UNCLASSIFIED

AD NUMBER

AD860060

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; JUL 1969. Other requests shall be referred to Army Enineer Topographic Laboratory, Attn: ETL-GS [Geographic Information Systems], Fort Belvoir, VA 22060.

AUTHORITY

USAETL ltr, 1 Oct 1970

THIS PAGE IS UNCLASSIFIED

AD 860060

SPECIAL INTERIM TECHNICAL REPORT
THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

by
Robert Haralick

CRES Technical Report No. 133-9
July 1969

Sponsored by
ADVANCED RESEARCH PROJECTS AGENCY, DEPARTMENT OF DEFENSE
Work Order No. 1079

STATEMENT #2 UNCLASSIFIED

This document is subject to special export controls and each transmittal to foreign government or foreign nationals may be made only with prior approval of .....

Monitored by
U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
GEOGRAPHIC INFORMATION SYSTEMS BRANCH
GEOGRAPHIC SYSTEM DIVISION
FT. BELVOIR, VIRGINIA
attn: ETL.GS.

Contract No. DAAK02-68-C-0089

D D C
RECEIVED
OCT 10 1969

CRES
THE UNIVERSITY OF KANSAS • CENTER FOR RESEARCH INC
ENGINEERING SCIENCE DIVISION • LAWRENCE, KANSAS

69

AD _____

SPECIAL INTERIM TECHNICAL REPORT

THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

Robert Haralick

CRES Technical Report No. 133-9

July 1969

Sponsored by
ADVANCED RESEARCH PROJECTS AGENCY, DEPARTMENT OF DEFENSE
Work Order 1079

Monitored by
U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
GEOGRAPHIC INFORMATION SYSTEMS BRANCH
GEOGRAPHIC SYSTEM DIVISION
FT. BELVOIR, VIRGINIA

Contract No. DAAK02-68-C-0089

ABSTRACT

The first part of this paper provides a brief tutorial introduction of the Bayesian Approach to identification of a remotely sensed environment. The second part describes the input data deck setup for the Fortran IV program which has been written to implement this approach. The third part describes file usage and subroutine organization. The fourth part provides a listing of the program with a simple sample data set.

ACKNOWLEDGEMENT

I wish to acknowledge the assistance of Carl Smith who did most of the programming, and the University of Kansas Computation Center for providing some of the necessary computer time.

TABLE OF CONTENTS

		Page
ABSTRACT		i
ACKNOWLEDGEMENT		ii
LIST OF ILLUSTRATIONS		iv
I	THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT	1
II	INPUT DATA DECK	7
III	FILE USAGE AND SUBROUTINE ORGANIZATION	15
IV	EXAMPLE PROBLEM AND PROGRAM LISTING	49

LIST OF ILLUSTRATIONS

		Page
Figure 1a	Flow Chart for Setting up Input Data Deck	13
Figure 1b	Flow Chart for Setting up Input Data Deck	14
Figure 2	Tape Usage	16
Figure 3	Storage Requirements for the Tape or Disc Files .	17
Figure 4a	Data for Test 1	21
Figure 4b	Data for Test 2	21
Figure 5	Gain Matrix	22
Figure 6	Input Data Deck for First Example Problem	25
Figure 7	Input Data Deck for Second Example Problem . . .	26

PART I

THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

I. THE BAYESIAN APPROACH TO IDENTIFICATION OF A REMOTELY SENSED ENVIRONMENT

Using remote sensors we can make measurements of an environment. The set of measurements made will be called the data set. Our job is to examine the data set in order to identify what the environment is made up of: our problem is how should we do it? In what follows we describe the Bayesian decision approach with a deterministic decision rule.

We assume that distinct boundaries enclose a limited environment, which is made up of small-area patches, one next to the other. The identification of the environment consists of identifying each small-area patch within one category of a given set of categories. We assume that such an identification is sensible and possible.

In order to make any identification we must have knowledge concerning which kind of measurements are typical measurements of the categories we wish to identify. This knowledge is succinctly contained in a classification, which is a mapping, associating with each measurement the category to which it is most typical - given a specific decision criterion. Therefore, if we are to identify measurements in a data set we must have a classification.

How do we obtain a classification? We perform an information gathering experiment. From the population of all environments, we sample one or a few in which it is possible to identify many small-area patches within each category of interest. The proportion of occurrence of each category in the sampled environment(s) does not have to be representative of the average probability of occurrence of each category in the entire environmental population. However, if we have no information regarding the average probability of occurrence of each category in the environmental population, then we would want to choose the sampled environment(s) so that the proportion of occurrence of each category in the sampled environment(s) is an unbiased estimate of the

average probability of occurrence of each category in the environmental population. In either case, the small-area patches within each of the sampled environment(s) do have to be representative of the categories with which they are identified.

With each of our sensors, we measure each small-area patch in the chosen environments. From photo-interpretation or field studies, the environments are examined first hand, and an identification of each small-area patch is made. The sequence of such identifications is called the "ground truth identification" or simply "ground truth". It is from the data set (the sequence of measurements) and the ground truth (the sequence of identifications) that we can find a Bayes classification.

At this point we must introduce some mathematical notation. Let $C = \{c_i\}_{i=1}^{K}$ be the set of K given categories; c_i is the symbol used for the ith category. We suppose, for convenience, that each sensor produces only one number for each measurement it makes of a small-area patch We suppose further, that the jth sensor must produce a number belonging to its range set $L_j = \{l_{j1}, l_{j2}, \ldots, l_{jN_j}\}$. This supposition is fully in accord with reality, since the output of any sensor is always equivalent to a pointer-reading on a dial. Pointer-readings can never be discerned precisely, and are thus discerned approximately to third, or fourth, or, . . . , N^{th} place accuracy.

Measurement space M is the set of all measurements which are possible to make with the set of S sensors. M is conveniently described as the cartesian product of the range sets; $M = L_1 \times L_2 \times \ldots \times L_S$. This is the set of measurements which contain for elements, all the possible numbers produced by sensor one, combined with all the possible numbers produced by sensor two, . . . , combined with all the possible numbers produced by sensor S. For convenience we number the measurements in M; $M = \{m_n\}_{n=1}^{N}$, where N is the total number of elements in measurement space. Finally we must provide a goodness criterion; thus, we introduce a gain function g. $g(c_i, c_j)$ is our economic gain if we identify a measure-

ment as belonging within the ith category when that measurement was made of a small-area patch actually belonging within the jth category.

We have already mentioned that a classification is a mapping or rule which associates with each measurement m_n in M, the category c_i to which it is most typical - according to some decision criterion. Our decision criterion is economic; "most typical to" translates to, "that association by which we, on the average, gain the most economically". Therefore, according to our decision criterion, we can judge each possible classification. That classification which enables us to gain the most, on the average, is the classification which is best; it is that classification which we wish to find.

