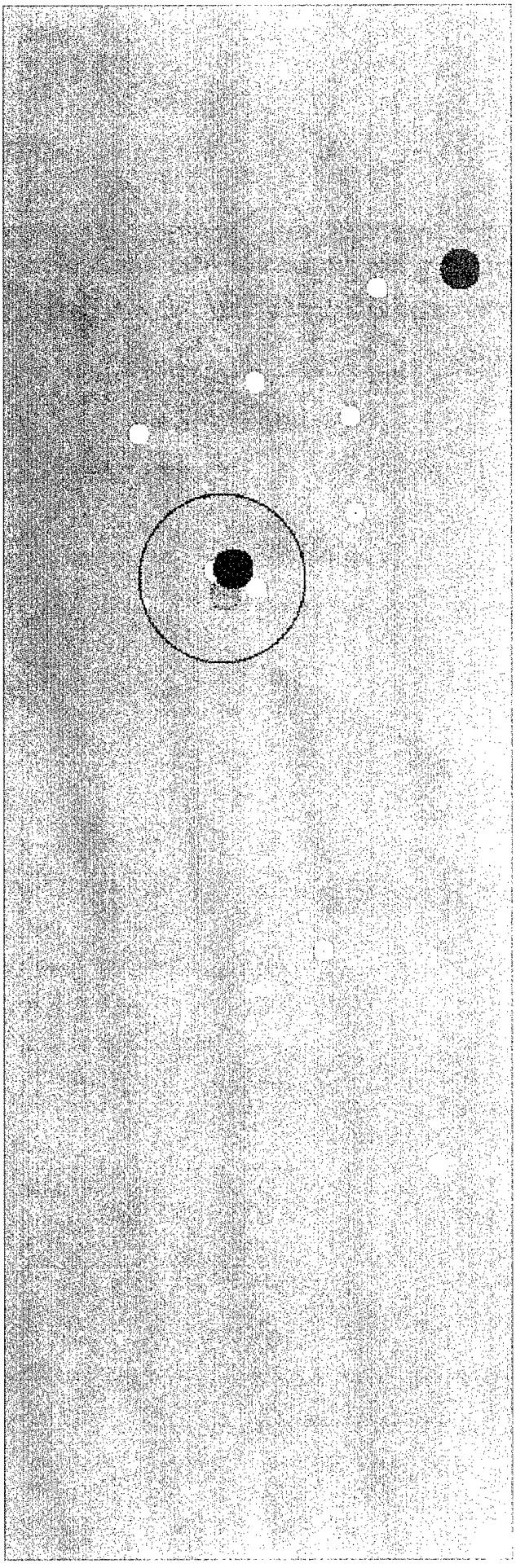
MAMMOGRAM #c13c

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.4: Classified mammogram #c13c.



EXAMPLE DSSA CLASSIFICATION RESULT

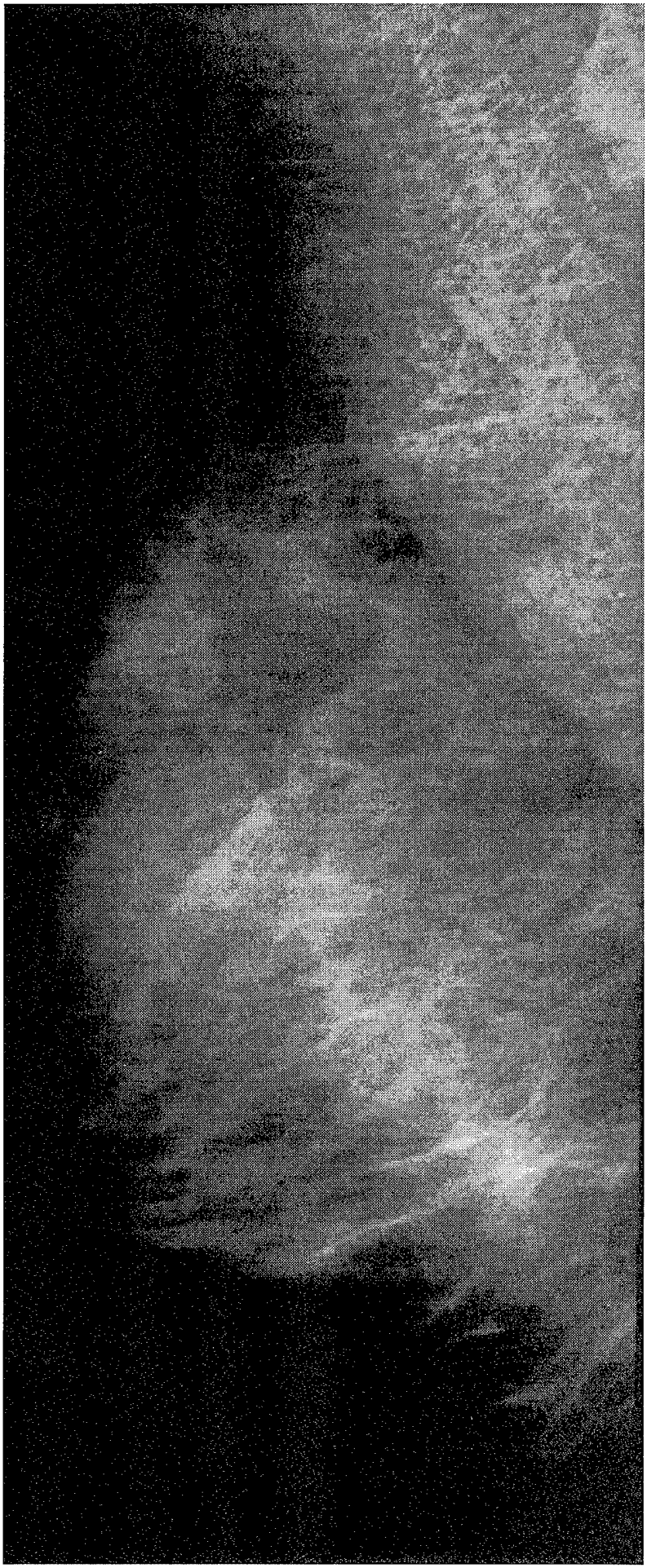
MAMMOGRAM #c13o

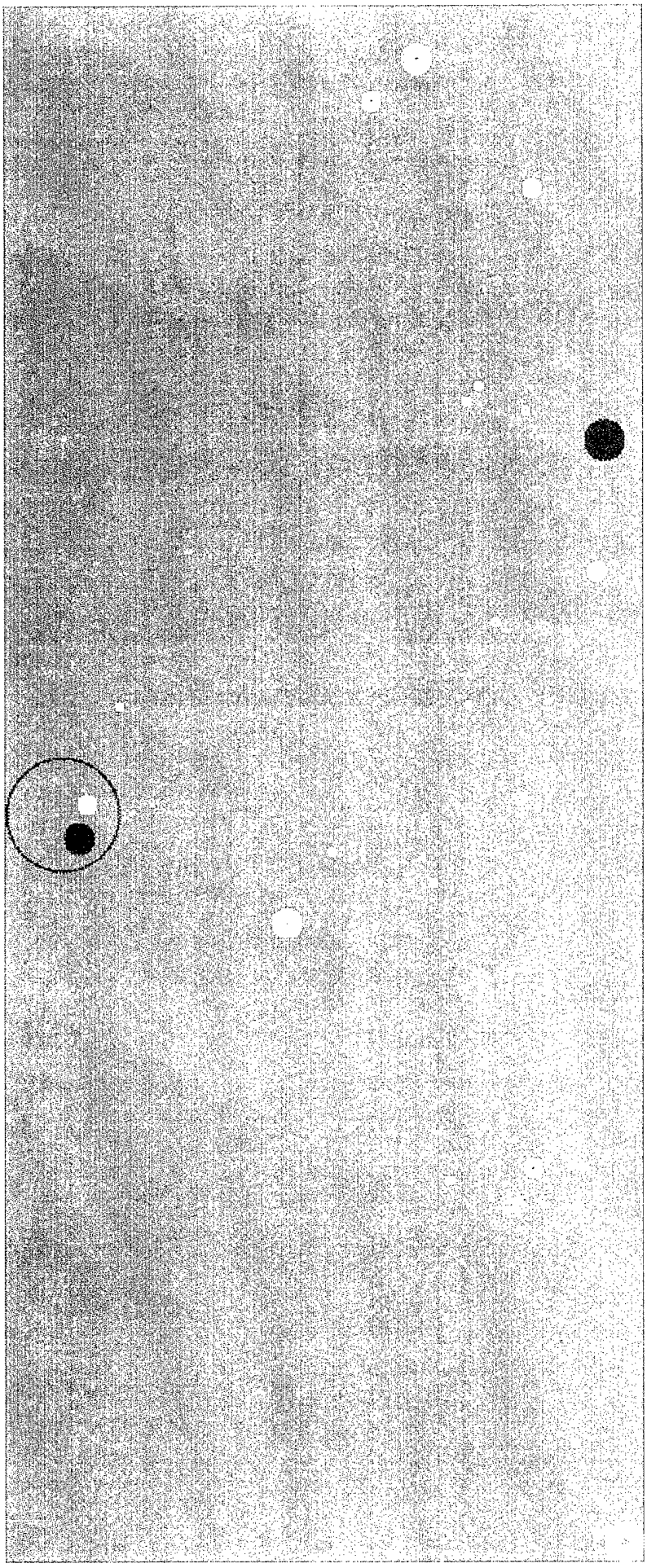
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.5: Classified mammogram #c13o.





EXAMPLE DSSA CLASSIFICATION RESULT

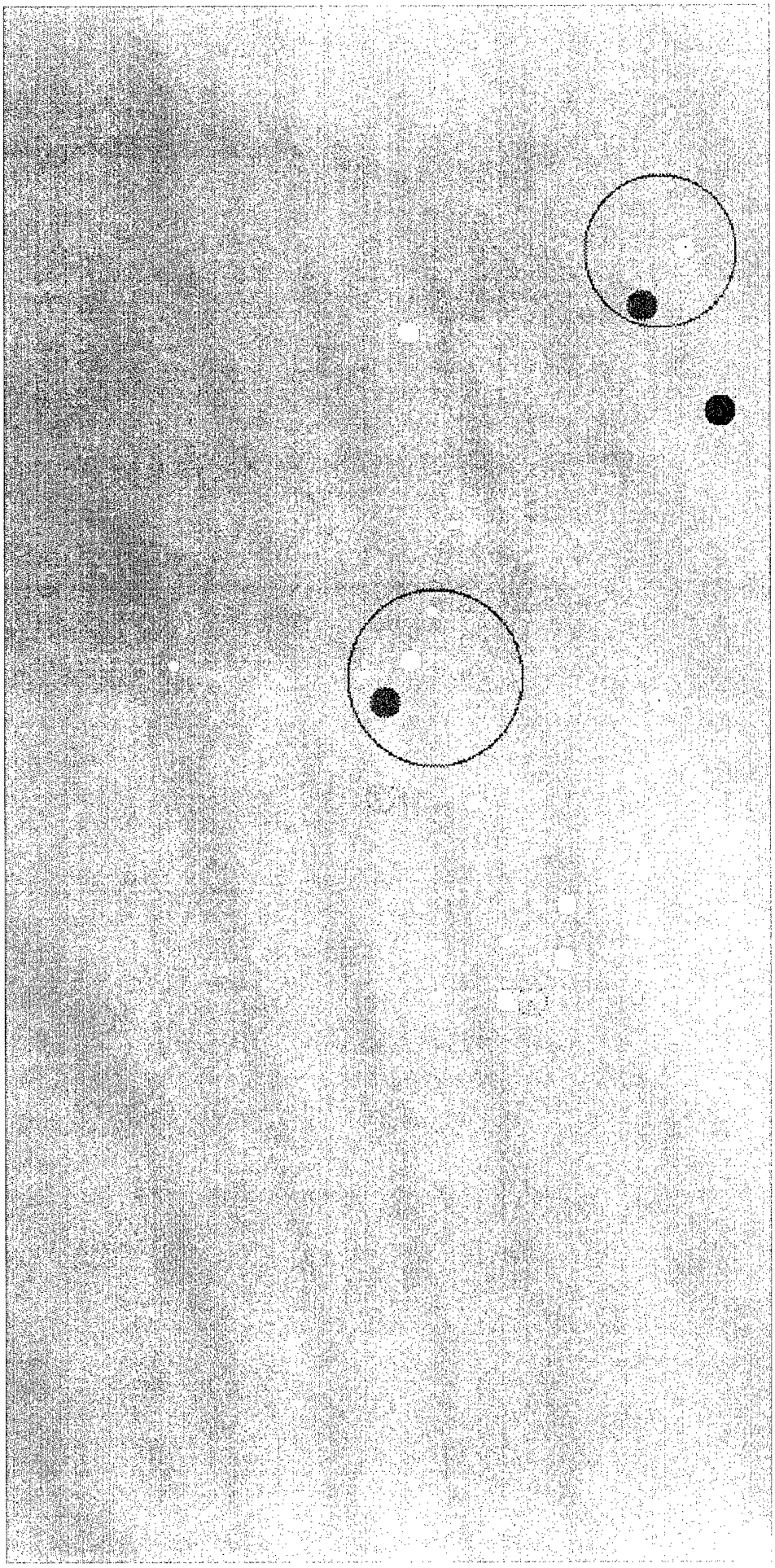
MAMMOGRAM #c14c

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	2
Number of Computer-Detected Clusters	=	2
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.6: Classified mammogram #c14c.