Let us now examine how the average gain may be calculated. Let f be a classification mapping. f is a function whose domain is the set M, and whose range is the set C; f: M→C. For each element $m_n \in M$ the function associates one and only one category $c_i \in C$. We define the characteristic function h_f for f as follows: for every $m_n \in M$, $c_i \in C$.

$$\left.\begin{array}{l} h_f(c_i, m_n) = 1 \text{ if and only if } f(m_n) = c_i \\ \qquad\qquad\qquad\quad 0 \text{ otherwise} \end{array}\right\} .$$

In other words $h_f(c_i, m_n)$ is 1 if and only if the classification f identifies the measurement m_n as belonging within the category c_i. The average gain A for the classification f is easily seen to be:

$$A(f) = \sum_{i=1}^{K} \sum_{k=1}^{K} \sum_{n=1}^{N} g(c_k, c_i) h_f(c_k, m_n) P(m_n | c_i) P(c_i)$$

where $P(m_n | c_i)$ is the conditional probability that the measurement m_n will be made of a small-area patch given that the patch belongs within category c_i. $P(c_i)$ is the probability that any small-area patch of the

environments in the population belongs within category c_i, and $g(c_k, c_i)$ is the amount gained if a patch which actually belongs within category c_i is identified within category c_k.

Of the four terms in the summation, $g(c_k, c_i)$ is specified as part of the identification goodness criteria, $h_f(c_k, m_n)$ is defined from the classification f, $P(m_n|c_i)$ will be determined from the data gathered in the experiment, and $P(c_i)$ is an additional a priori probability which we will have to specify. Let us now examine in detail how the conditional probabilities are determined from the experimental data.

The data set is a sequence D of R measurements;

$$D = \langle m_{r_1}, m_{r_2}, \ldots, m_{r_R} \rangle .$$

The ground truth corresponding to sequence D is a sequence T of R not necessarily different category identifications; $T = \langle c_{r_1}, c_{r_2}, \ldots, c_{r_R} \rangle$. Let # be the counting measure. #(D) is the number of elements in the sequence D; thus, #(D) = R. A sequence is really a function whose domain is the set of integers I. The data set D is then a function which associates with each integer, a measurement; $D: I \rightarrow M$. The ground truth T is also a function and it associates with each integer a category; $T: I \rightarrow C$. D(7), for example, is then just the seventh element in the sequence D; $D(7) = m_{r_7}$. $D^{-1}(m)$ is the set of all integers i for which $D(i) = m$. The statistic $\hat{P}(m_n|c_i)$ estimating $P(m_n|c_i)$ is defined as

$$\left.\begin{aligned} \hat{P}(m_n|c_i) &= \frac{\#(D^{-1}(m_n) \cap T^{-1}(c_i))}{\#(T^{-1}(c_i))} \quad \text{when } \#(T^{-1}(c_i)) \neq 0 \\ &= 0 \text{ otherwise} \end{aligned}\right\} .$$

$\hat{P}(m_n|c_i)$ is the number of integers which are associated with the measurement m_n in the sequence D and with the category c_i in the sequence T, divided by the number of integers associated with the category c_i in sequence T. Stated simply, $P(m_n|c_i)$ is just the number of times the measurement m_n was made of a small-area patch belonging within category c_i, divided by the number of times a small-area patch belonged within the category c_i.

The a priori probabilities $P(c_i)$ can either be estimated from the sampled data set (if this data set is representative of the population) or from our foreknowledge of the population of environments. If we can assume that the few environments we have chosen to sample for our experiment are representative of the population, then

$$\hat{P}(c_i) = \frac{\#(T^{-1}(c_i))}{R}$$

is a reasonable estimate. If we cannot make such an assumption and we believe that a small-area patch is just as likely to belong within one category as within another, then $\hat{P}(c_i) = 1/K$ is a reasonable estimate.

From the estimates $\hat{P}(m_n|c_i)$ and $\hat{P}(c_i)$ we may estimate the average gain $\hat{A}$ for any classification f. As before let h_f be the characteristic function for f.

$$\left.\begin{aligned} h_f(c_k, m_n) &= 1 \text{ if and only if } f(m_n) = c_i \\ &= 0 \text{ otherwise} \end{aligned}\right\} .$$

$$\hat{A}(f) = \sum_{i=1}^{K} \sum_{k=1}^{K} \sum_{n=1}^{N} g(c_k, c_i) h_f(c_k, m_n) \hat{P}(m_n|c_i) \hat{P}(c_i).$$

We seek the Bayes classification f* which maximizes $\hat{A}$. f* is easily defined. For each measurement m_n and for any classification f,

there will be one and only one category c_j such that $h_f(c_j, m_n) = 1$. Consider the amount $\hat{a}(c_j, m_n)$ gained due to the identification of measurement m_n as belonging within category c_j.

$$\hat{a}(c_j, m_n) = \sum_{i=1}^{K} g(c_j, c_i)\hat{P}(m_n | c_i)\hat{P}(c_i)$$

The maximum $\hat{A}(f)$ is certainly achieved if for each measurement m_n, $f(m_n) = c_j$ where c_j maximizes $\hat{a}(c_j, m_n)$. Therefore we just have to compute $\hat{a}(c_j, m_n)$ for $j = 1, 2, \ldots, K$ to determine which category, c_j, maximizes it. Then we define $f^*(m_n) = c_j$.

In this manner we can define how to best identify each measurement which actually occurred in the data sequence D. However, there may be many measurements in measurement space M which did not occur in the data sequence. How should these measurements be identified in the classification? Since we have no data or statistics for these measurements it seems that we have no way to deal with them! Here we must draw upon our knowledge of the structure of reality. We know that in any environment if a measurement m is made of a small-area patch belonging within category c_i, then it is likely to make measurements $m + \delta$ for other small-area patches which also belong within category c_i. If a measurement m is typical of category c_i, then for small δ, $m + \delta$ is also typical of category c_i. Similar or close measurements are usually associated with similar or the same categories. Thus in the classification we can identify a measurement m, which did not occur in the data sequence, with the category associated with m', its nearest neighbor.

The part of the classification f^* which was defined by means of the statistics generated by the experiment is called a Bayes Classification and hence the name "Bayesian approach." The part of the classification which is not Bayesian is said to be defined by a nearest neighbor search.

Acknowledgement: This work was supported by Project Themis (USAETL Contract DAAK02-68-C-0089, ARPA order No. 1079).

PART II

INPUT DATA DECK

II. INPUT DATA DECK

The data for this program are received as a sequence of measurements of small-area patches or objects with each measurement made by a sensor or set of sensors. A measurement may be, for example, the average backscatter power return from a small-area patch at incidence angles 5°, 10°, 15°, 20°, 30°, 40°, 50°, and 60°. In this case, each measurement has 8 components. The patches themselves are examined and identified as belonging within one of several given categories. The sequence of such identifications is called the "ground truth identification" or simply the "ground truth." The Bayes program can determine a Bayes classification of measurement space, based on the data and the ground truth for the data. Once a classification is determined, the Bayes program can identify each measurement within a sequence of measurements. This identification is done by a nearest-neighbor search.

The input deck is organized into four sections: title, program options, parameter cards, and format and data.

I. Title

A) The title section consists of a single card specifying the name of the data. The title may begin in column one and continue through column eighty.

II. Program Options

A) The program option section consists of two cards, the first card specifying all the input options and the second card specifying all the output options.

B) Each option is a six-character abbreviation or code.

C) The options start in column 16, are separated only by commas (no embedding blanks) and may appear in any order.

D) Only input options may appear on the input card and only output options may appear on the output card.