EXAMPLE DSSA CLASSIFICATION RESULT

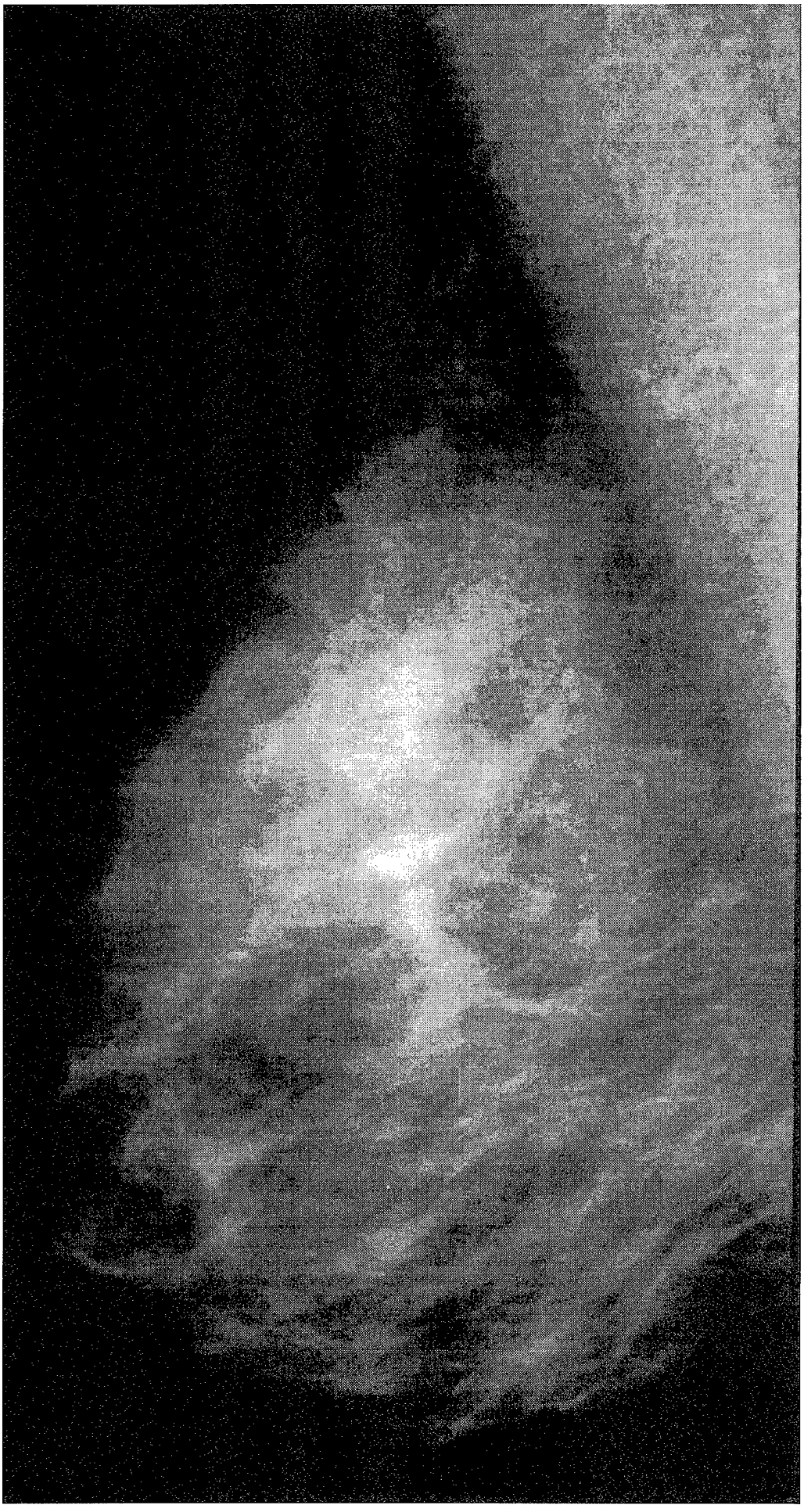
MAMMOGRAM #c14o

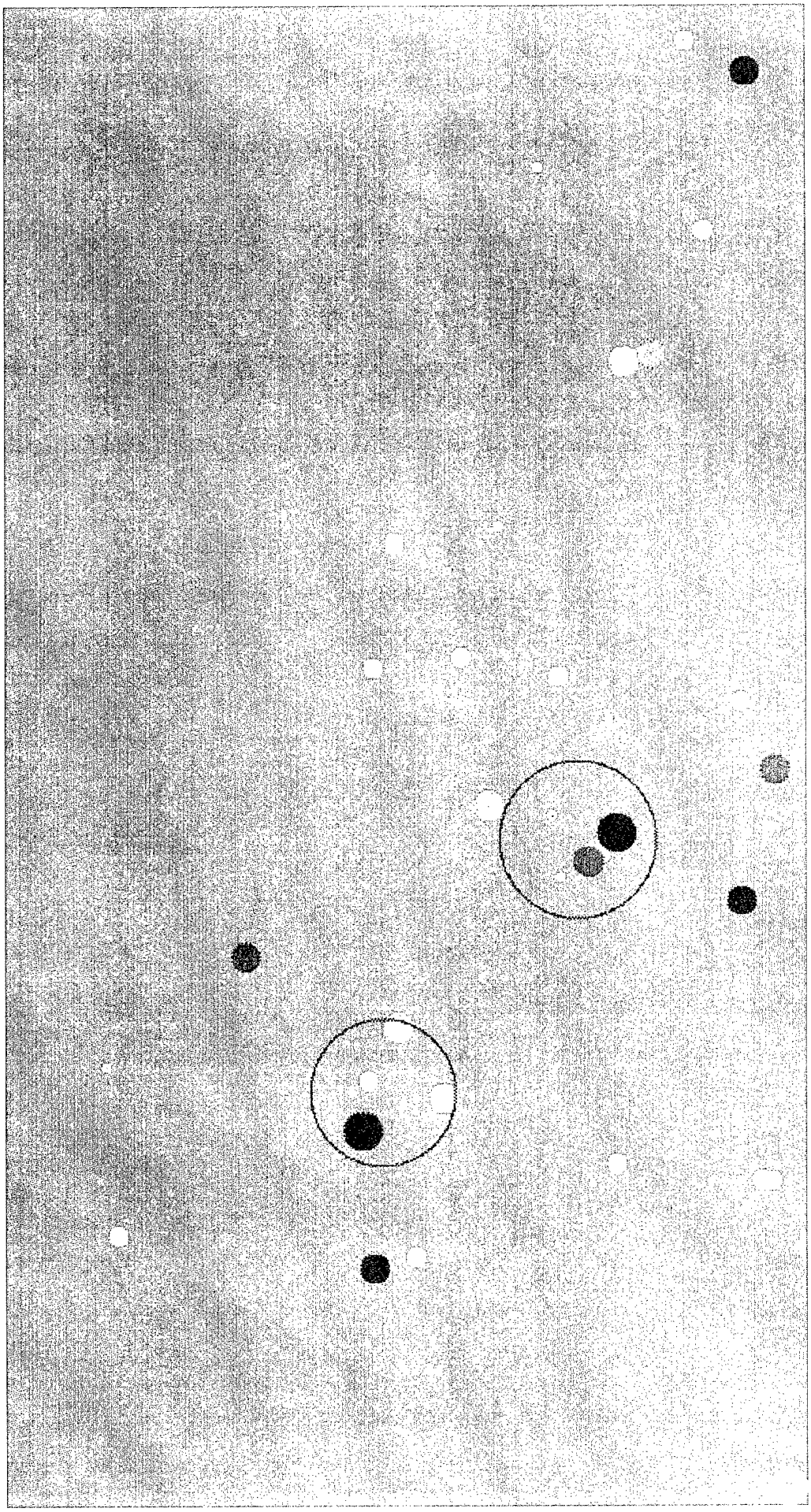
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	2
Number of Computer-Detected Clusters	=	2
Number of Computer-Missed Clusters	=	2

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.7: Classified mammogram #c14o.





EXAMPLE DSSA CLASSIFICATION RESULT

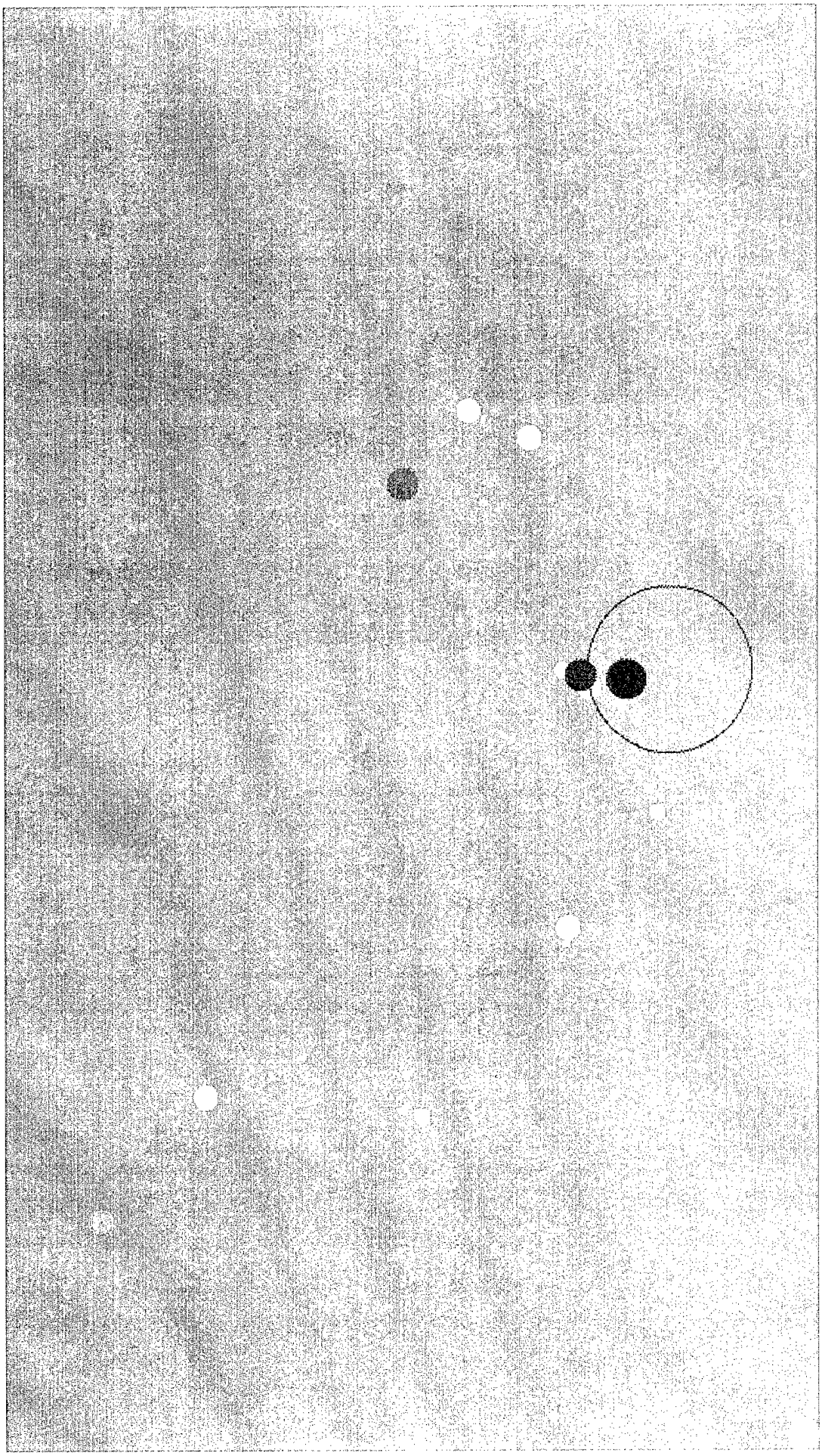
MAMMOGRAM #c15c

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.8: Classified mammogram #c15c.



EXAMPLE DSSA CLASSIFICATION RESULT

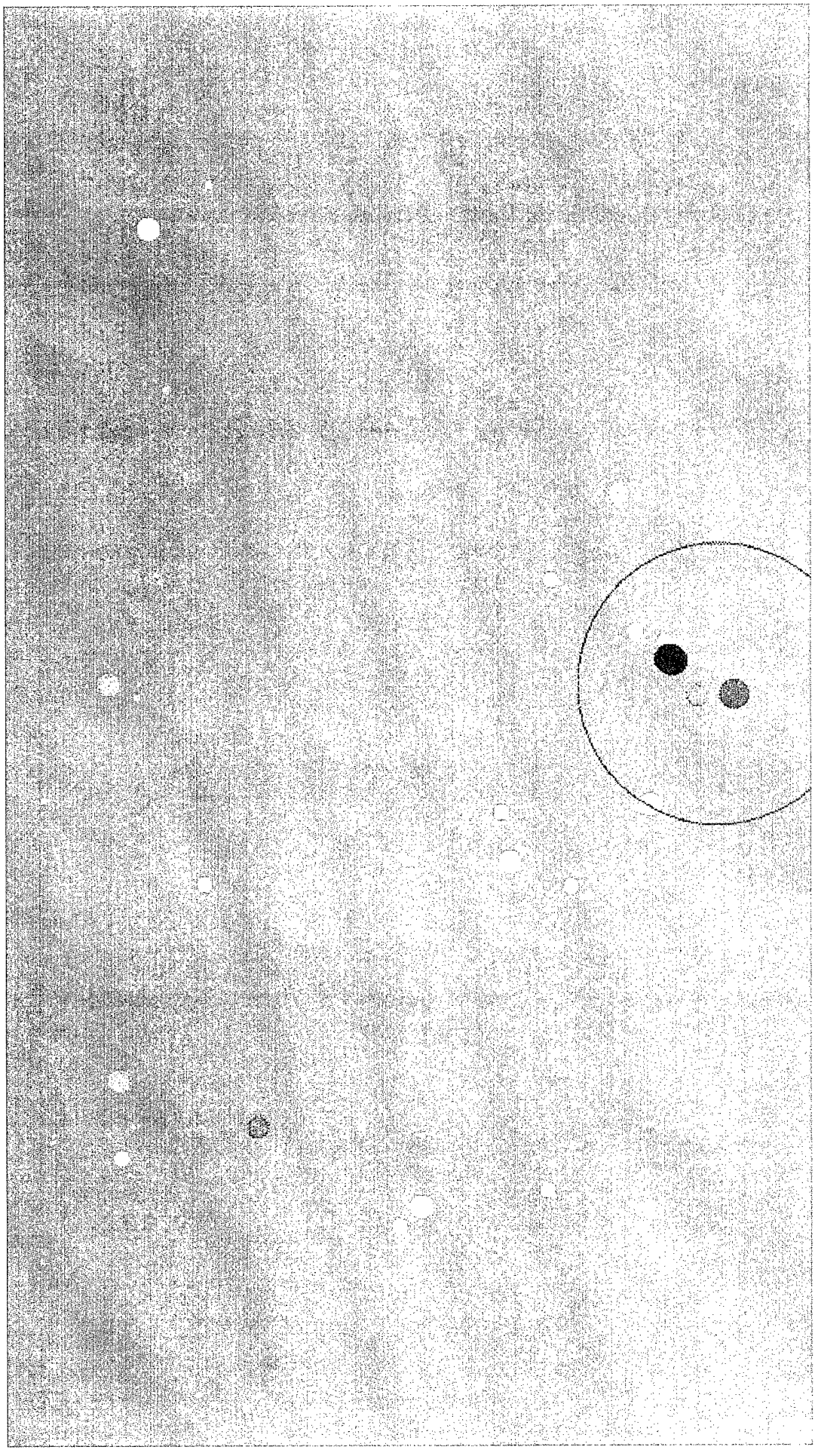
MAMMOGRAM #c15o

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.9: Classified mammogram #c15o.



EXAMPLE DSSA CLASSIFICATION RESULT

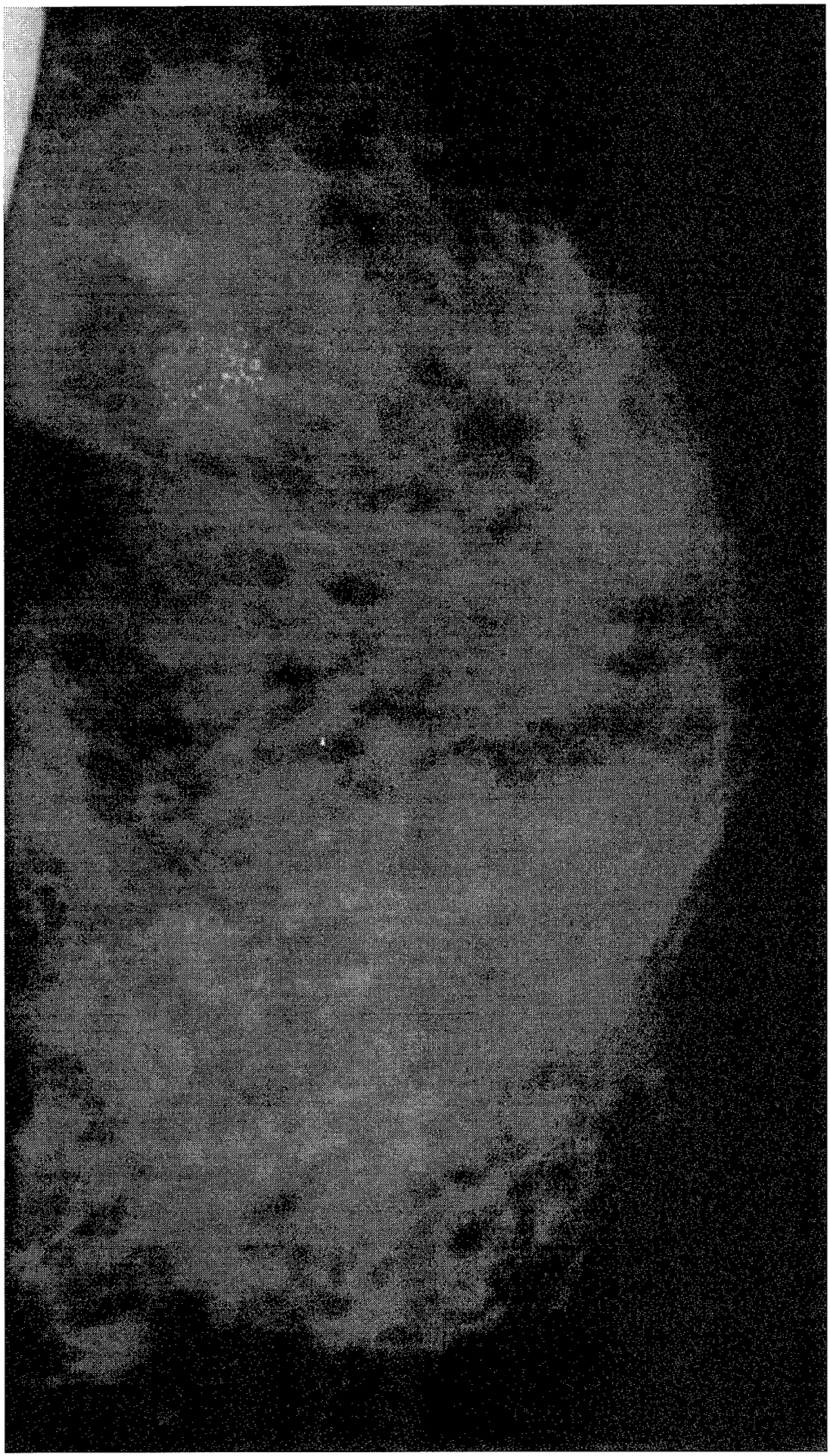
MAMMOGRAM #c16c

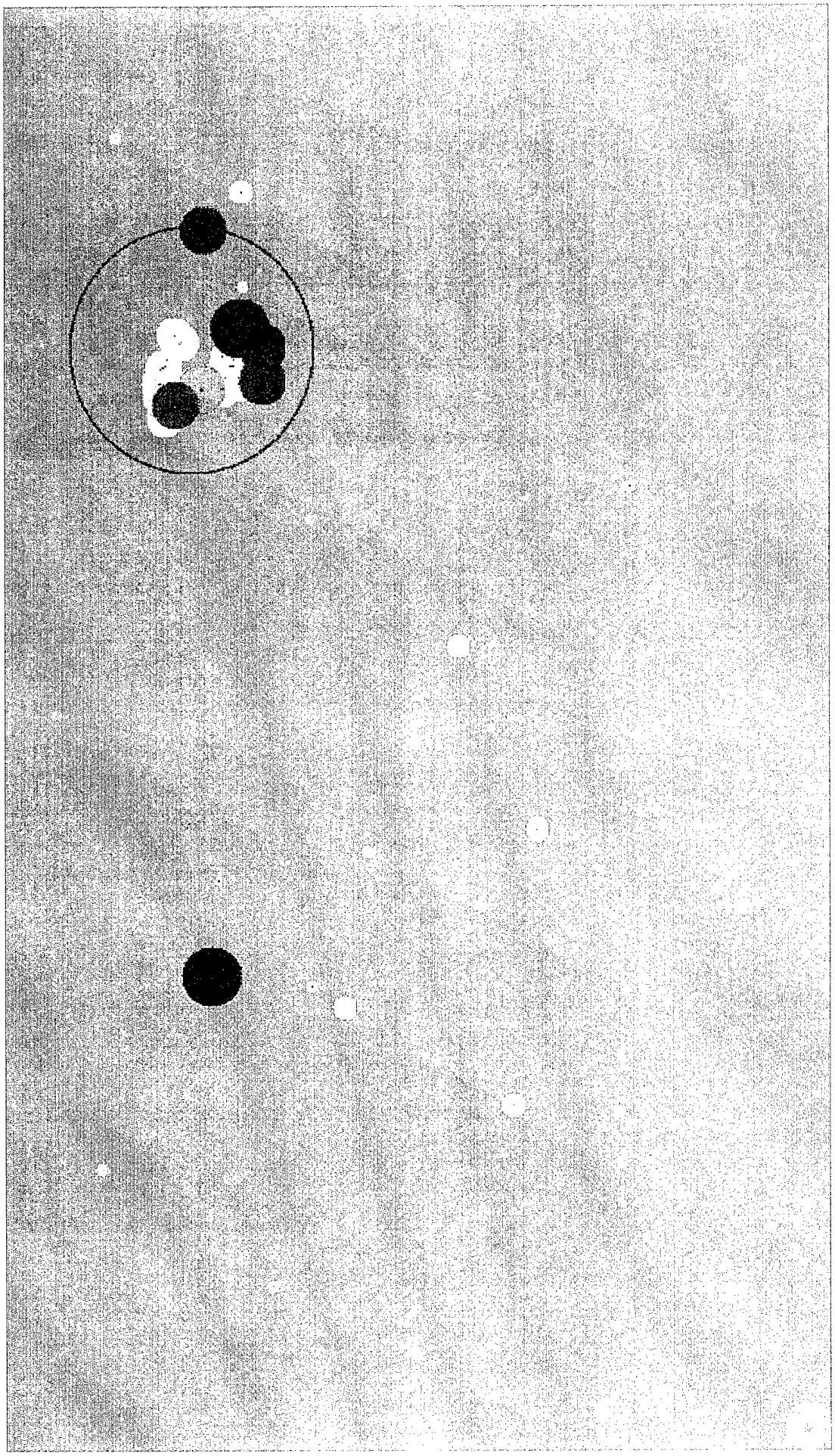
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.10: Classified mammogram #c16c.





EXAMPLE DSSA CLASSIFICATION RESULT

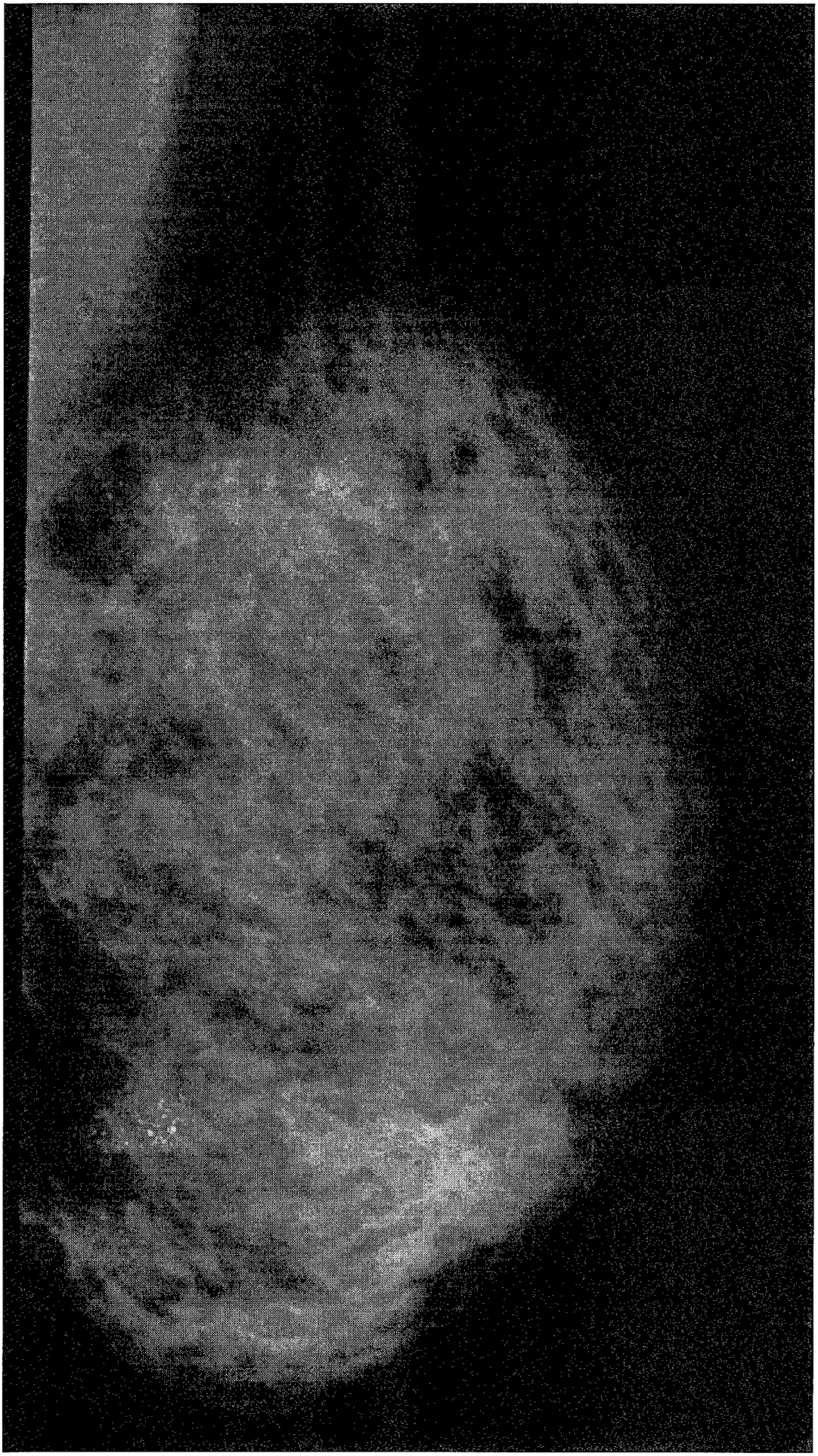
MAMMOGRAM #c16o

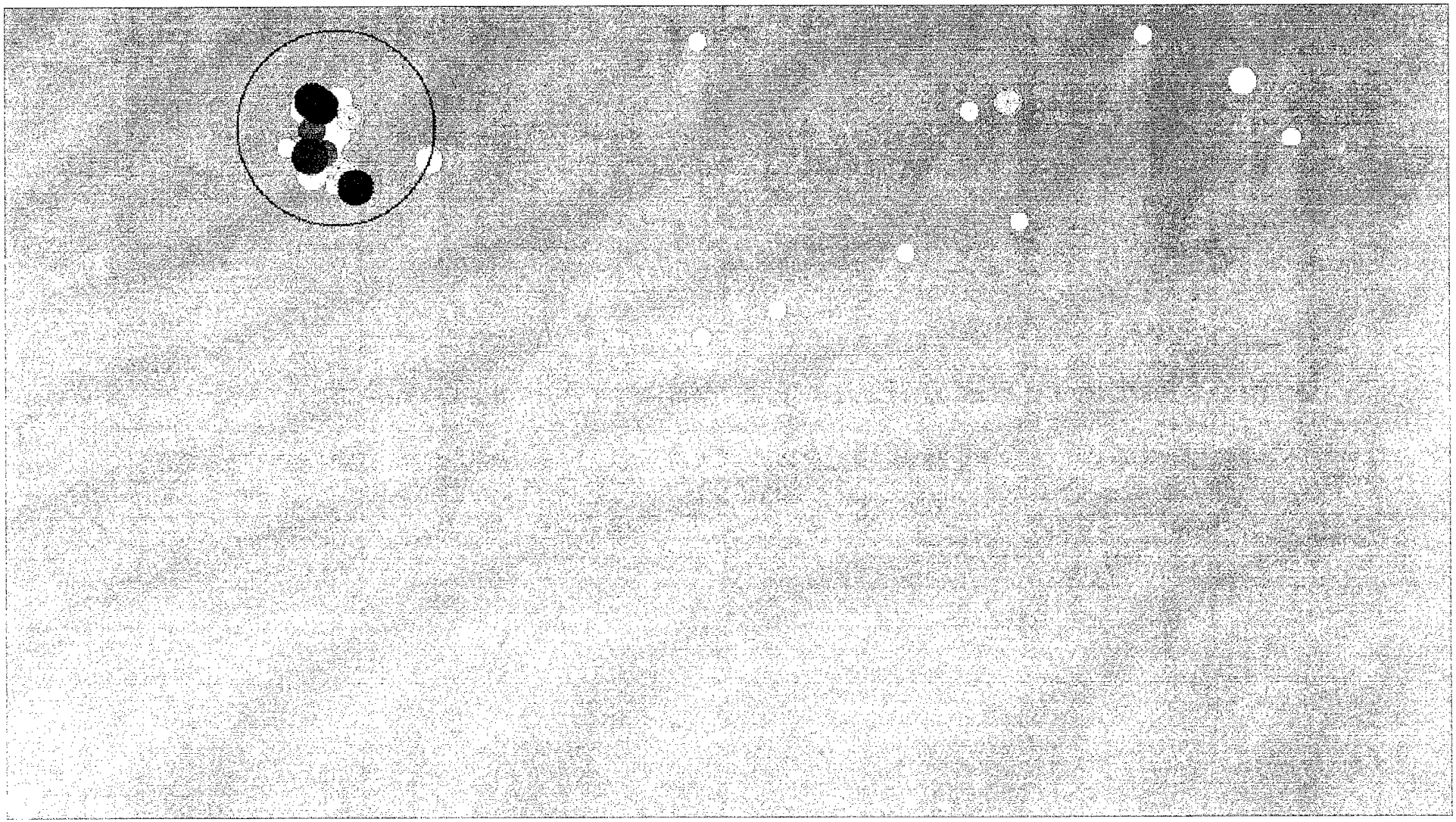
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.11: Classified mammogram #c16o.





EXAMPLE DSSA CLASSIFICATION RESULT

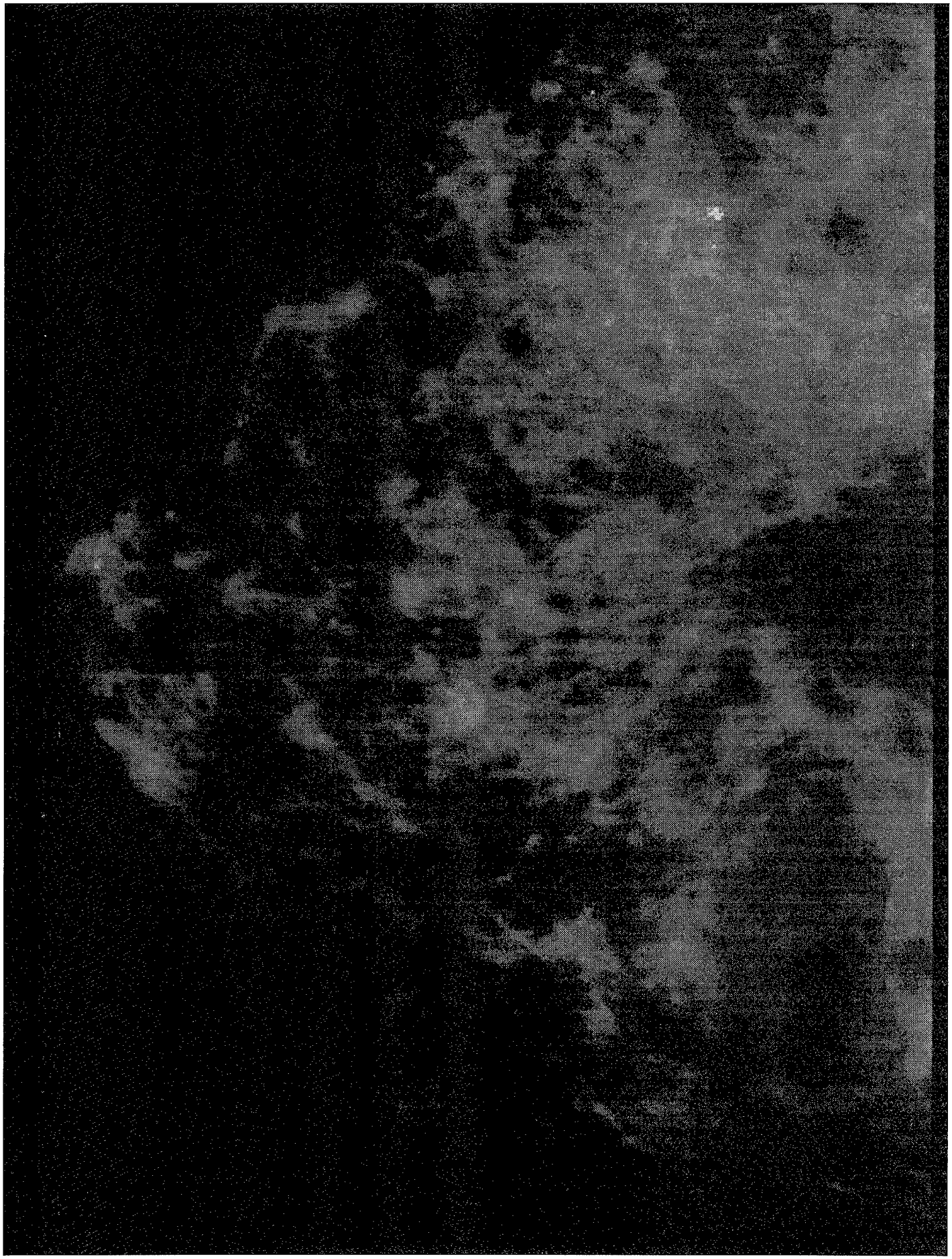
MAMMOGRAM #c17c

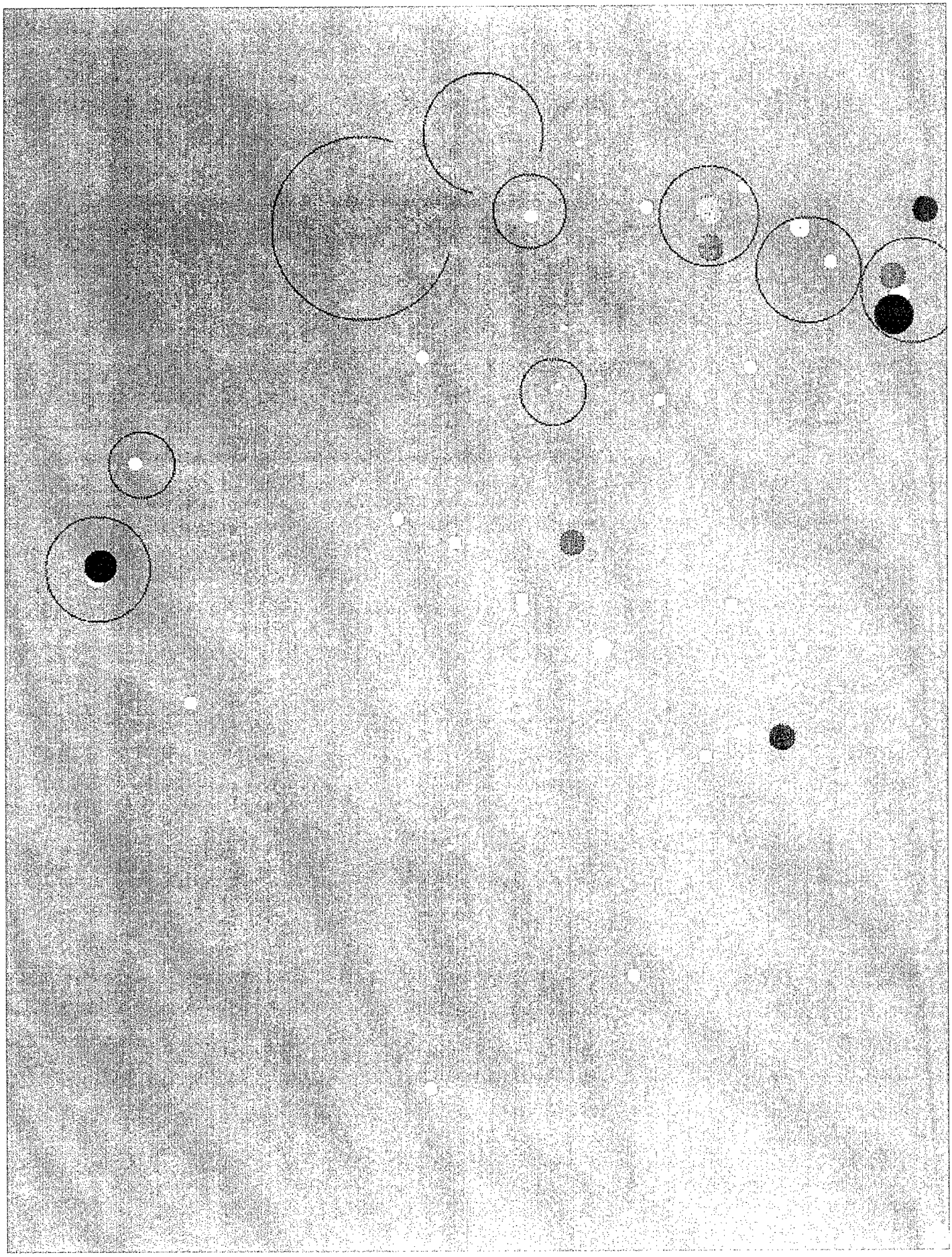
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	9
Number of Computer-Detected Clusters	=	8
Number of Computer-Missed Clusters	=	1

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.12: Classified mammogram #c17c.