1) The input options are: PHOPTS, CORPTS, FLTING, PATERN, DIAGON, HLFNHF, ABSQNT.

2) The output options are: ALTICH, STDPNT, DPUNCH, PHOUT1, PHOUT2, TERMNL.

III. Parameters

A) The parameter section consists of cards, the number of which varies with the options chosen.

B) There are seven basic types of information which can possibly appear in the parameter section: gain matrix, dimensionality of measurements, number of measurements in the data set, number of categories in the classification, display size, number of levels to which the measurements will be quantized, and means of estimating the a priori probability distribution.

IV. Format and Data

A) There are two ways to organize the format and data: photographic form and corresponding-point form. Depending on the options chosen, the ground truth identification and its format may or may not be present. One and only one of the two forms must be specified: otherwise, an error message and termination of the job will result.

1) In the photographic form the data are organized as follows:

a) format for identification (if any)

b) identification (if any) for measurement one, measurement two, . . . , measurement N.

c) format for measurements

d) component one, measurement one; component one, measurement two; . . . ; component one, measurement N; component two, measurement one; component two, measurement two; . . . ; component two, measurement N; . . . component M, measurement one; component M, measurement two; . . . ; component M, measurement N.

2) In the corresponding-point form the data are organized as follows:

a) format for identification (if any) and measurements

b) identification (if any) for measurement one; component one, measurement one; component two, measurement one;...; component M, measurement one: identification (if any) for measurement two; component one, measurement two; component two, measurement two;... component M, measurement two:... identification (if any) for measurement N; component one, measurement N; component two, measurement N;... component M, measurement N.

We now describe the options.

I. Input Options

A) PHOPTS -- is the abbreviation for photographic form, and is used when the data are in that form.

B) CORPTS -- is the abbreviation for corresponding point form, and is used when the data are in that form.

C) FLTING -- is the abbreviation for floating point, and is used when the data are punched on cards in floating-point form. It is not used when the data are punched on cards in integer form.

D) PATERN -- is the abbreviation for pattern classification by Bayes' strategy. Use of this option will output a probability matrix where element (i, j) is the conditional probability that a measurement which was identified as within the ith ground truth category is identified in the classification as within the jth category. A punched deck of the compacted quantized measurements with their identifications in the Bayes' classification will also be produced.

E) DIAGON -- is the abbreviation for diagonal gain matrix with ones on the diagonal. Specification of DIAGON will internally generate an identity matrix for the gain matrix. This relieves the user of the need to supply the appropriate cards in the parameter section.

F) HLFNHF -- is the abbreviation for half and half. Specification of HLFNHF will divide the data into halves: even and odd points. The first, third,. . ., data points are used to construct a Bayes' classification, and the second, fourth, . . .,data points are identified on the basis of the classification.

G) ABSQNT -- is the abbreviation for absolute maximum quantization. Specification of ABSQNT will quantize the measurements by determining the minimum and maximum values occurring among all the components, normalizing each component by subtracting this minimum, dividing by this maximum minus this minimum, and multiplying by the number of quantized levels desired. If ABSQNT is not specified, the program finds the maximum and minimum for each component, and normalizes component by component.

II. Output Options

A) ALTPCH -- is the abbreviation for alternate punch. Specification of ALTPCH produces a punched quantized data deck in the alternate form. If the data are in photographic form, the punched deck will be in corresponding-point form. If the data are in corresponding-point, the punched deck will be in photographic-point form.

B) STDPNT -- is the abbreviation for standard identification print out. Specification of STDPNT will print out the ground truth identification.

C) DPUNCH -- is the abbreviation for punched deck. Specification of DPUNCH will produce a punched quantized data deck in the same form as the input data.

D) PHOUT1 -- is the abbreviation for photograph output. Specification of PHOUT1 will identify data according to a given classification. The classification can be internally generated by the Bayes routine or it can be externally supplied in the data deck. Identification is done by a

table look-up procedure. If the quantized measurement cannot be found in the classification, a nearest-neighbor search is initiated, and the measurement is identified with the same identification as the closest measurement to it in the classification.

If the input includes ground truth identification for a data set and the data are further identified relative to a classification (by specification of PHOUT1), then a contingency table of the ground truth identification versus the classification identification will be printed out. The $(i,j)^{th}$ element of the table is the number of measurements which were identified in the ith ground truth category and classified in the jth ground truth category.

E) PHOUTD -- is the abbreviation for photograph output deck. Specification of PHOUTD will produce a punched deck of the identification which the photograph output routine determined. PHOUTD can only be specified if PHOUT1 is specified.

F) TERMNL is the abbreviation for remote terminal format. Specification of TERMNL will format all printed output from the Bayes program so that each line has no more than seventy columns. If TERMNL is not specified, each line is printed with one hundred thirty columns.

The generality of the program requires that the input parameters be quite variable, depending on the type of data to be processed and the options desired. For example, the PATERN option may or may not require the gain matrix as input depending on whether or not DIAGON was also specified. To facilitate setting up the input data deck, a flow chart is illustrated in Figure 1a and 1b.

The flow chart contains two symbols. The first symbol is the elongated circle (rectangle with rounded edges), which asks the question printed in the circle and requires a "true" or "false" answer. Depending upon the user's answer, one branch is chosen from the bottom of the circle.

The second symbol is the rectangle, which usually signals the addition of a single data card. Data cards are added when rectangles are encountered in the flow chart. Two exceptions to the "one-rectangle, one-data-card" rule are the gain matrix and the data itself. If the gain matrix is required (PATERN is specified and DIAGON is not specified), the user must supply all the elements for the matrix. This will take up more than one card if there are more than two ground truth identification groups. Similarly, if there are more than eight measurements in the data set, the data deck will have more than one card.

The final control card is the STOP card and is the last card in the input deck. The four-character word STOP is punched in columns one through four. The program is designed to handle more than one data set per run, and the user may place behind the first (or second, or third, etc.) data set the NAME of the next data set, the proper /INPUT and /OUTPUT cards, and the other necessary parameter cards as specified by the flow chart. The STOP card is placed after the final set of data, and informs the system to terminate the job. Excluding the STOP card or misplacing any control cards will cause a read error and termination of the job.