EXAMPLE DSSA CLASSIFICATION RESULT

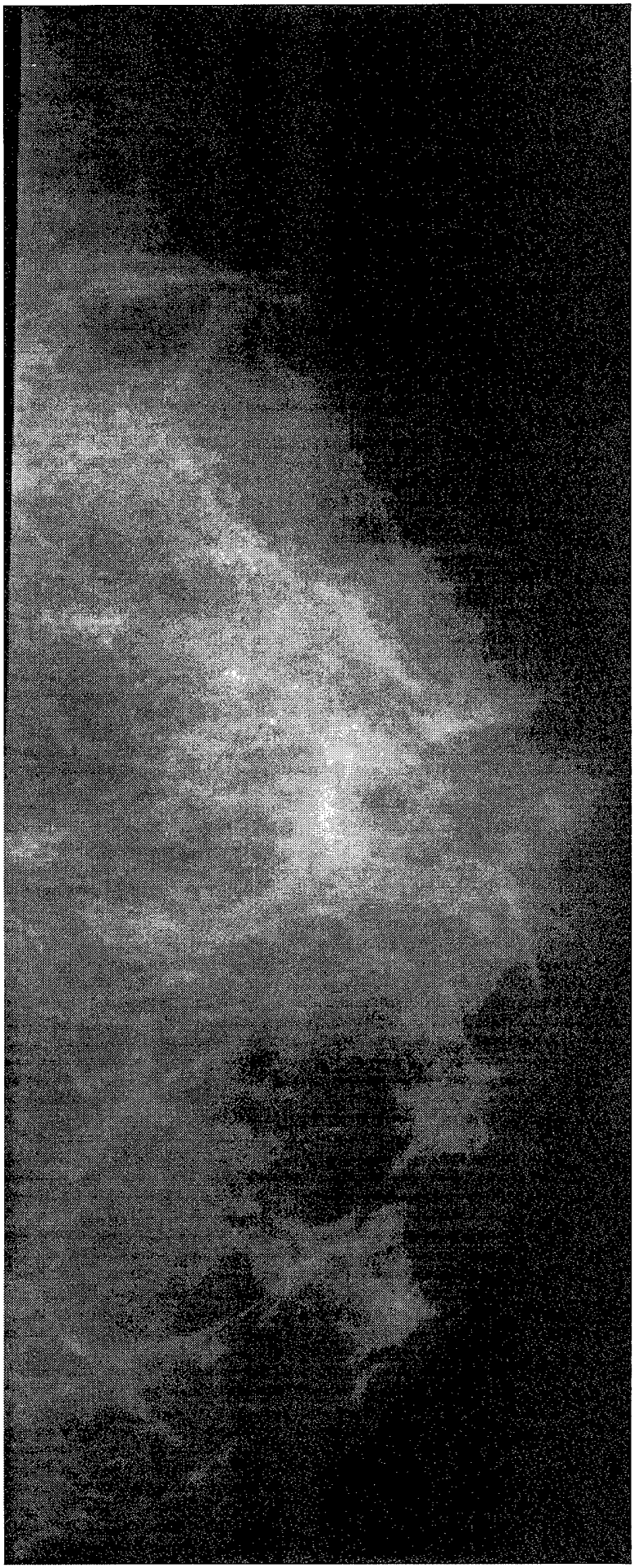
MAMMOGRAM #c18c

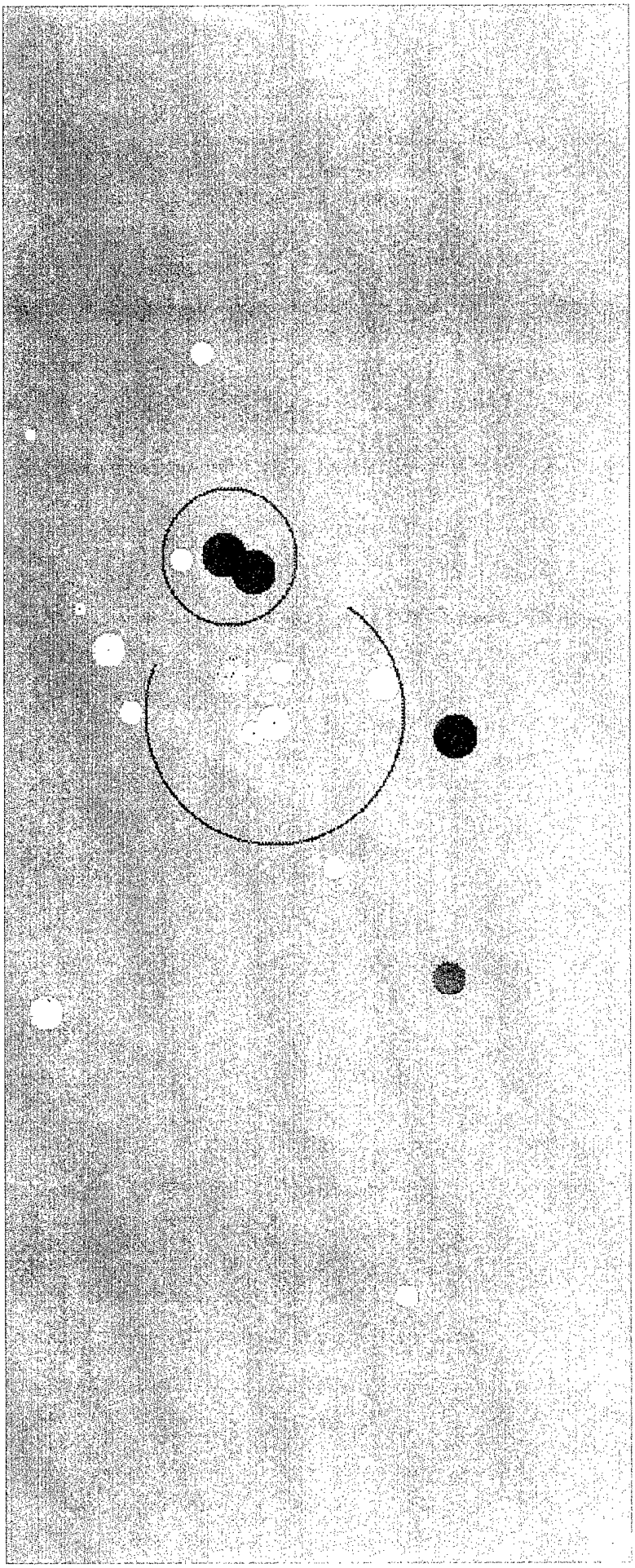
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	2
Number of Computer-Detected Clusters	=	2
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.13: Classified mammogram #c18c.





EXAMPLE DSSA CLASSIFICATION RESULT

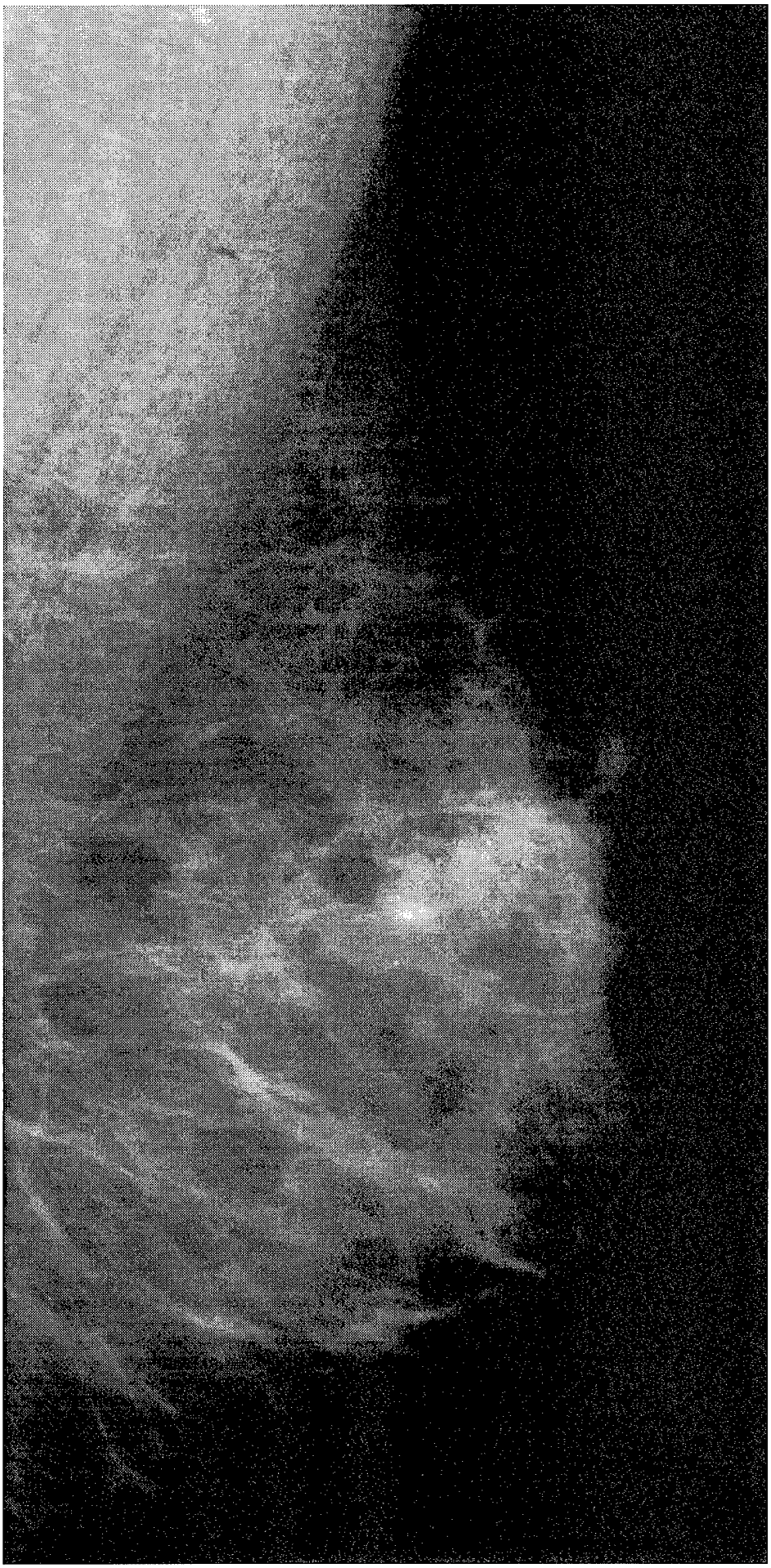
MAMMOGRAM #c18o

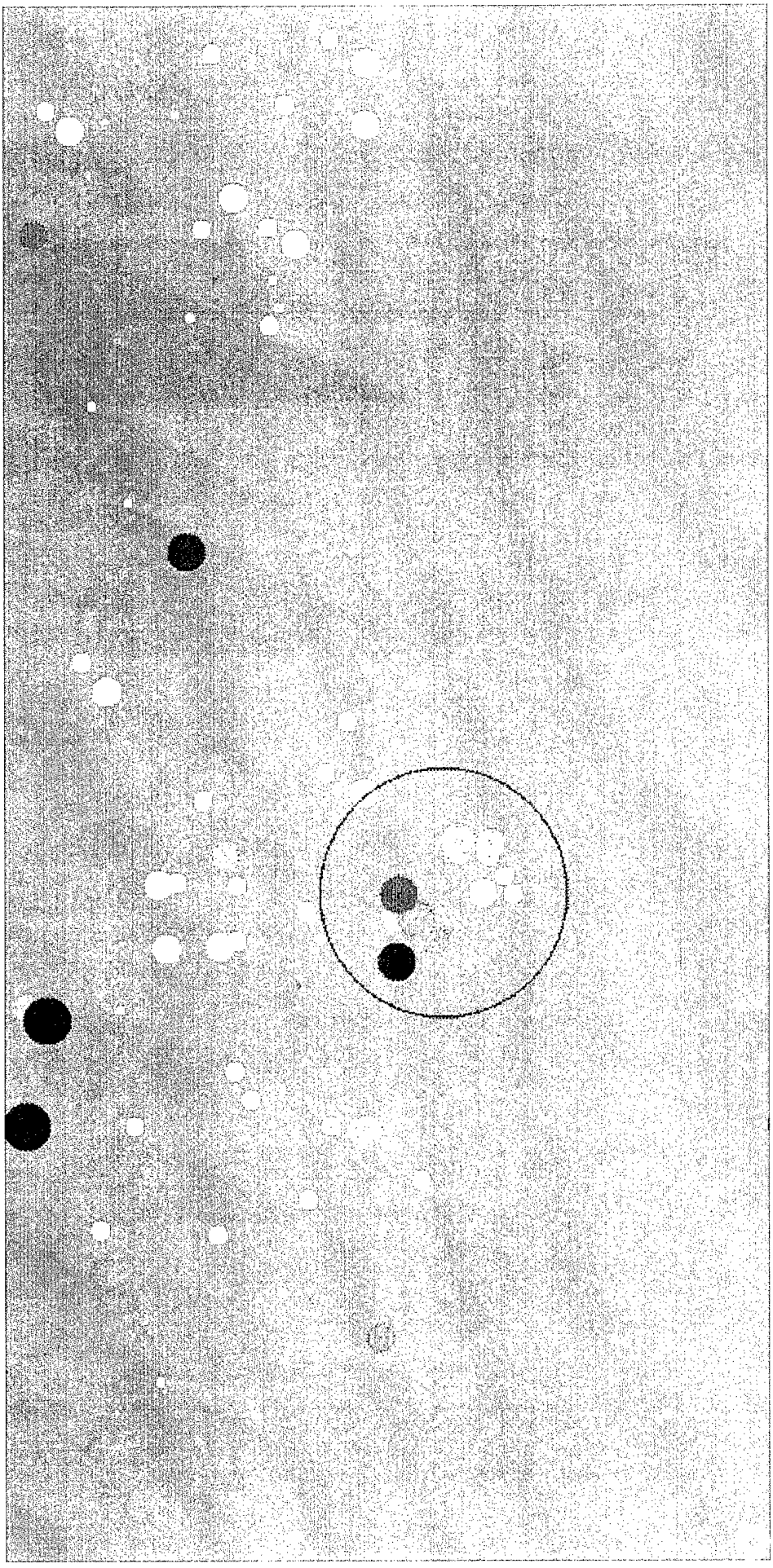
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.14: Classified mammogram #c18o.





EXAMPLE DSSA CLASSIFICATION RESULT

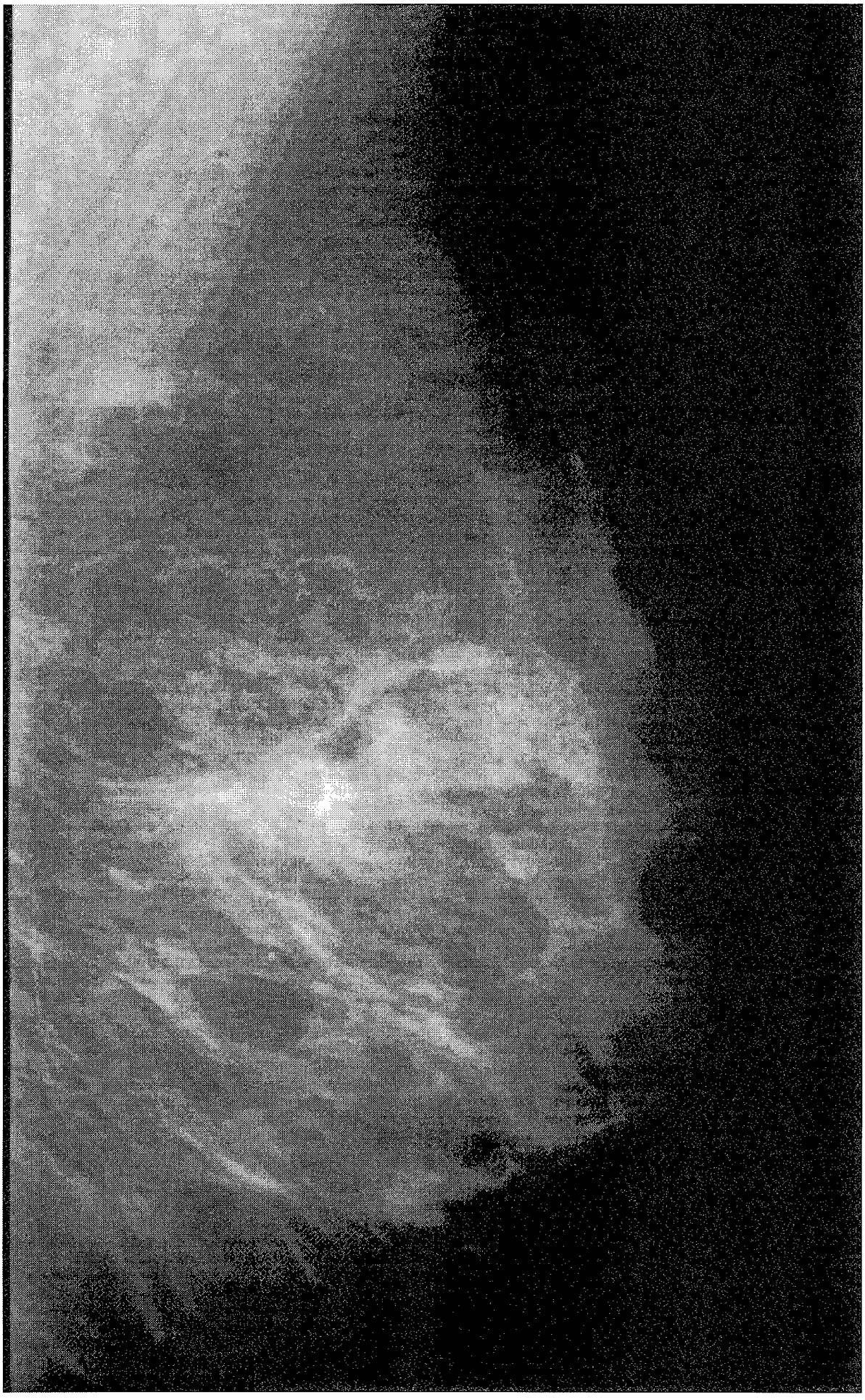
MAMMOGRAM #c18e

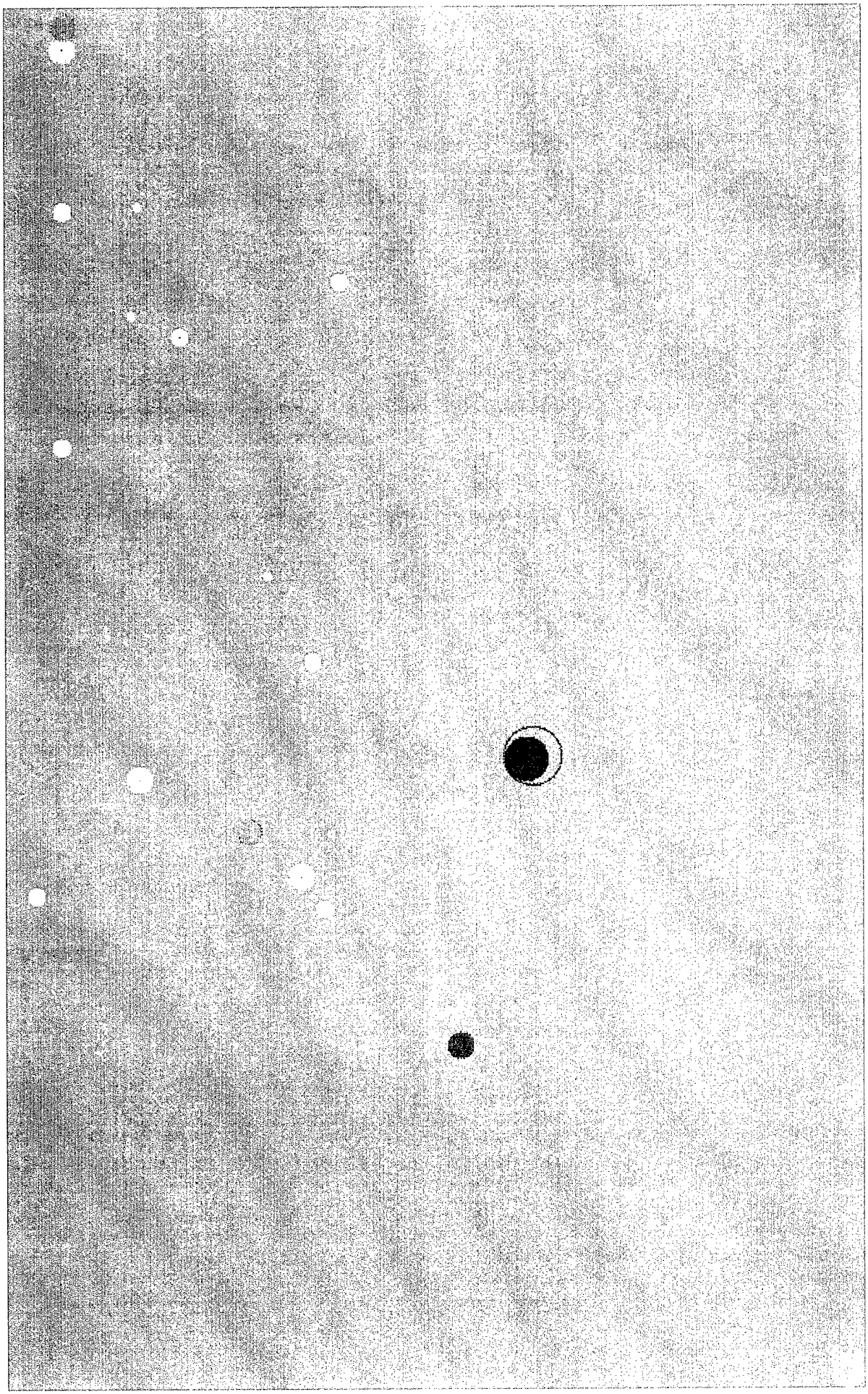
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.15: Classified mammogram #c18e.





EXAMPLE DSSA CLASSIFICATION RESULT

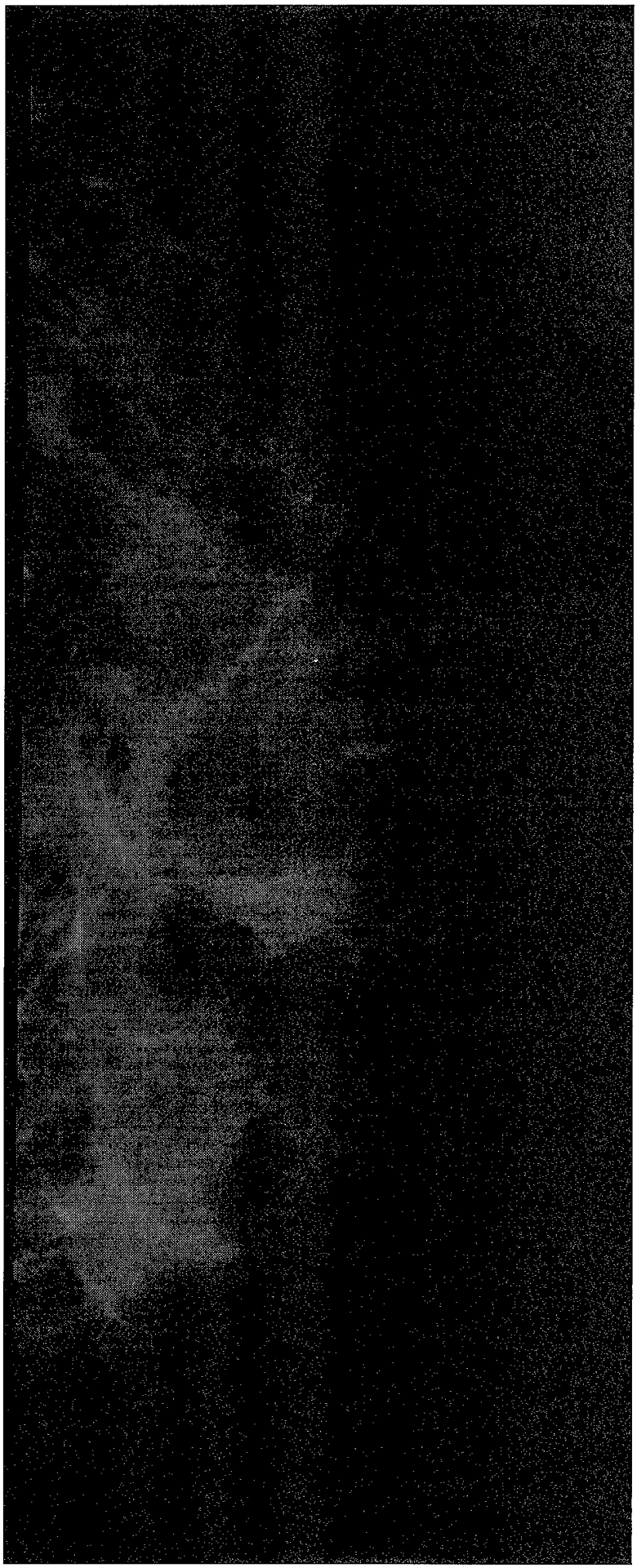
MAMMOGRAM #c19c

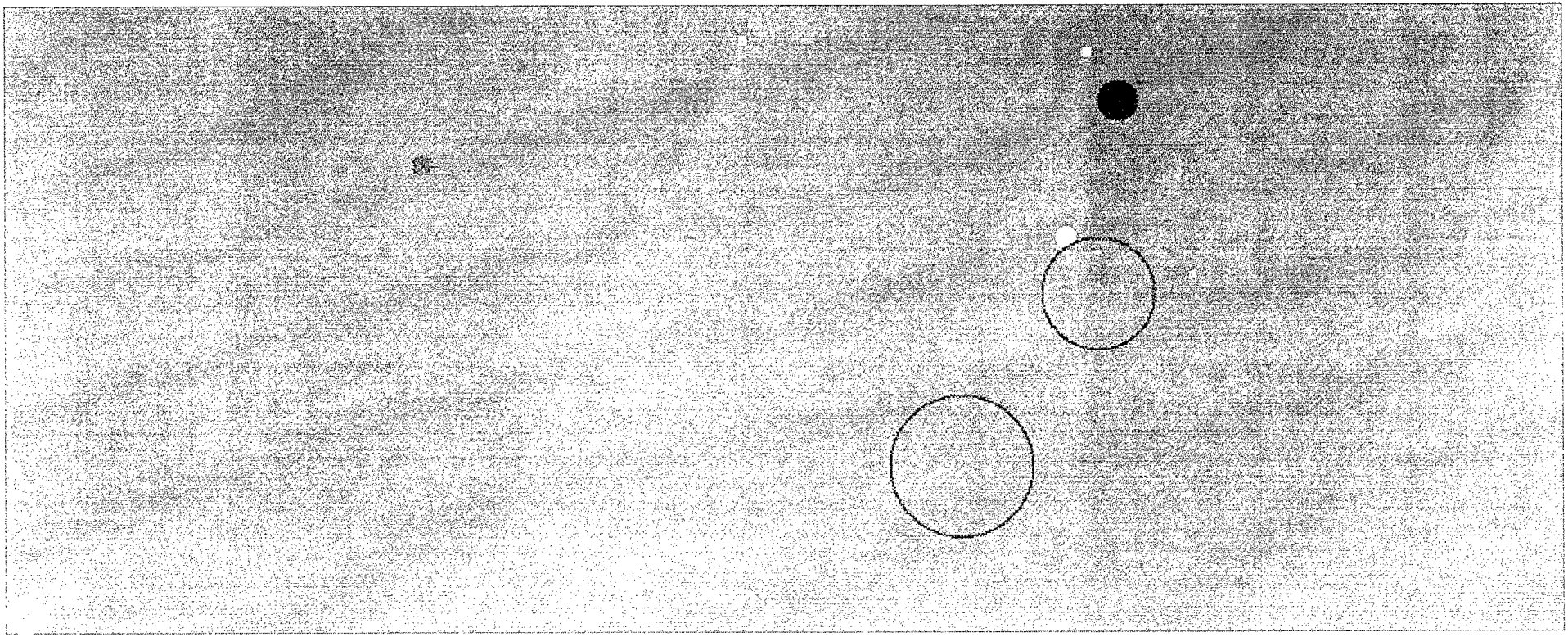
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	2
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	1

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.16: Classified mammogram #c19c.





EXAMPLE DSSA CLASSIFICATION RESULT

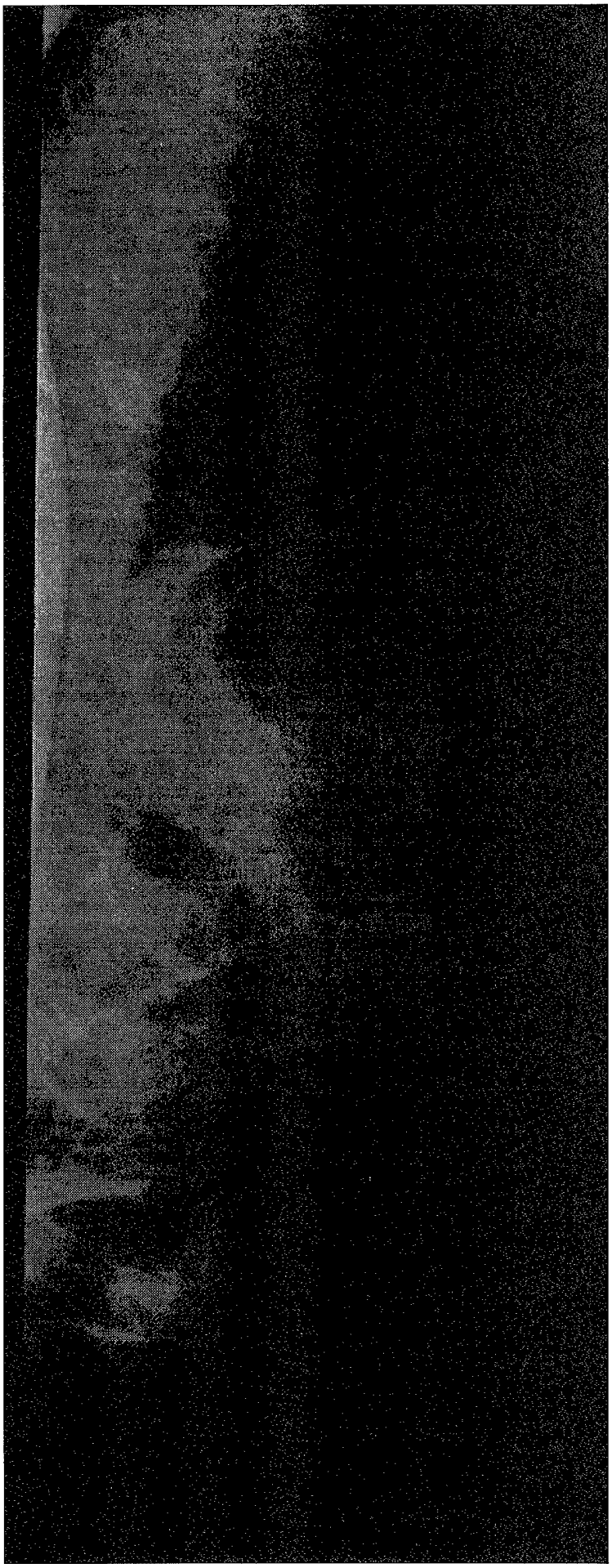
MAMMOGRAM #c19o

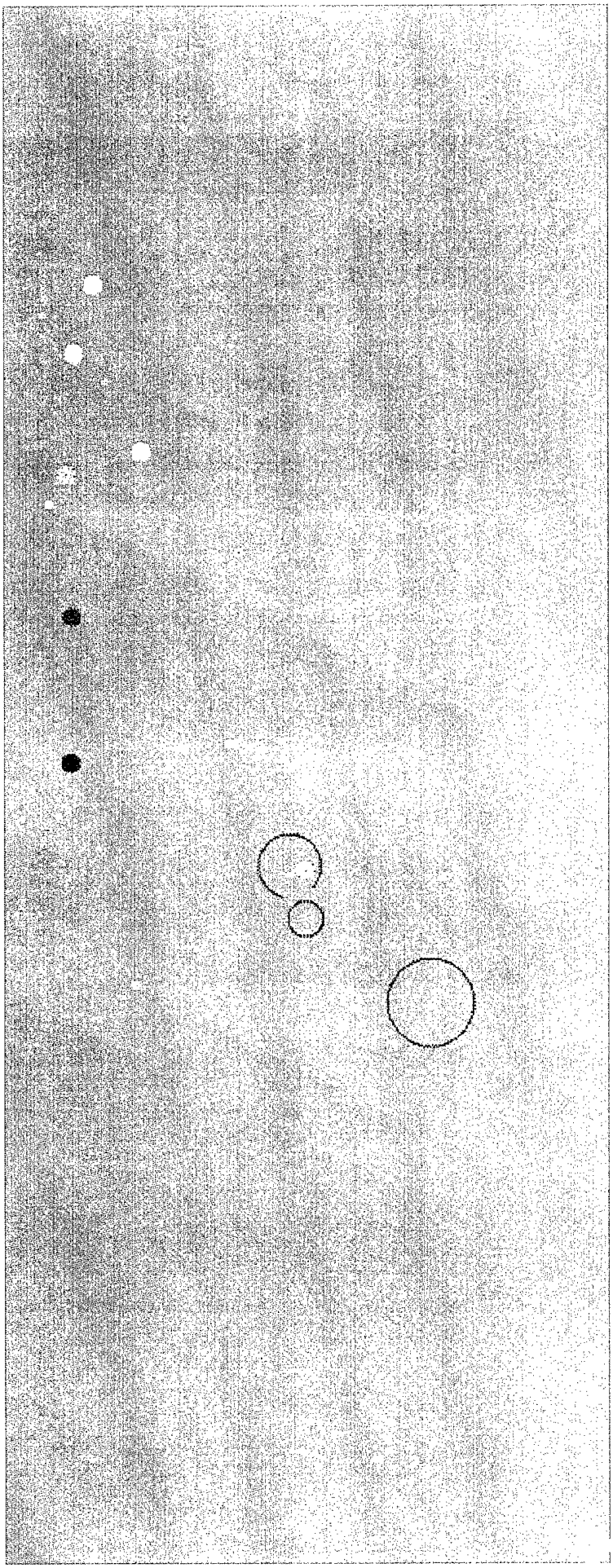
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	3
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	2

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.17: Classified mammogram #c19o.