FLOW CHART FOR SETTING UP INPUT DATA DECK

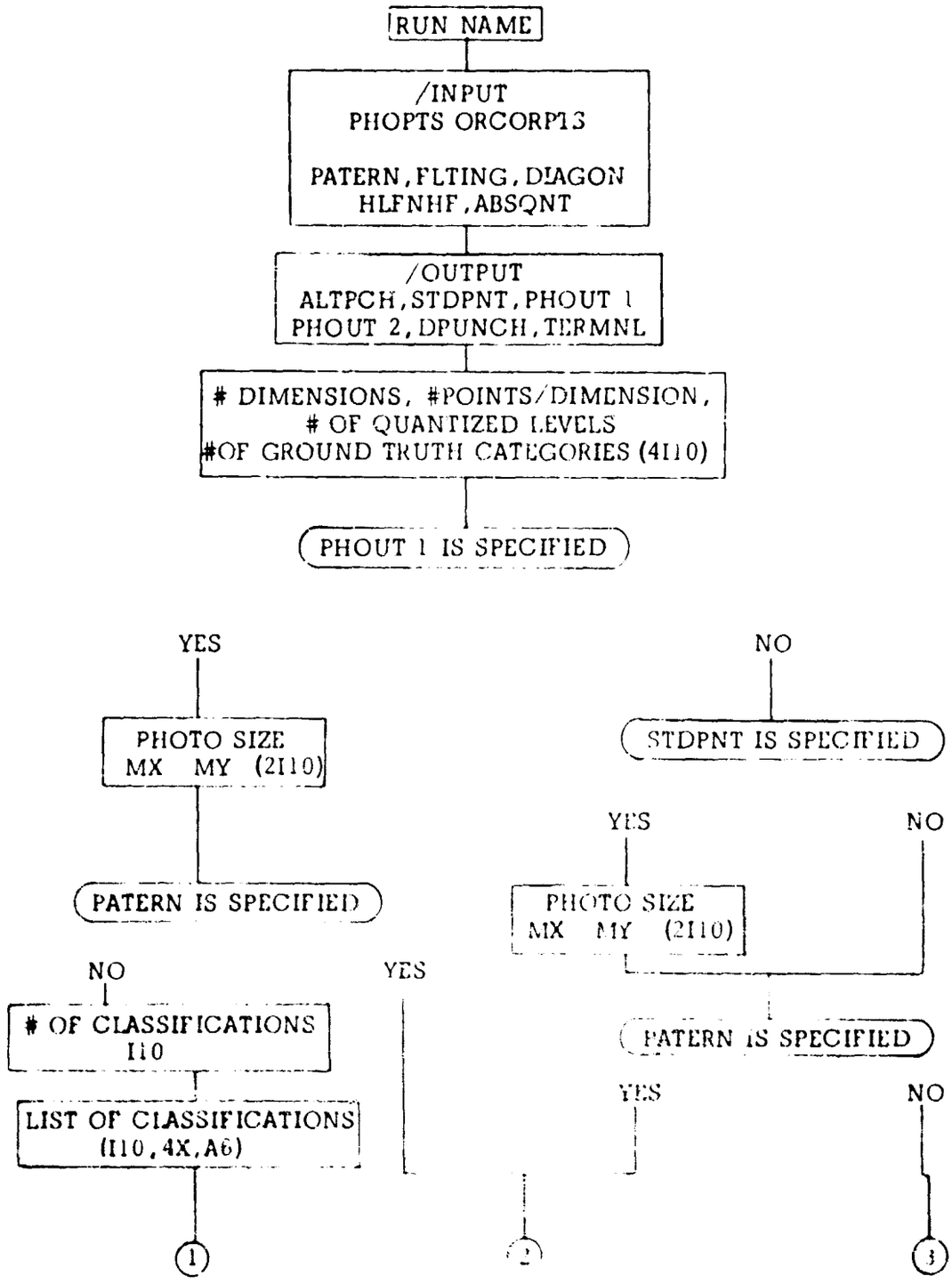


Figure 1a.

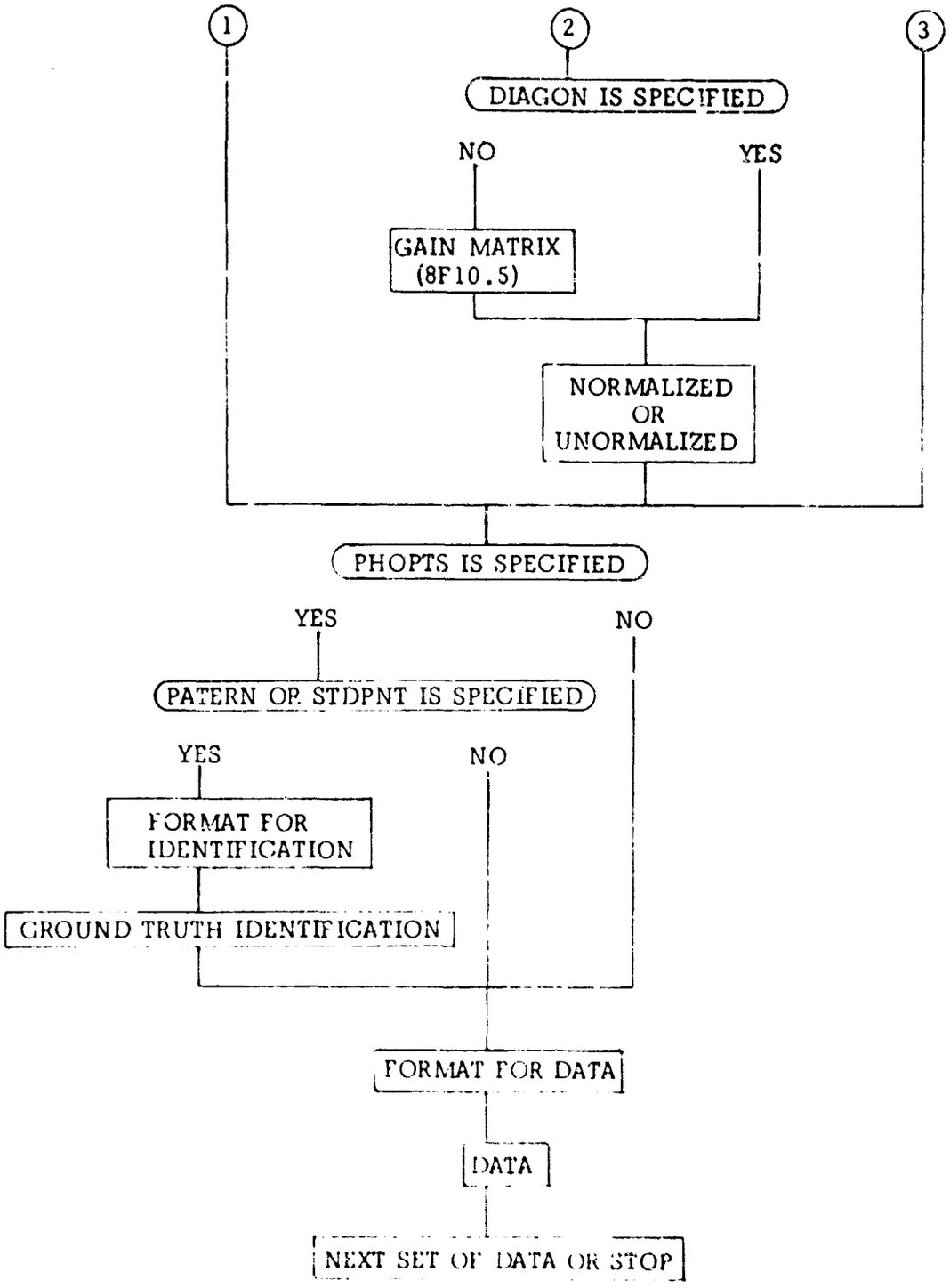

1
2
3
DIAGON IS SPECIFIED
NO
YES
GAIN MATRIX
(8F10.5)
NORMALIZED
OR
UNORMALIZED
PHOPTS IS SPECIFIED
YES
NO
PATERN OR STDPNT IS SPECIFIED
YES
NO
FORMAT FOR
IDENTIFICATION
GROUND TRUTH IDENTIFICATION
FORMAT FOR DATA
DATA
NEXT SET OF DATA OR STOP

Figure 1b.