EXAMPLE DSSA CLASSIFICATION RESULT

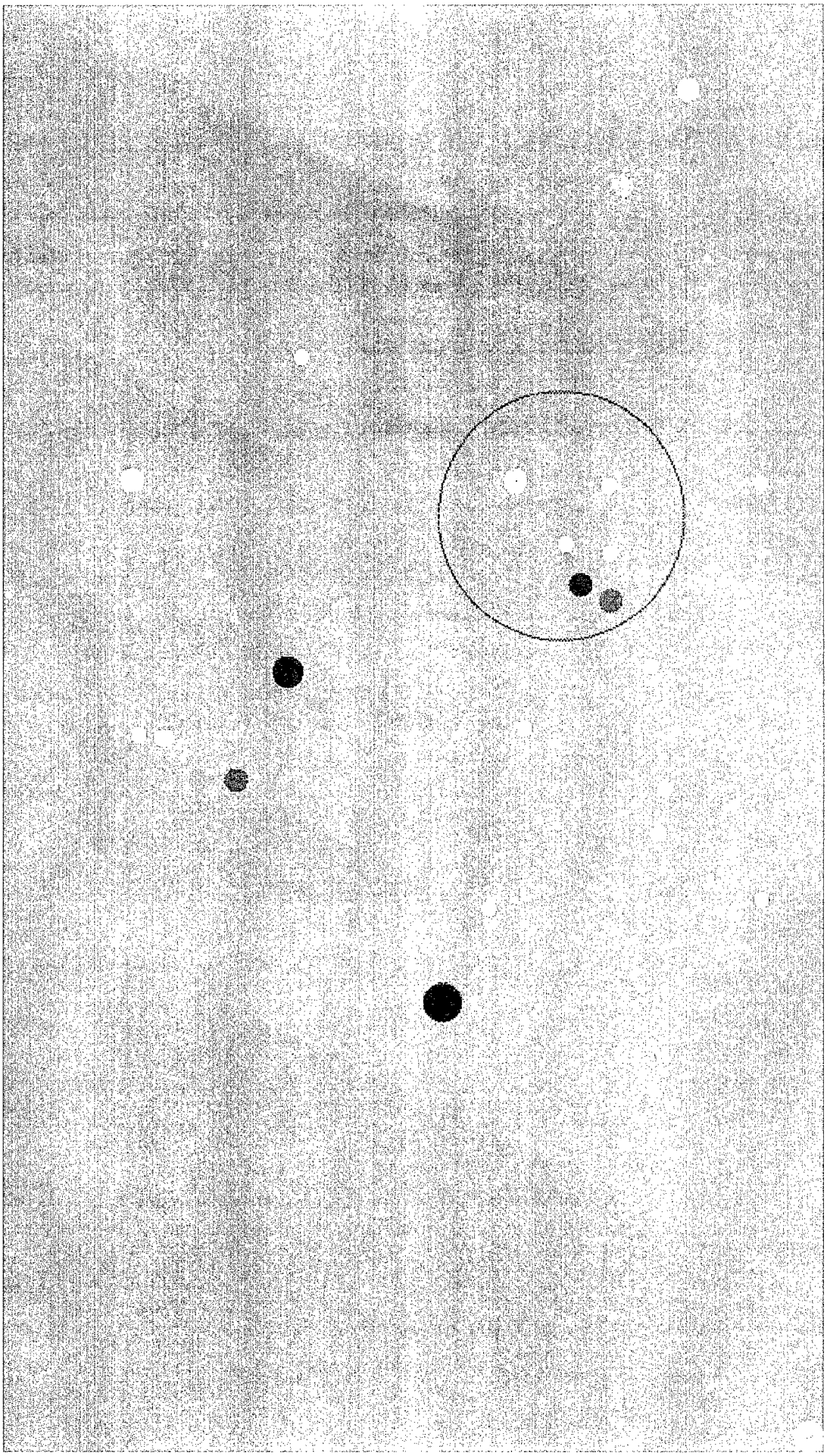
MAMMOGRAM #c20c

FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.18: Classified mammogram #c20c.



EXAMPLE DSSA CLASSIFICATION RESULT

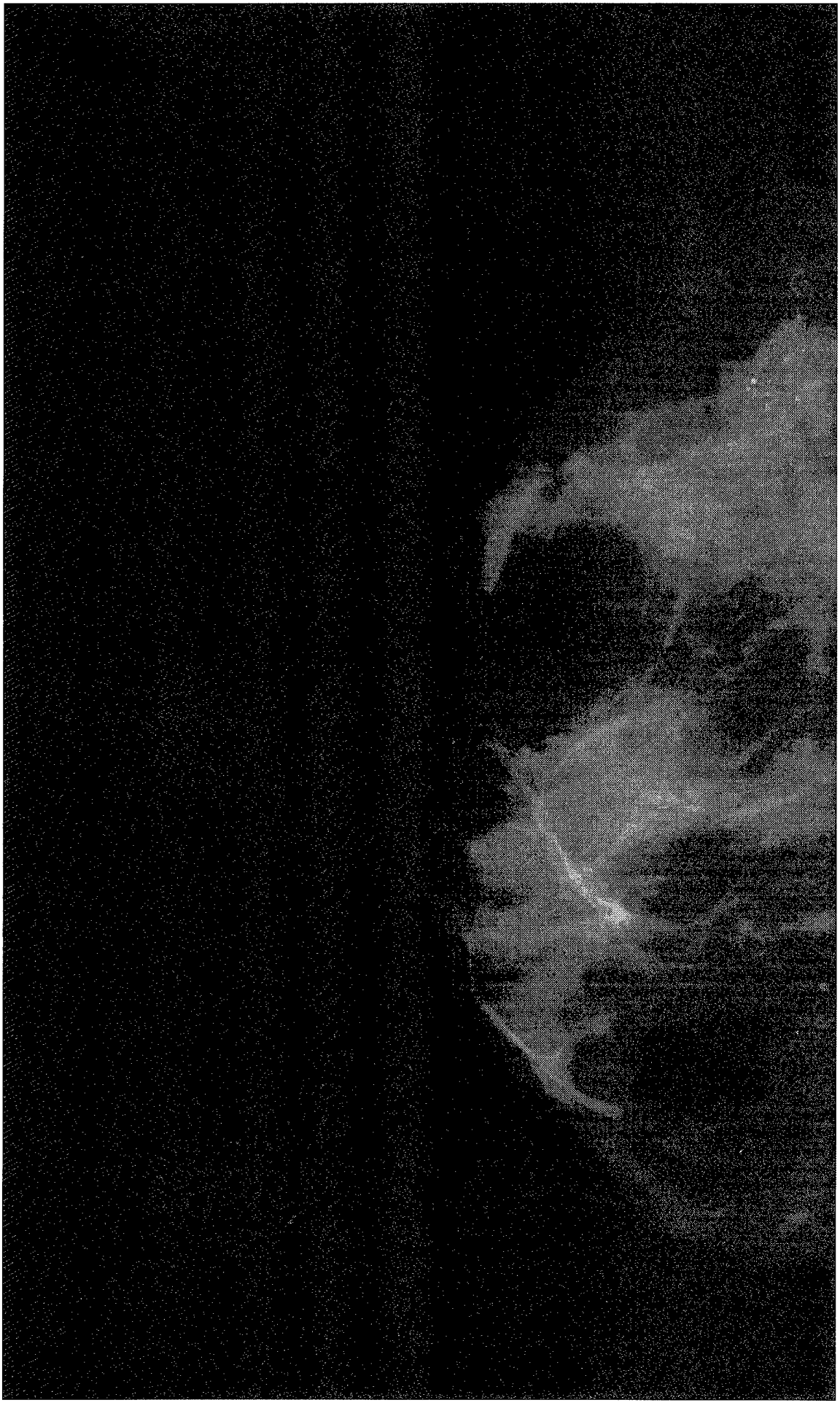
MAMMOGRAM #c21o

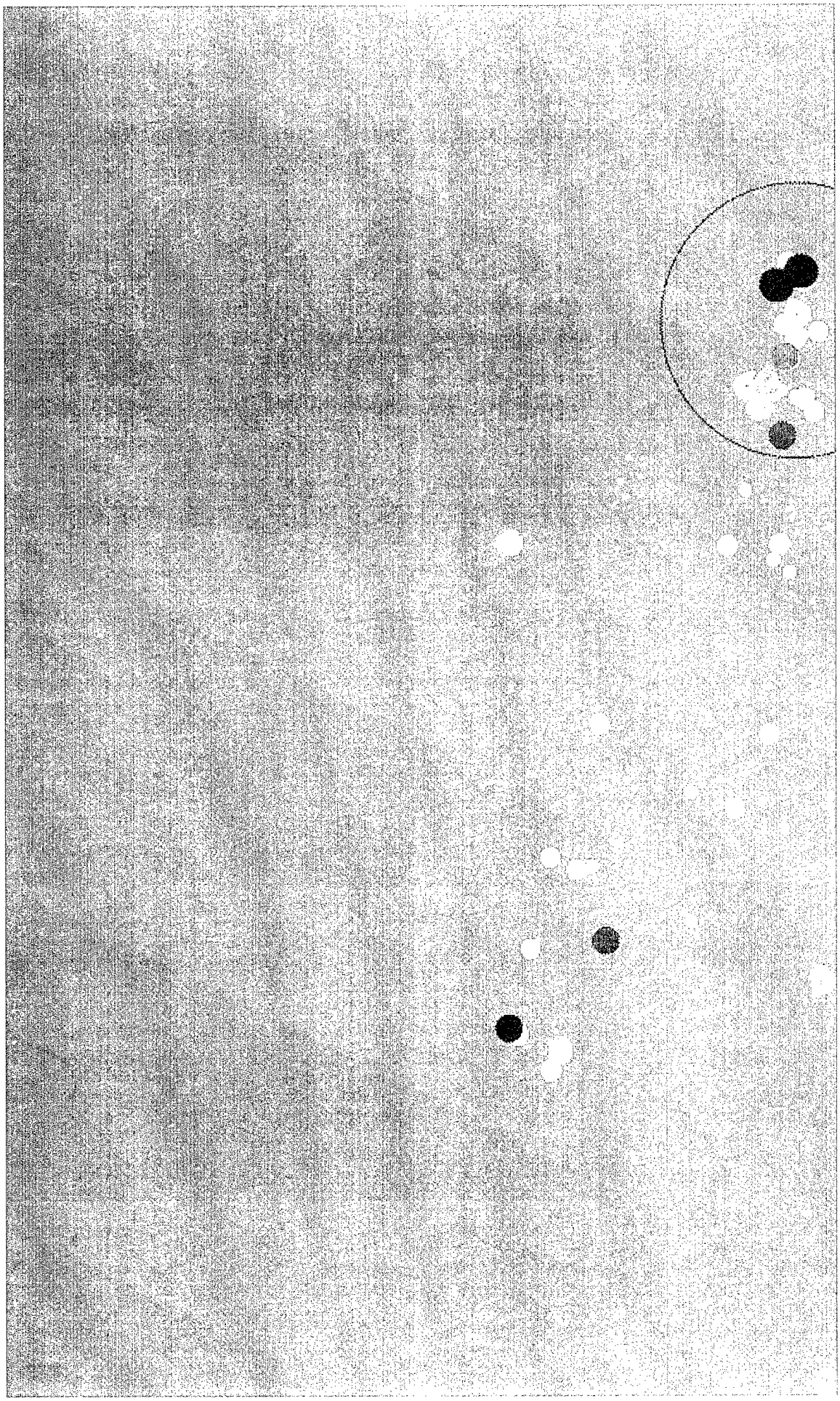
FIGURE WITH DETECTION OVERLAY SHOWN ON NEXT TWO PAGES

DSSA Classification Results		
Number of Human-Detected Clusters	=	1
Number of Computer-Detected Clusters	=	1
Number of Computer-Missed Clusters	=	0

DSSA Classifier Parameter Settings		
Number of DSSA Stages	=	37
Number of DSSA Templates per Stage	=	64
DSSA Template Size	=	5×5

Figure A.19: Classified mammogram #c21o.





Appendix B

DSSA and NN-RVQ Technical Overview

This appendix gives a brief overview of the fundamental problem of detection theory and explains the basic approach used by GTRI to develop the concepts of DSSA, which is the fundamental technology upon which the NN-RVQ classifier is based.

Decision Systems

A decision system ascertains which of several possible hypotheses best estimates the state of a sensed environment. Given measured data, for example, a decision system could provide an answer to the following question. Does the environment contain a target or threat, or does the environment contain only harmless clutter? The distinction between a target and a threat depends on the application. The presence of a target usually requires an aggressive response (e.g., in a dual-use medical application, tumor removal after positive diagnosis), and the presence of a threat usually requires an evasive response (e.g., avoidance after hazard detection in autonomous navigation systems). The clutter state almost always permits a passive response.

The maximum performance attainable by a decision system is influenced by the following three factors.

Factor 1: The intrinsic discriminability of the states of nature as determined by the phenomenon being observed and the class-conditional measurement probabilities over the observation space.

Factor 2: The quality and sensitivity of the sensor obtaining measured data from the observation space.

Factor 3: The allowable complexity of the decision system processing the measured data.

The first factor is governed by nature and cannot be influenced by the system designer (except, perhaps by the engineer's choice of phenomenon for observation and measurement). The second factor is determined by the state-of-the art in sensor technologies. The third factor represents practical complexity constraints imposed on the implementation of the decision system. In this research, the sensor is assumed to be a given, fixed system. The problem of maximizing decision system performance for a given level of sensor quality and a restricted level of tolerated complexity resides with the engineer and is the primary topic of this research report.

Decision Spaces

The decision space is the set of all possible measurements on which classification decisions could be based. The choice of which decision space to use for a particular application is an issue associated with sensor selection, the classifier design problem, and the imposed complexity limitations. The decision space could be the measured data space, a space associated with some set of features derived from the measured data, or, as is shown later in this appendix, a space of coded representations of the measured data. Generally, one seeks a decision space where data related to each class are clustered and the clusters associated with the different classes are well separated.

Mathematically, a decision space is a subset $\mathcal{X}$ of the k-dimensional real space $\mathcal{X} \subseteq \mathbf{R}^k$ or complex space $\mathcal{X} \subseteq \mathbf{C}^k$. A measurement $\boldsymbol{x}$ is a realization of a random vector $\boldsymbol{X}$ defined over $\mathcal{X}$. The dimensionality k of the $\boldsymbol{x} \in \mathcal{X}$ depends on the classification problem. In some applications, each measurement $\boldsymbol{x} = [x_1 x_2 \cdots x_k]^T$ is a vector, where the vector elements x_i are scalar features or measured samples (e.g., pixels in an image block); in other applications, the decision space is a scalar space.

The random vector $\boldsymbol{X}$ is described by a (class-independent) probability density function $p(\boldsymbol{x})$. To introduce class-conditional nomenclature, let $\mathcal{H} = \{H_m : m \in \mathcal{M}\}$ be a collection of M classes, where m is a *class label,* and $\mathcal{M}$ is the *class label set.* The class-conditional probability density function $p(\boldsymbol{x}|H_m)$ describes the probability of measuring $\boldsymbol{x}$ when the state of nature is H_m. The state of nature H_m occurs with an *a priori* probability $P(H_m)$. The *a posteriori* probability of the state of nature H_m after observation of a particular $\boldsymbol{x}$ is $P(H_m|\boldsymbol{x})$. From basic probability theory, the relationship

$$P(H_m|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|H_m)}{p(\boldsymbol{x})} P(H_m)$$

shows how the *a posteriori* probability $P(H_m|\boldsymbol{x})$ of the state of nature H_m is related to the *a priori* probability $P(H_m)$.

Decision Rules

A classifier is a rule that assigns each point in a decision space to a class that describes an estimated state of nature (a hypothesis). To be more precise, a classifier is a mapping

$$K : \mathcal{X} \mapsto \hat{\mathcal{H}}$$

that associates a hypothesis $K(\boldsymbol{x}) = \hat{H}_m$ with each $\boldsymbol{x}$ in $\mathcal{X}$, where the set of all possible hypotheses is denoted $\hat{\mathcal{H}} = \{\hat{H}_m : m \in \mathcal{M}\}$.

An M-ary decision rule partitions the decision space into a collection of M equivalence sets called *decision regions*, where all points in a particular decision region are assigned to the same hypothesis. Each decision region is specified by the inverse image $K^{-1}(\hat{H}_m) \subset \mathcal{X}$.

Joint Decision Rules

For future reference, note that the domain of a decision rule may be a Cartesian product space. If this is the case, the classifier would be a joint decision rule defined by a mapping

$$K : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_P \mapsto \hat{\mathcal{H}},$$

where the composite decision space is formed by the Cartesian product of a set of individual spaces $\mathcal{X}_p$ with $p \in \{1, 2, \ldots, P\}$.

A joint observation space may arise in applications involving sensor fusion or decentralized observations from a collection of remote sensors, or, as is shown later in this appendix, in combined estimation/quantization systems with product-code quantization structures. Surprisingly, in combined estimation/quantization systems, the use of a joint observation space is a simplifying feature that reduces the complexity of the nonparametric vector quantizer models that represent class-conditional probability density functions.

Bayes Classifiers

One formulation of a decision rule is based on the minimization of the expected "risk" or "cost" associated with classification errors, where specific costs are assigned to each of the possible errors. For applications with only clutter and target classes, the following cost assignments are appropriate; but first, let class labels be selected from the set $\mathcal{M} = \{T, C\}$ such that H_T represents a target state, H_C a clutter state, and $\hat{H}_T$ and $\hat{H}_C$ represent corresponding state estimates. Then, the following cost assignments are suitable for this two-class case.

$C(\hat{H}_T \mid H_C) = C_{fa}$	The *false alarm* penalty associated with incorrectly classifying clutter data as target data.
$C(\hat{H}_C \mid H_T) = C_{md}$	The *missed detection* penalty associated with incorrectly classifying

	target data as clutter data.
$C(\hat{H}_C\|H_C) = C_C$	The "cost" of correctly classifying clutter data.
$C(\hat{H}_T\|H_T) = C_T$	The "cost" of correctly classifying target data.

Because correct decisions usually incur no penalty, it is reasonable to assign $C_C = 0$ and $C_T = 0$.

With these cost assignments, the conditional false alarm risk for a given measured observation $\boldsymbol{x}$ is

$$\begin{aligned} R(\hat{H}_T|\boldsymbol{x}) &= C_T P(H_T|\boldsymbol{x}) + C_{fa} P(H_C|\boldsymbol{x}), \\ &= C_{fa} P(H_C|\boldsymbol{x}), \end{aligned}$$

and the conditional missed detection risk is

$$\begin{aligned} R(\hat{H}_C|\boldsymbol{x}) &= C_{md} P(H_T|\boldsymbol{x}) + C_C P(H_C|\boldsymbol{x}), \\ &= C_{md} P(H_T|\boldsymbol{x}). \end{aligned}$$

More generally, for an arbitrary classifier K, the two-class conditional risk is given by

$$R\left(K|\boldsymbol{x}\right) = C\left(K(\boldsymbol{x})|H_T\right) P(H_T|\boldsymbol{x}) + C\left(K(\boldsymbol{x})|H_C\right) P(H_C|\boldsymbol{x}).$$

A *Bayes classifier* K^* minimizes the conditional risk for each $\boldsymbol{x} \in \mathcal{X}$, i.e.,

$$K^*(\boldsymbol{x}) = \arg \min_{\hat{H}_m \in \hat{\mathcal{H}}} \left[R(\hat{H}_m|\boldsymbol{x}) \right]. \tag{B.1}$$

For the two-class case, the Bayes classifier can be reduced to a particularly simple form: select the target hypothesis if

$$C_{md} P(H_T|\boldsymbol{x}) > C_{fa} P(H_C|\boldsymbol{x}),$$

else, select the clutter hypothesis. The associated minimum risk, or *Bayes risk,* is

$$\begin{aligned} R^*\left(K^*\right) &= \int_{\boldsymbol{x} \in \mathcal{X}} R\left(K^*|\,\boldsymbol{x}\right) p(\boldsymbol{x}) d\boldsymbol{x}, \\ &= \int_{\boldsymbol{x} \in \mathcal{X}} \left\{ \min \left[\begin{array}{c} C_{fa} P(H_C|\boldsymbol{x}) \\ C_{md} P(H_T|\boldsymbol{x}) \end{array} \right] \right\} p(\boldsymbol{x}) d\boldsymbol{x}. \end{aligned}$$

The Bayes risk is a lower bound to the performance of all classifiers, i.e., $R \geq R^*$. Hence, the Bayes classifier is an optimal classifier from the perspective of minimum classification risk.