PART III

FILE USAGE AND SUBROUTINE ORGANIZATION

III. FILE USAGE AND SUBROUTINE ORGANIZATION

Blank common storage carries all problem parameters and user options (CORPTS, PHOPTS, etc.), as well as providing a 24,000-word scratch area. Many routines in different links require such parameters as the number of dimensions of the current problem, the number of points being processed, and the number of ground truth categories.

All other communications between links are handled by tape, disc, or drum files. The program requires nine files -- 01, 02, 03, 04, 09, 10, 11, 20, 21 — as well as the input (05), output (06), and punch (43) files normally used in FORTRAN. Figure 2 describes file usage, and Figure 3 illustrates how much storage is needed on each of the files.

The Bayes program, as mentioned before, requires 36,000 words of storage in the computer core, of which 25,000 may be shared during loading. The program has been observed to process 350 sixteen-dimensional data points in less than ten minutes' processor time.

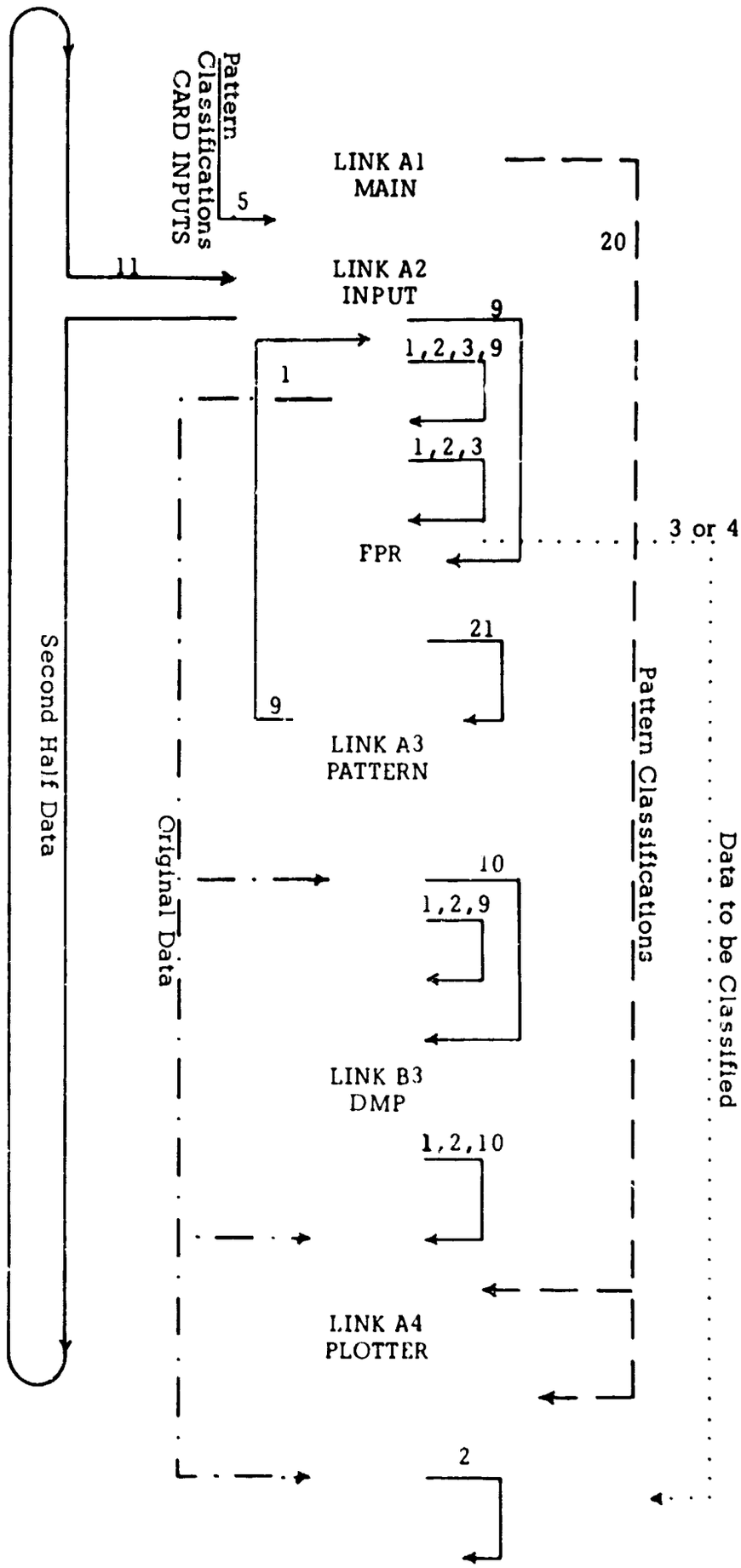


Figure 2. Tape Usage

FILE	NUMBER OF WORDS NEEDED
01	Total points
02	Total points
03	(Total Points) * # dimensions
04	
09	(Total Points) * # dimensions
10	2*(total points)
11	1/2*(total point)
20	2*(# unique n-tuples in data)
21	(Total Points) * # dimensions

Figure 3. Storage Requirements for the Tape or Disc Files

The program organization is briefly described below. Listed first are the mainline and CHNXT, the system supervisor, followed by the six-character link names used during overlay processing. Under each link name are listed the routines contained in the link. Link names containing the alphabetic character "A" refer to links essential for proper data processing: link names containing the alphabetic character "B" refer to links which output intermediate calculations, but which do not contribute to the overall program results.

LINKB2

.....(mainline) -- reserves all common storage, provides entry point to program, and contains comment cards stating program parameters and input deck setup.

CHNXT -- a small resident program used to control the entry and exits of the different links. This routine must comply with the overlay rules of the operating system in use.

LINKA1

MAIN -- parameter input and selection is accomplished in this routine. Also, if the photo classifications and the gain matrix must be read in, it is done in MAIN.

FORM -- selects and places in common all formats affected by terminal or non-terminal use.

LINKA2

INPUT -- performs data input, saves every even point for later processing if desired, and changes data to alternate form if called for.

FPR-FPR1 -- searches data for the maximum and minimum points so that proper quantization and shifting may be done in INPUT.

DEF -- defines the single-character symbols to be used in the classification.

TRANE -- prints a cross-reference between the single-character symbols used in the program and the original symbols.

CHANGE -- an assembly language routine which creates a suitable output format for the data.

LINKA3

PATERN -- Bayes program

OUTP -- prints out results of PATERN in eye-appealing format.

DECSON -- selects and assigns to each n-tuple the proper classification according to Bayes theory.

LINKB3

DMP -- used for debugging; prints out n-tuples vs. categories and n-tuples vs. classification.

LINKA4

PLOTER -- classifies and plots input data according to n-tuple classifications.

SEARCH -- searches a list of n-tuples to find the list element which is closest in distance to another n-tuple.

IDIST -- calculates n-dimensional space distances.

PART IV

EXAMPLE PROBLEM AND PROGRAM LISTING

IV. EXAMPLE PROBLEM

Suppose the problem is to input a set of data in photographic form, punch out the data in the alternate form, print the data identification, and then classify a new set of data. Let there be nine points in the horizontal direction and ten points in the vertical direction for the first data set. The data appear in Figure 4a. For the first part of the problem, the program must produce an alternate data deck, a print-out of the identification, and a Bayes' classification based on the first set of input data.

The first card is the title card. The next two cards specify the input and output options needed. The options start in column 16 and are separated by commas. The data are in photographic form, so the input card is:

```
/INPUT          PHOPTS,PATERN
```

The output card is:

```
/OUTPUT         ALTPCH,STDPNT
```

The parameter card follows. From the flow chart we see that the fourth card must specify the number of dimensions per measurement (number of photographs), total number of measurements, number of quantized levels, and number of ground truth categories. In our example the number of photographs is two and the number of measurements is ninety. We wish to quantize the data to ten levels and there are two ground truth categories. The fourth card thus appears as:

```
2          90          10          2
```

Since PHOUT1 is not specified and STDPNT is, the fourth card must specify the photograph size, which is nine points horizontal by ten vertical. The fifth card appears as:

```
9          10
```

```
111151111
111151111
111151111
111151111
555555555
555555555
111151111
111151111
111151111
111151111
111151111
```

Photo 1 for Test 1

```
222262222
222262222
222262222
222262222
333333333
333333333
222262222
222262222
222262222
222262222
```

Photo 2 for Test 1

Figure 4a. Data for Test 1

```
51115
15151
11511
15151
51115
```

Photo 1 for Test 2

```
32223
26262
22322
23232
62226
```

Photo 2 for Test 2

Figure 4b. Data for Test 2

Since PATERN is specified and DIAGON is not, the user must supply the gain matrix. Suppose we choose a gain matrix where we gain ten for a correct decision and lose five for an incorrect decision, as illustrated in Figure 5.

$$\begin{matrix} 10 & -5 \\ -5 & 10 \end{matrix}$$

Figure 5. Gain Matrix

The sixth card, specifying the above matrix, is

10. -5. -5. 10.

We must now indicate how the a priori probability distribution is estimated. We choose to suppose that each identification group has equal probability. Therefore NORMALIZED is specified on the next card,

NORMALIZED

The data are in photographic form, so the format for the ground truth identification must come next. In our example this would be:

(9A1).

After the identification format, the identification itself comes:

AAAABAAAA
AAAABAAAA
AAAABAAAA
AAAABAAAA
BBBBBBBBB
BBBBBBBBB
AAAABAAAA
AAAABAAAA
AAAABAAAA
AAAABAAAA

Finally we reach the format for the data, the data itself, and the STOP card. In our example these appear as:

```
(9I1)
111151111
111151111
111151111
111151111
555555555
555555555
111151111
111151111
111151111
111151111
111151111
222262222
222262222
222262222
222262222
333333333
333333333
222262222
222262222
222262222
222262222
STOP
```

The input data deck is illustrated in Figure 5.

At this point we must run a job with the input deck as shown in Figure 6. The job produces a punched deck of the Bayes classification. We must now identify a new set of data which is illustrated in Figure 4b. This is done by a separate job. The first card is, as usual, the title card. The second and third cards are the input and output option cards. The new set of data is in photographic form, and we wish to have a print-out of the identification for it, based on the classification of the previous job. The next cards thus appear:

```
/INPUT        PHOPTS
/OUTPUT       PHOUT1
```

There are two dimensions, twenty-five measurements, ten quantized levels, and two identification groups. The fourth card is:

```
2          25          10          2
```

Since PHOUT1 is specified, the next card must indicate the photographic size which is, in our example, five points horizontal by five vertical. The next card is thus:

```
5          5
```

The number of quantized measurements in the classification and the Bayes classification itself come next. These cards were obtained from the output of the previous job.

For our example, they are:

```
3
                65.02       35.02       21.01
```

All that now remains is the format for the data, the data itself, and the STOP card.

```
(5I1)
51115
15151
11511
15151
51115
32223
26262
22322
23232
62226
STOP
```

The input data deck is illustrated in Figure 7.