Bayes Classifier Complexity

A major drawback of the Bayes classifier is that in some cases it has a high level of implementation complexity. The root cause of this high cost is the complexity of the decision region boundaries induced by Bayes rule (given by Equation (B.1)). If the decision regions are not well behaved (e.g., if the $K^{-1}(\hat{H}_m) \subset \mathcal{X}$ are not connected nor convex subsets of $\mathcal{X}$), then in general the implementation costs of the Bayes classifier are high.

Decision Rule Implementations

Theoretically, a decision rule could be implemented for each and every point in the decision space by labeling the $\boldsymbol{x} \in \mathcal{X}$ with a class label. Such a (nonparametric) classifier would function as a table lookup operation: each measurement $\boldsymbol{x} \in \mathcal{X}$ is a different index into the classification table. Of course, a direct implementation of this approach would be impractical if the number of points in the decision space is large (excessive memory requirements) and impossible if the decision space is a continuum (infinite memory requirements).

In practice, decision rules are most often implemented by explicit characterization of the decision region boundaries. This approach is possible if the boundaries have simple forms (e.g., points, lines, hyperplanes, parabolas), which is most often the case with parametric classification methods; however, and especially so for nonparametric classifiers operating in high dimensional decision spaces, the decision region boundaries can be very complicated. Complicated decision region boundaries lead to high implementation costs.

The high costs of nonparametric classifiers have led to the development of simpler but suboptimal decision rules. The definitions of these suboptimal decision rules are influenced by a desire to restrict the structure of the decision region boundaries to simple implementable forms. This is done by the imposition of structure to form parametric classifiers.

Nonparametric Modeling

When decision region boundaries are used to implement a decision rule, the complexity of the classifier depends on the particular decision rule. The alternative (albeit impractical) implementation approach of assigning class labels to every possible measurement does not have this property. For a given observation space, all point decision rules are equally complex: the forms of the decision regions do not add any additional complexity to the table lookup classifier. This fact motivates the development of practical implementations of labeled point decision rules. The following describes two ways that a point decision rule can be implemented by using nonparametric modeling techniques.

Nearest Neighbor Modeling

To develop a practical decision system that efficiently implements a point decision rule, let's assume for the moment that a (very complicated) table lookup system exists that is capable of implementing any arbitrary point decision rule. Since its implementation complexity is proportional to the number of points in the decision space, let's reduce the complexity of this system by reducing the number of points that must be explicitly dealt with by the classifier. This can be done with a set of sample measurements and a nearest neighbor rule. Class labels are first assigned to each measurement in the sample set, then the entire space is implicitly classified via the nearest neighbor rule that maps all decision space points to the closest measurement contained in the sample set. This approach is called *nearest neighbor*

modeling.

The complexity of implementing a point decision rule with a nearest neighbor model is proportional to the sample set size, since the entire sample set must be searched for each sensor output that is to be classified.

Vector Quantizer Modeling

The performance of a nearest neighbor classifier increases as the number of measurements included in the sample set increases, but so do implementation costs. The complexity of a nearest neighbor classifier can be reduced by the use of *vector quantizer* (VQ) modeling. VQ modeling is identical to nearest neighbor modeling, except that, instead of using distances to sample measurements, distances to sample measurement representatives called *codevectors* are used. A vector quantizer uses a smaller set of codevectors to represent the larger set of measured samples. The set of codevectors is called a *codebook*. As before, the classification assignment of the preclassified codevectors is implicitly extended to the entire decision space with the use of a nearest neighbor rule.

Estimation and Quantization

This section examines the performance characteristics of estimation systems that use vector quantizer codebooks for nonparametric modeling. The performance limits and complexities of such systems are described. An understanding of this background material is essential for an appreciation of the advantages of the approach adopted by GTRI for the development of highly efficient and effective NN-VQ classifiers.

Vector Quantizers

The amplitudes of infinite precision points in $\mathcal{X}$ must undergo a many-to-one mapping to form a set of quantized representatives for all points in a decision space. This may be accomplished with the use of a vector quantizer which is a mapping $Q : \mathcal{X} \mapsto \mathcal{C}$ of each $\boldsymbol{x}$ in the continuous decision space $\mathcal{X}$ to a representation $\hat{\boldsymbol{x}}$ selected from the discrete set $\mathcal{C}$ of codevectors $\boldsymbol{y}(\cdot)$. The set $\mathcal{C} = \{\boldsymbol{y}(\cdot)\}$ is called the vector quantizer codebook.

In practice, a vector quantizer is realized as a composition of two functions: the *encoder* mapping $\mathcal{E} : \mathcal{X} \mapsto \mathcal{I}$ and a *decoder* mapping $\mathcal{D} : \mathcal{I} \mapsto \mathcal{C}$. The encoder is a many-to-one mapping from $\mathcal{X}$ onto the discrete *index set* $\mathcal{I}$. The decoder is a one-to-one mapping from $\mathcal{I}$ onto the codebook $\mathcal{C}$. Conventionally, and without loss of generality, the index set can be assumed to be the set $\mathcal{I} = \{0, 1, \ldots, N-1\}$, where N is the number of codevectors contained in the codebook. The inverse image of each $i \in \mathcal{I}$ defines a *partition cell* of the quantizer input space; this cell is denoted $S(i) = \mathcal{E}^{-1}(i) \subset \mathcal{X}$. The collection of cells forms the *partition* $\mathcal{P} = \{S(i) : i \in \mathcal{I}\}$ of the decision space $\mathcal{X}$.

The average distortion D of a vector quantizer is

$$D(Q) = E\left\{d\left(\boldsymbol{X}, Q(\boldsymbol{X})\right)\right\},$$

where $E\{\cdot\}$ is the expectation operator, and $d(\cdot,\cdot)$ is a distortion or similarity measure. The expected distortion is a quantitative measure of the "fidelity" of the representation of $\mathcal{X}$ generated by the codebook $\mathcal{C}$. Mean squared error, $d(\boldsymbol{x}, \hat{\boldsymbol{x}})) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$, where $\hat{\boldsymbol{x}} = Q(\boldsymbol{x}) = \boldsymbol{y}(i)$ for some $i \in \mathcal{I}$, is commonly used as a fidelity measure.

The *rate* of a vector quantizer with dimension k and codebook size N is defined to be

$$r = \frac{1}{k} \log_2 N.$$

In general, a VQ must store and search the entire codebook to find the best match between a measured observation and the set of codevectors. Thus, a vector quantizer's implementation complexity is proportional to its codebook size. Although the fidelity of a VQ codebook increases with increasing N, in terms of dimension and rate, the codebook size and associated VQ complexity increases exponentially with the product of rate and dimension, that is, $N = 2^{kr}$.

Vector Quantizer Design Methods

The usual approach used to design VQ codebooks for data compression applications is called the Linde, Buzo, and Gray (LBG) algorithm [2], or the Generalized Lloyd Algorithm (GLA) [3]. The LBG algorithm is basically the same as the previously developed k-means algorithm used for data clustering [4] and is based on the use of a set of sample measurements of the observation space called a *training set.* The training phase is performed off-line to generate a codebook that provides a good representation of the training set. Once a VQ codebook has been obtained and the VQ codevectors have been preclassified by assigning class labels, the codebook is used on-line to classify new measured observations via the nearest neighbor rule.

Bayes Classification of Quantized Measurements

A quantized data classifier K_Q can operate on the output of a quantizer by assigning a class label to each codevector $\boldsymbol{y}(i)$ in the quantizer codebook $\mathcal{C}$. Since a one-to-one decoder mapping $\mathcal{D}$ associates a unique index $i \in \mathcal{I}$ with each codevector $\boldsymbol{y}(\cdot) \in \mathcal{C}$, however, the classifier K_Q may also be defined on the index set $\mathcal{I}$ of the quantizer, i.e., $K_Q : \mathcal{I} \mapsto \hat{\mathcal{H}}$. In this case, note that $K_Q(i) = K_Q(\mathcal{E}(\boldsymbol{x}))$, that is, the classifier classifies coded representations of measured observations.

Let's determine the relationship between a Bayes classifier that operates on quantized and coded data and a Bayes classifier that operates directly on measured data. In the following,

assume without loss of generality that the set of class labels is the set $\mathcal{M} = \{1, 2, \ldots, M\}$. The conditional risk of classifying the state of nature as $\hat{H}_m$ for a given $\boldsymbol{x}$ is

$$R(\hat{H}_m|\boldsymbol{x}) = \sum_{n \in \mathcal{M}} C(\hat{H}_m|H_n) P(H_n|\boldsymbol{x}).$$

Next, given an arbitrary quantizer mapping Q, consider all $\boldsymbol{x}$ such that $E(\boldsymbol{x}) = i$, that is, all $\boldsymbol{x}$ in the VQ partition cell $S(i)$. The expected conditional risk for the quantized data classifier when it is known that $\boldsymbol{X} \in S(i)$ is

$$\begin{aligned} R_Q(\hat{H}_m|i) &= E\{R(\hat{H}_m|\boldsymbol{X} \in S(i))\}, \\ &= \int_{\boldsymbol{x} \in S(i)} R(\hat{H}_m|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}, \\ &= \int_{\boldsymbol{x} \in S(i)} \left[\sum_{n \in \mathcal{M}} C(\hat{H}_m|H_n) P(H_n|\boldsymbol{x}) \right] p(\boldsymbol{x}) d\boldsymbol{x}, \\ &= \sum_{n \in \mathcal{M}} C(\hat{H}_m|H_n) \int_{\boldsymbol{x} \in S(i)} P(H_n|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}. \end{aligned}$$

Hence,

$$R_Q(\hat{H}_m|i) = \sum_{n \in \mathcal{M}} C(\hat{H}_m|H_n) P(H_n|i),$$

where, by definition,

$$P(H_n|i) = \int_{\boldsymbol{x} \in S(i)} P(H_n|\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}.$$

For an arbitrary quantized data classifier K_Q, the conditional risk is

$$R_Q(K_Q|i) = \sum_{n \in \mathcal{M}} C(K_Q(i)|H_n) P(H_n|i).$$

The quantized data Bayes classifier minimizes the conditional risk for each $i \in \mathcal{I}$,

$$K_Q^*(i) = \arg \min_{\hat{H}_m \in \hat{\mathcal{H}}} \left[R_Q(\hat{H}_m|i) \right].$$

The associated minimum Bayes risk is

$$\begin{aligned} R^*(K_Q^*) &= \sum_{i \in \mathcal{I}} R(K_Q^*|i) P(i), \\ &= \sum_{i \in \mathcal{I}} \left\{ \min_{\hat{H}_m \in \hat{\mathcal{H}}} \left[\sum_{n \in \mathcal{M}} C(\hat{H}_m|H_n) P(H_n|i) \right] \right\} P(i). \end{aligned}$$

The quantized data classifier mapping induces a partition of $\mathcal{I}$ into common class subsets $K_Q^{-1}(\hat{\mathcal{H}}_m) \subset \mathcal{I}$ and also induces a partition of the codebook into subsets $\mathcal{D}(K_Q^{-1}(\hat{\mathcal{H}}_m)) \subset \mathcal{C}$.

Nearest Neighbor VQ Classifiers

Although imprecise VQ representations of measured data entail some loss of estimation performance with respect to the use of exact observations, as the fidelity of the VQ representation increases, it is reasonable to expect that the performance of the discrete system will approach the performance of the ideal infinite precision system. More specifically, we have seen that the conditional Bayes risk associated with an unquantized data classifier is

$$R^*(K^*|\boldsymbol{x}) = \sum_{n\in\mathcal{M}} C(K^*(\boldsymbol{x})|H_n)P(H_n|\boldsymbol{x}),$$

and the conditional Bayes risk of a quantized data classifier is

$$R^*_Q(K^*_Q|i) = \sum_{n\in\mathcal{M}} C(K^*_Q(i)|H_n)P(H_n|i),$$

where i is a VQ-coded representation of the measurement $\boldsymbol{x}$. These expressions are similar in form, and they are similar in value, in that the associated unconditional risk $R^*_Q(K^*_Q)$ converges to $R^*(K^*)$ in the limit as the number of codevectors N gets large, i.e., $R^*_Q \mapsto R^*$ as $N \mapsto \infty$. Thus, a Bayes nearest neighbor classifier with VQ modeling can asymptotically achieve the Bayes risk of an unquantized data classifier. But increasing the codebook size also increases the computational speed, memory, and bandwidth requirements of the system that implements the NN-VQ classifier.

One of the research objectives assigned to GTRI was the development of reduced complexity classifiers. The complexity of a quantized data classifier is proportional to the number of codevectors used by the quantizer; so, if a simpler NN-VQ classifier is desired, one simplification method is to reduce the codebook size. Hence, the performance characteristics of quantized data classifiers as the codebook size is reduced is also of interest.

An analysis shows that, if the number of codevectors is reduced but still remains "sufficiently large," then the expected risk of the nearest neighbor classifier is no worse than only twice the classification risk of the Bayes classifier that operates on unquantized measurements [5]. That is, $R^*_Q \leq 2R^*$. Thus, as long as the codebook is "large enough," then NN-VQ performance is guaranteed to be within the limits $2R^* \geq R^*_Q \geq R^*$.

Since NN-VQ performance can (with appropriate VQ codebook design procedures) approach the lower bound R^* as N is increased, GTRI offers the following observation that may provide an estimate for the codebook size that is required for R^*_Q to be close to R^*.

A Note on NN-VQ Codebook Size

Ideal (but fictitious) sensors generate perfect measurements with absolute precision. With infinitely precise measurements, it is possible for $[x_1 \cdots x_k]^T$ and $[(x_1+\epsilon)\cdots(x_k+\epsilon)]^T$ to provide substantially different classification information no matter how small ϵ is, and thus it is necessary to label each and every point in $\mathcal{X}$ to achieve Bayes risk. If measurements are influenced by sensor and system noise, however, then there exists an uncertainty region about

each measurement. The size of this uncertainty region varies as a function of the noise power $\mathcal{N}_s$. This implies that there exists a positive $\epsilon_{\mathcal{N}_s}$ such that $[x_1 \cdots x_k]^T$ and $[(x_1+\epsilon) \cdots (x_2+\epsilon)]^T$ provide essentially the same amount of classification information whenever $\epsilon < \epsilon_{\mathcal{N}_s}$. If a decision space is bounded (has compact support), a question related to the required codebook size is: how many uncertainty regions can be packed into a given decision space (assuming that both the decision space and uncertainty regions have volume)? An answer to this question may provide an estimate for the number of codevectors required to have R_Q^* close to the Bayes risk R^* of the noisy sensor system. A similar packing problem in communication theory shows that, if the noise is additive white Gaussian noise, then

$$\left(\frac{\mathcal{P}_s + \mathcal{N}_s}{\mathcal{N}_s}\right)^{\frac{k}{2}} \tag{B.2}$$

uncertainty regions can be packed into a k–dimensional spherical decision space [6], where $\mathcal{P}_s$ is the sensor power level. It is conjectured by the principal investigator that, for sensor systems subject to noise, a codebook size in the neighborhood of of the value expressed by (B.2) is sufficient for achieving a NN-VQ performance level that is close to the Bayes risk obtainable with unquantized measurements. Unfortunately, this required codebook size can be very large. For example, if the signal-to-noise ratio (SNR) is 20 dB and the dimensionality of the observation space is 192, then

$$\begin{aligned}\left(\frac{\mathcal{P}_s + \mathcal{N}_s}{\mathcal{N}_s}\right)^{\frac{k}{2}} &= \left(10^{\frac{\text{SNR}}{10}} + 1\right)^{\frac{192}{2}}, \\ &= \left(10^{\frac{20}{10}} + 1\right)^{\frac{192}{2}}, \\ &\approx 10^{192},\end{aligned}$$

which is astronomically large. But as is shown below, there are ways to implicitly deal with VQ structures where the equivalent codebook sizes are indeed very large.

VQ Implementation Complexity

Since VQ complexity is proportional to $N = 2^{kr}$, the use of VQ modeling can also have exorbitant implementation costs if the dimension or rate is large. There is, however, a variant of VQ called *residual vector quantization* (RVQ) with complexity proportional to the much simpler cost function $2 \times kr$ but with an effective codebook size of $N = 2^{kr}$. Hence, RVQs offer implementation advantages in the application of VQ modeling for nearest neighbor classification. For example, an astronomically large equivalent codebook size of 2^{192} codevectors can be designed and manipulated using a RVQ with a cost function that is proportional to only $2 \times 192 = 384$.

RVQ modeling is the fundamental approach used by GTRI in the development of a *nearest neighbor residual vector quantizer* (NN-RVQ) classifier. The innovative features of the NN-RVQ permit a large number of codevectors to be efficiently and effectively used to model class-conditional probability density functions. Details of the novel features of the NN-RVQ are given next.