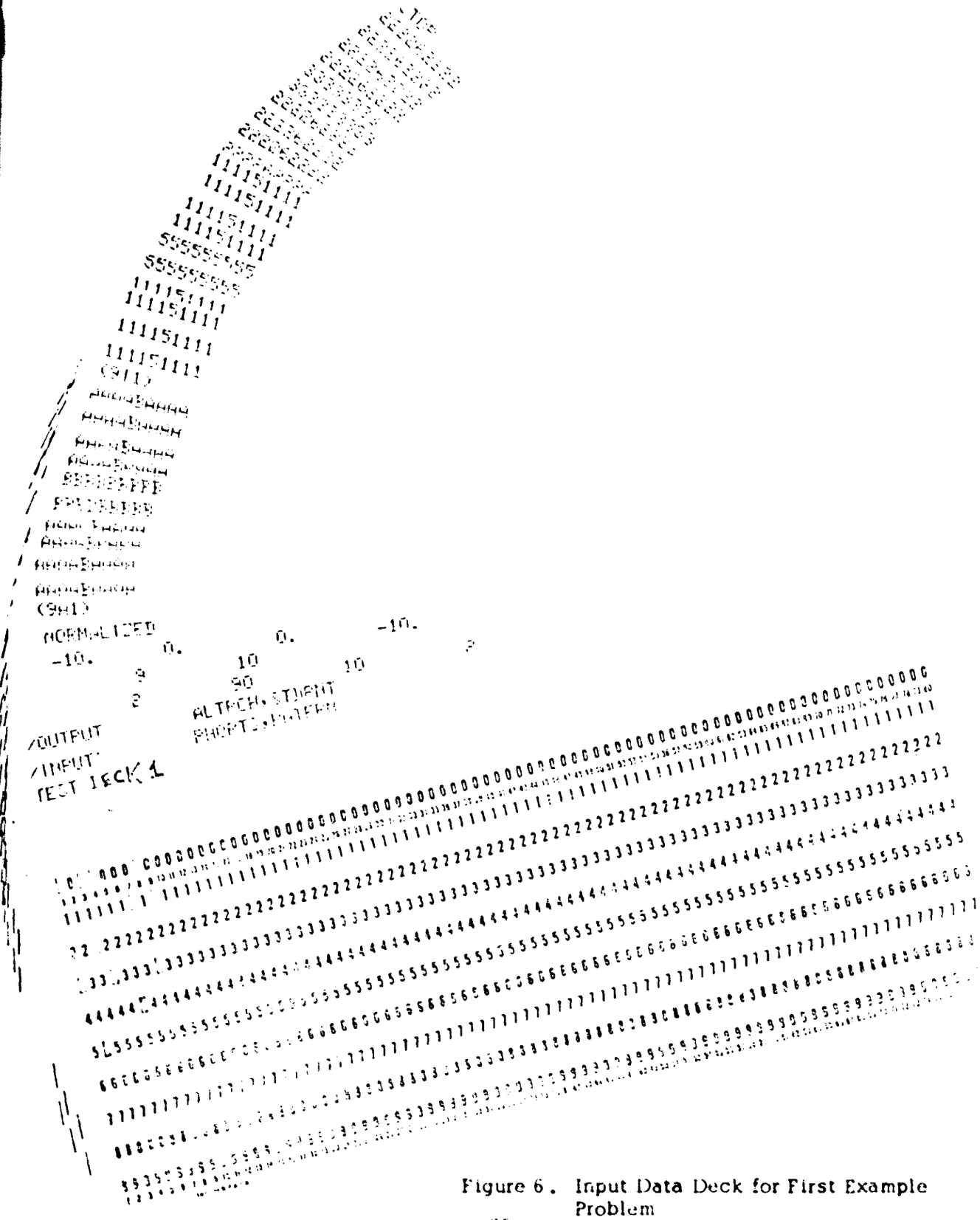


Figure 6. Input Data Deck for First Example Problem

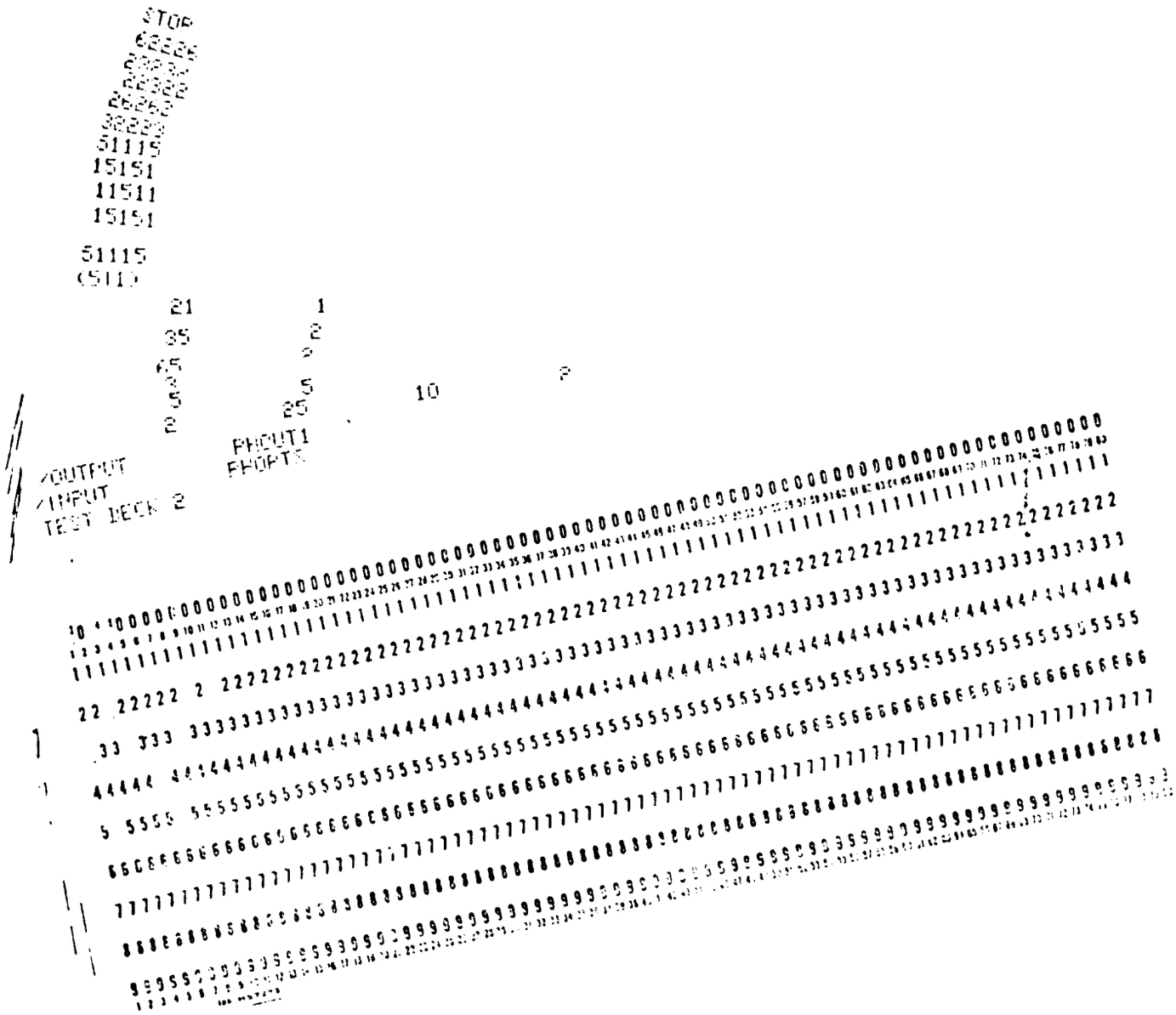


Figure 7. Input Data Deck for Second Example Problem

TEST DATA SPECIFICATIONS

```
C                                   ...
C        SYSTEM DATA SPECIFICATIONS
C
C     ******************************************************************
C
C      EACH DATA SET SHOULD BE PREFIXED WITH A SINGLE  CARD
C        CONTAINING THE RUN NAME
C
C        1 ST CARD  (INSTRUCTION PARAMETERS) FORMAT(5(A6,1X))
C          THIS CARD CONTAINS UP TO FIVE SPECIFICATIONS SEPERATED BY A
C          COMMA ENABLING THE USER TO ENTER AND EXIT THE PROGRAM AT
C          ANY POINT DESIRED.
C
C          THE PROGRAM EXPECTS THE INPUT DATA IN ONE OF TWO FORMS, THE
C          USER IS RESPONSIBLE FOR SPECIFYING THE FORM.  IF NEITHER OR
C          BOTH ARE SPECIFIED AN ERROR MESSAGE RESULTS.
C
C          LITERAL              DATA FORM
C
C          PHOTO                TRAINING REGIONS,PHOTO 1 ' PHOTO 2 ' ' ETC.
C
C          COORD                POINT1-PHOTO 1 , POINT1-PHOTO 2, ----------
C                               POINT2-PHOTO 1, POINT 2-PHOTO 2, ----------
C                                  .
C                                  .
C                               POINT N-PHOTO 1, POINT N-PHOTO 2, ---------
C
C          THE OTHER PARAMETERS THAT MAY BE SPECIFIED ARE
C
C          LITERAL
C
C          ALTPCH               A DATA DECK WILL BE MADE IN THE ALTERNATE D
C                               'DATA FORM'
C
C          STDPUT               A PRINT OUT OF THE TRAINING REGIONS WILL BE
C                               MADE
C
C          PATERN               SUBROUTILE PATERN WILL BE CALLED
C                               (SZATTEROMETRY PROGRAM) A TAPE SUCH AS IS
C                               MADE IN PATTAP MUST BE AVAILABLE ON FORTRAN
C                               UNIT 1
C
C          PHOUT1               PREPARES A PHOTO PRINT USING PATERN
C                               CLASSIFICATION ( IF PATERN IS NOT SPECIFIED
C                               PATERN CLASSIFICATIONS MUST BE READ IN-
C                               SEE 3RD CARD )
C
C          PUNOUT               PREPARES A PHOTO PUNCHED OUTPUT USING
C                               PATERN CLASSIFICATIONS
C
C          SPUNCH               PUNCH OUTPUT OF QUANTIZED DATA
C
C          FLTING               SPECIFIED IF DATA IS IN FLOATING POINT
C                               FORMAT
C
C          TERMNL                SPECIFIED IF REMOTE TERMINAL IS BEING USED
C
C          DIAGON               SPECIFIED IF THE GAIN MATRIX USED IN PATERN
C                               IS TO HAVE TERMS ON ALL DIAGONAL ELEMENTS
C                               AND ZEROES ELSEWHERE
C
C     ******************************************************************
C
C        2 ND  CARD
C
C          NO. PHOTOS,NO. POINTS PER PHOTO,
C                               DESIRED LEVEL OF QUANTIZATION, NO. OF
C          TRAINING REGIONS (NR), FORMAT(5I10)
C          NR MUST BE LESS THAN 26
C     ******************************************************************
C
C
C        3 RD   CARD / CARDS
C
C          IF PHOUT1 IS SPECIFIED, READ IN PHOTO SIZE
C                  (NA,NB) FORMAT(2I10) )
C          AND PATERN CLASSIFICATIONS IF SUBROUTINE PATERN IS NOT USED.
C          NCLAS = NO. OF POINTS CLASSIFIED FORMAT(I10), ATTACHED WITH
C          CLASSIFICATIONS FORMAT(8F10 2)
C
C        IF STDPUT IS SPECIFIED, READ IN PHOTO SIZE (NA,NB) FORMAT(2I10)
C          IF PATERN IS SPECIFIED,
C            READ IN THE LOSS MATRIX TO BE USED (NRNR) FORMAT(8F10,5)
C            READ IN 'NORMALIZED' OR 'UNNORMALIZED' FORMAT(A6,A6) LDPT-
C            AUGTIFIED
C
C     ******************************************************************
C
C
C        OTHERS
C
C          READ FORMAT FOR DATA
C        READ DATA
C
C          PUNCH STOP IN COL 1-4 OF THE LAST CARD - THIS ENABLES MORE
C          THAN ONE SET OF DATA TO BE PROCESSED ON A SINGLE RUN
C
C     ******************************************************************
C
C                                   ...
      COMMON NPOL,FORMAT(10)
      COMMON F
      COMMON FAT1(10),FAT2(3),FAT3(3),FAT4(3),FAT5(10),FAT6(2),FAT7(10),FAT8(3),FAT9(3)
     1,FAT10(3),FAT11(3),FAT12(3),FAT13(3),FAT14(3),FAT15(3),FAT16(3)
      COMMON L0,L1,L2,L3,L4,L5,L6,L7
      COMMON N,M,LEVEL,NP1,NPP,NR(26),ITAPE,ANATA(80),L7,L1,L2,L3,L4,L5,
     1L6,NA,NB,NP,N,L7,NA,NB,NE,ZZZZZZZZ
      LOGICAL L7,L1,L2,L3,L4,L5,L6,L7
      LOGICAL L0,L1,L2,L3,L4,L5,L6,L7
      INTEGER F
C       SET LAST WORD IN COMMON TO 9 AND THE WORD IN COMMON TO 0
      F=3
      END
      CALL EXMAT
      STOP
      END
```

```
*************************************************************************
*    DEGREE OF MEMORY USED BY THIS COMPILATION
*************************************************************************
```

```
      SUBROUTINE CHNXT                                                  A   1
      DIMENSION NAME(3)                                                 A   2
      COMMON MPO1,FORMT1(12)                                            A   3
      COMMON F                                                          A   4
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2 A   5