Residual Vector Quantization

Vector quantizers are classified into two major categories: structured and unstructured. Unstructured VQs give the best performance for a given rate and dimension, but their implementation costs in terms of both computation and memory increase exponentially with the product of rate and dimension. Structured VQs, on the other hand, give poorer performance than an unstructured VQ for a given rate-dimension product, but they may actually provide better performance for a given level of implementation complexity. Structured VQs are further categorized as to whether the imposed structural constraints return computation or memory savings, or both. One of the most important structurally constrained VQs which reduces both memory and computational cost is a successive approximation VQ structure called residual VQ [7, 8, 9].

Fixed Rate RVQ

A successive approximation residual vector quantizer consists of a sequence of relatively small VQ stages, where each stage encodes the residual error of the prior stage. That is, after the first stage VQ, a residual vector is formed by subtracting the codevector selected from the first stage codebook from the original input vector. The second VQ stage then finds the best match in its codebook of "residual" codevectors and outputs the corresponding index. This process is repeated on the residuals formed from the second stage and so forth for a total of P stages. The P-tuple of indices output by the multiple stage RVQ encoder is transmitted to the RVQ decoder where table lookups are performed in corresponding reconstruction stage codebooks. The decoder output set of stage codevectors are vector summed to produce the RVQ representation of the original input vector.

To be more mathematical in our description, the pth stage of an RVQ is a k-dimensional vector quantizer defined by the mapping $Q_p : \mathcal{X} \mapsto \mathcal{C}_p$, where

- $p \in \{1, 2, \ldots, P\}$ is the *stage index,*
- $C_p = \{\boldsymbol{y}_p(0), \boldsymbol{y}_p(1), \ldots, \boldsymbol{y}_p(N_p - 1)\}$ is the pth stage *codebook,*
- $\boldsymbol{y}_p(i_p) \in \mathcal{X}$ are pth stage *codevectors,* and
- i_p is an *index* in the pth stage *index set* $\boldsymbol{I}_p = \{0, 1, \ldots, N_p - 1\}$.

Residual quantizer stage mappings are collectively equivalent to a single stage mapping $\mathbf{Q} : \mathcal{X} \mapsto \mathbf{C}$, where

- $\mathbf{C}$ is the *direct sum codebook* $\{\mathbf{y}(\mathbf{i}) : \mathbf{i} \in \mathbf{I}\}$,
- $\mathbf{y}(\mathbf{i})$ is a *direct sum codevector* $\mathbf{y}(\mathbf{i}) = \boldsymbol{y}_1(i_1) + \boldsymbol{y}_2(i_2) + \cdots + \boldsymbol{y}_P(i_P)$, and
- $\mathbf{i} = (i_1 i_2 \cdots i_P)$ is a *P-tuple index* contained in $\mathbf{I} = \boldsymbol{I}_1 \times \boldsymbol{I}_2 \times \cdots \times \boldsymbol{I}_P$.

The direct sum codebook is the direct sum of the stage codebooks, $\mathbf{C} = \boldsymbol{C}_1 + \boldsymbol{C}_2 + \cdots + \boldsymbol{C}_P$.

Variable Rate RVQ

Variable rate RVQ is a type of RVQ where different numbers of stages are used on different input vectors. Variable rate RVQ has certain efficiency advantages in the application of RVQ to data compression. In the sonar image classification problem, variable rate RVQs allow greater flexibility in generating RVQ codebooks for nonparametric modeling of class-conditional probability density functions. The variable rate codebook structure allows the RVQ codevector density in the decision space to vary to more accurately model the characteristics of the class-conditional probability density functions.

One way variable rate RVQ can be accomplished is by determining the energy of the residual vector between the RVQ stages. For example, comparative decision blocks can be be placed between the RVQ stages. A predetermined threshold value associated with each decision block and the corresponding input vector energy level at that stage can be compared to determine whether the following stage is used to encode the input vector. This approach permits various performance-complexity trade-offs to be controlled. By designing a large number of RVQ stages beforehand and then using only a smaller subset of stages in practice, one can prevent the complexity of the classification system from becoming too large. Variable rate RVQs also permit different stages to be used to encode different input vectors with varying energy levels. Various approaches for designing the variable rate RVQ decision thresholds are described in References [10, 11].

RVQ Complexity

Residual vector quantizers are subject to two structural constraints: a sequential search encoder constraint and a direct sum codebook constraint. The purpose of the sequential search encoder constraint is to reduce computation requirements. The purpose of the direct sum codebook constraint, which is imposed at both the RVQ encoder and decoder, is to reduce memory requirements. As a result, the composite RVQ structure returns both computation and memory savings over single stage full search vector quantizers (VQs), and memory savings over tree structured VQs [2]. For example, if a k-dimensional RVQ with output rate r bits per sample consists of P stages with $N = 2^{kr/P}$ codevectors in each stage codebook, then the memory and computation costs of the RVQ are proportional to

$$\text{Complexity} \quad \sim \quad P2^{kr/P},$$

whereas the memory or computation cost, or both, of most other VQ structures (with the same equivalent codebook size) is proportional to

$$\text{Complexity} \quad \sim \quad 2^{kr}.$$

For example, if $P = 128$ and $N_p = 32$, then the complexity of the RVQ is proportional to 4096. The complexities of the other VQ structures with the same equivalent codebook

sizes are proportional to 2^{640}, which is a cost level requirement impossible to realize, or even imagine!

Most of the implementation efficiency of an RVQ over a full search VQ is obtained in going from one stage to two stages. The most efficient (in terms of memory and computation) RVQs are those with two or four codevectors per stage (the associated implementation costs are the same in both cases and are proportional to $2P$ for the two codevector case). Most empirical evidence, however, suggests that RVQs with many stages can be expected to give unsatisfactory performance [2]. These poor results for RVQs with many stages have proven, at least for some sources, to be more a result of the design method used to generate the stage codebooks and not necessarily inherent in the RVQ structure itself. The next section describes various methods used by GTRI to design RVQ codebooks and how previous research conducted by GTRI [9, 12, 13, 1, 10, 14] has improved the performance of RVQs when applied to the data compression problem.

RVQ Design

The following paragraphs describe the history of some problems encountered when multiple sets of vector quantizer codebooks were first designed for application to the RVQ data compression problem and how research pioneered by GTRI has solved many of these problems.

Single Stage VQ Codebook Design

In 1980, Linde, Buzo, and Gray [3] generalized Lloyd's Method I [15] for scalar quantizer design to develop a vector quantizer design algorithm, known variously as the LBG or Generalized Lloyd Algorithm (GLA). Although very similar, there is a significant difference between Lloyd's Method I and the GLA. Lloyd's method requires an analytically specified probability density function to describe the data source; but for many data sources encountered in practice, the multidimensional probability density function is either unknown or is not easily specified analytically. The GLA circumvents this difficulty by substituting a *training set* for the probability density function. This substitution is proven in the limit of large training set size to produce asymptotically equivalent designs [16].

Lloyd's Method I and the GLA are based on the principle of iteratively finding an optimal encoder for a fixed decoder and, vice versa, finding an optimal decoder for a fixed encoder. This approach is based on a more general design principle used to design both data compression systems and k-means classification systems [17, 18, 19, 20, 5]. Use of this method requires optimality conditions for the encoder and decoder when the other is held fixed. Let $\mathcal{O}_E$ represent a rule that is used to optimize the encoder when the decoder is held fixed, and let $\mathcal{O}_D$ represent a rule that is used to optimize the decoder when the encoder is held fixed. A general design procedure for determining encoder/decoder pairs that satisfy necessary conditions for optimality is as follows [18]:

GENERAL ITERATIVE DESIGN PROCEDURE

1. Select and hold fixed an initial decoder.
2. For the fixed decoder, use $\mathcal{O}_E$ to select an optimal encoder.
3. For the fixed encoder, use $\mathcal{O}_D$ to select an optimal decoder.
4. Compute the average distortion of the resulting code. If the average distortion falls below some predetermined value, or the relative change (from the previous iteration) in average distortion falls below some predetermined threshold, then STOP; otherwise, continue with step (2).

Steps (2) and (3) of the iterative design procedure only reduce or leave unchanged the average distortion. Since the distortion is bounded from below by zero, the monotonically nonincreasing sequence of distortion measures converges to a fixed point [21].

Using this design principle, one can describe the GLA as follows. Let $\mathcal{T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L\}$ be a training set of L sample vectors, where each sample vector $\boldsymbol{x}_l$ is drawn according to a probability density function p on $\mathbf{R}^k$. The GLA improves (in the sense of reducing the *average* distortion) a vector quantizer for the training set $\mathcal{T}$. The GLA starts with some initial codebook and then iterates by first replacing the codebook with the centroids of the training set vectors that are in each of the partition cells (this rule provides $\mathcal{O}_D$). The algorithm then determines a new partition by a nearest–neighbor mapping of the training set to the new codebook of centroids (this rule provides $\mathcal{O}_E$), and the entire process is repeated until a stopping criterion is satisfied.

Multiple Stage RVQ Codebook Design

The first method suggested for designing RVQs was sequential application of the GLA [7]. That is, the GLA is applied to design the first stage codebook, and this codebook is held fixed and used to form a set of residual training data for the second stage. The GLA is then applied to design the second stage codebook, and so on until all RVQ codebooks have been created. Although it has been shown that this greedy, sequential use of the GLA is nearly optimum for two-stage RVQs with moderated to high output rates [22], it is widely recognized that this design method is increasingly suboptimal as the number of stages increases. This phenomenon has resulted in sequential GLA-RVQs being limited in practice to only two or three stages [8, 2].

In Reference [1], necessary conditions for the joint optimality of RVQ stage codebooks were derived. Given necessary conditions for the optimality of RVQ encoders and decoders, application of the GLA design approach described above to design RVQs is straightforward (if the complexity of optimal encoding is tolerated). The difference, however, between the GLA for single stage quantizers and a similar algorithm for multiple stage RVQs is that there must be two interlaced iterative fixed point procedures: one for optimization of the encoder/decoder

pair, and another to simultaneously satisfy a certain conditional mean or centroid condition for all RVQ stages. In the second iterative procedure, each stage codebook is optimized while holding the codebooks of all other stages fixed. The new codevectors of the optimized stage satisfy the necessary centroid condition with respect to the fixed partitions and the fixed codebooks of the other stages. The codevector update procedure is then repeated for a different stage; however, the process of optimizing the codevectors of a different stage causes the first stage that was optimized to no longer satisfy the centroid condition. It is necessary to eventually return to each stage and repeat the process in round robin fashion. Since the changes made to the codebooks of each stage can only decrease or leave unchanged the average distortion of the RVQ (assuming a constant fixed partition), this iterative procedure converges to a fixed point. After this first fixed point has been reached (or approached sufficiently close), a new encoder/decoder iteration is performed (a new partition is selected), and the entire process is repeated until both processes converge to fixed points.

The approach to joint GLA RVQ design encounters some difficulties when the complexity of optimal (exhaustive) encoding is not tolerated. Optimal sequential search partitioning for an arbitrary set of direct sum codebooks often requires stage decision regions that are not connected or convex, while efficient sequential search encoders generally require well behaved stage partition cells. Proceeding with the joint design process with suboptimal encoding rules can lead to nonmonotonic design behavior, and occasionally, catastrophically poor performance results, especially if the RVQ has a high output rate or many stages.

These problems have led to the acceptance of various ad hoc procedures to stabilize the design process. For example, a "block" approach to joint RVQ optimization can be used. That is, only a subset of the RVQs are jointly optimized before proceeding to the next block of stages in the RVQ. This method was used to generate the RVQ codebooks discussed in References [11, 23, 24, 25]. As another example, if an increase in distortion does occur, the design process can be terminated without recourse [26]. Other researchers have proposed a "fuzzy" approach to sequential search encoder optimization [27].

Another ad hoc, but in some ways, more fitting approach to solving this design problem is the use of separate, and in general, different stage codebooks at the RVQ encoder and decoder [10, 14]. Direct sum encoder codebooks should be designed to give good sequential search performance. Direct sum decoder codebooks should be designed to satisfy conditions necessary for joint optimality. A design method for RVQ is required that is capable of providing separate encoder and decoder codebooks, but where the two sets of codebooks work together to provide good overall RVQ performance. Such a design method is described in detail in Reference [28].

This prior extensive work in RVQ codebook design has led to the development of very stable design procedures that permit RVQs with many, possibly hundreds of, stages to be designed and implemented.

Appendix C

List of Abbreviations

This list provides expansions for abbreviations and acronyms used in this report.

Abbreviation/Acronym	Explanation
AAA	Amphibious Assault Areas
AUV	Autonomous Underwater Vehicles
CAD	Computer Assisted Detection
DSSA	Direct Sum Successive Approximation
EO	Electro-Optical
GTRI	Georgia Tech Research Institute
IR	Infrared
MS	Multispectral
NCCOSC	Naval Command, Control, and Ocean Surveillance Center
NRaD	Research, Development, Test & Evaluation Division
NSWC	Naval Surface Warfare Center
ONR	Office of Naval Research
PACS	Picture Archival System
PI	Principal Investigator
SAR	Synthetic Aperture Radar
RVQ	Residual Vector Quantization
UAV	Unmanned Aerial Vehicles
UUV	Unmanned Underwater Vehicles
VQ	Vector Quantization
VSW	Very Shallow Water

Bibliography

[1] C. F. Barnes and R. L. Frost. Vector quantizers with direct sum codebooks. *IEEE Transactions on Information Theory*, 39(2):565–580, March 1993.

[2] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer, Boston, 1992.

[3] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, January 1980.

[4] E. W. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21(3):768, abstract, 1965.

[5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[6] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37:10–21, January 1949.

[7] B. H. Juang and A. H. Gray. Multiple stage vector quantization for speech coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 597–600, April 1982.

[8] J. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11):1551–1588, November 1985.

[9] C. F. Barnes. *Residual Quantizers.* PhD thesis, Brigham Young University, Provo, UT, December 1989.

[10] C. F. Barnes and E. J. Holder. Classified variable rate residual vector quantization applied to image subband coding. In J. A. Storer and M. Cohn, editors, *Proceedings of the Data Compression Conference*, pages 272–281, Snowbird, UT, March 30–April 2, 1993.

[11] F. Kossentini, M. J. T. Smith, and C. F. Barnes. Image coding with variable rate RVQ. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 369–372, San Francisco, CA, March 23–26, 1992.

[12] C. F. Barnes and R. L. Frost. Necessary conditions for the optimality of residual vector quantizers. In *Abstracts of the IEEE International Symposium on Information Theory*, page 34, San Diego, CA, January 14–19, 1990.

[13] C. F. Barnes and R. L. Frost. *Image Mathematics and Image Processing*, volume 84 of *Advances in Electronics and Electron Physics*, chapter Residual Vector Quantizers with Jointly Optimized Code Books, pages 1–59. Academic Press, 1992. Edited by P. W. Hawkes.

[14] C. F. Barnes and E. J. Holder. Successive approximation quantization with generalized decoding for wavelet transform image coding. In *Proceedings of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 533–537, November 1993.

[15] J. Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(2):7–12, March 1960.

[16] M. J. Sabin and R. M. Gray. Global convergence and empirical consistency of the generalized Lloyd algorithm. *IEEE Transactions on Information Theory*, 32(2):148–155, March 1986.

[17] G. Gabor and Z. Györfi. *Recursive Source Coding.* Springer-Verlag, New York, 1986.

[18] T. Fine. Properties of an optimum digital system and applications. *IEEE Transactions on Information Theory*, pages 287–296, 1964.

[19] J. D. Gibson and T. R. Fischer. Alphabet-constrained data compression. *IEEE Transactions on Information Theory*, pages 443–457, May 1982.

[20] J. G. Dunham. An iterative theory for code design. *IEEE International Symposium on Information Theory. Abstracts of Papers*, pages 89–90, September 1983.

[21] R. M. Gray, J. C. Kieffer, and Y. Linde. Locally optimal block quantizer design. *Inform. and Control*, 45(2):178–198, May 1980.

[22] D. H. Lee. *Asymptotic Quantization Error and Cell-Conditioned Two-Stage Vector Quantization.* PhD thesis, The University of Michigan, 1990.

[23] F. Kossentini, M. J. T. Smith, and C. F. Barnes. Large block RVQ with multipath searching. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 5, pages 2276–2279, San Diego, CA, May 10–13, 1992.

[24] F. Kossentini, M. J. T. Smith, and C. F. Barnes. Residual VQ with state prediction: A new method for image coding. *SPIE/SPSE Symposium on Electronic Imaging: Science and Technology*, February 24–March 1, 1991.

[25] F. Kossentini, M. J. T. Smith, and C. F. Barnes. A perspective view of finite state binary residual VQ. *1991 International Symposium on Circuits and Systems*, pages 300–303, June 11–14, 1991.

[26] W.-Y. Chan, S. Gupta, and A. Gersho. Enhanced multistage vector quantization by joint codebook design. *IEEE Transactions on Communications*, 40(11):1693–1697, November 1992.

[27] D. Miller and K. Rose. An improved sequential search multistage vector quantizer. In *Proceedings of the Data Compression Conference*, pages 12–21, Snowbird, UT, March 30–April 2, 1993.

[28] C. F. Barnes. Adaptive successive approximation quantization of image waveforms with efficient codebook updates. *IEEE International Conference on Image Processing*, III:876–880, November 1994.

[29] D. Burton. Acoustic transient classification of passive sonar signals by using vector quantization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1493–1496, Toronto, Ontario, Canada, 1991.

[30] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, January 1967.

[31] E. E. Hilbert. Cluster compression algorithm: A joint clustering/data compression concept. Publication 77–43, Jet Propulsion Laboratory, Pasadena, CA, December 1977.