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)                    A   6
      COMMON L8,L9,L10,L11,L12,L13,L14                                  A   7
      COMMON N,M,LEVEL,KZ1,KCOUNT(36),ITAPE,AMAT(625),LT,L1,L2,L3,L4,L5, A   8
     1L6,KK,MX,MY,K,L7,MA,MB,MC,ZZ(24000)                               A   9
      LOGICAL LT,L1,L2,L3,L4,L5,L6,L7                                   A  10
      LOGICAL L8,L9,L10,L11,L12,L13                                     A  11
      INTEGER F                                                         A  12
      DATA NN/6HSTOP  /                                                 A  13
C                                 ***                                   A  14
C      K POINTS TO LINK IN USE                                          A  15
C                                 ***                                   A  16
    1 K=K+1                                                             A  17
      GO TO (2,3,4,5,6,7), K                                            A  18   2
    2 READ (F,9) NAME                                                   A  19   3
      IF (NAME(1).EQ.NN) STOP                                           A  20   6
C      CALL MAIN
      CALL LINK (6HLINKA1)                                              A  21   9
C      CALL INPUT
    3 CALL LINK (6HLINKA2)                                              A  22  10
    4 IF (L6) GO TO 6                                                   A  23  11
      CALL LINK (6HLINKA3)                                              A  24  14
    5 IF (L6) GO TO 6                                                   A  25  15
      CALL LINK (6HLINKB3)                                              A  26  18
    6 K=5                                                               A  27  19
      IF (L7) GO TO 7                                                   A  28  20
      CALL LINK (6HLINKA4)                                              A  29  23
    7 IF (L12) GO TO 8                                                  A  30  24
      K=0                                                               A  31  27
      GO TO 1                                                           A  32  28
C                                 ***                                   A  33
C        IF DATA IS RUN IN HALFS SET UP PARAMETERS                      A  34
C                                 ***                                   A  35
    8 L12=.FALSE.                                                       A  36  29
      L13=.TRUE.                                                        A  37  30
      L5=.FALSE.                                                        A  38  31
      L6=.TRUE.                                                         A  39  32
      L7=.FALSE.                                                        A  40  33
      M=MPO1/2                                                          A  41  34
      K=1                                                               A  42  35
      WRITE(6,100) M                                                        36
  100 FORMAT(//34H SECOND HALF, NUMBER OF POINTS IS ,I10)                   39
      GO TO 1                                                           A  43  39
      ENTRY HNAME                                                       A  44  40
C                                 ***                                   A  45
C     PRINT ID NAME                                                     A  46
C                                 ***                                   A  47
      WRITE (6,FMT22) NAME                                              A  48  40
      RETURN                                                            A  49  43
C                                 ***                                   A  50
C                                                                       A  51
C                                 ***                                   A  52
    9 FORMAT (3A6)                                                      A  53  44
      END                                                               A  54- 44
```

```
<><><>
```

```
$      FORTRAN FTRCK,COMDB
       SUBROUTINE PAT                                                   MAIN
C      LINKAGE                                                          C   1
       DIMENSION LL(14), PAR(12), CH(14)                                C   2
       COMMON HPD',FORMT1(12)                                           C   3
       COMMON F                                                         C   4
       COMMON FMT1(19),FMT2(3),FMT9(9),FMT1(6),FMT12(7),FMT13(3),FMT14(2  C   5
      $1,FMT16(2),FMT22(3),FMT56(8),FMT97(4),FMT99(8)                   C   6
       COMMON LA,LV,L10,L11,L12,L13,L14                                 C   7
       COMMON NP,M,LEVEL,K71,KCOUNT(34),:TAPE,AMAT(629),L7,L1,L2,L3,L4,L5  C   8
      $,L6,MX,MY,KOUNT,L7,M4,MR,MC,ZZ(2+09B)                            C   9
       LOGICAL LL,L1,L3,L4,L5,L2,L6,L7,L7                               C  10
       LOGICAL L8,L9,L10,L11,L12,L13,L14                                C  11
       LOGICAL SS                                                       C  12
       INTEGER F                                                        C  13
       REAL A(81,K(P)                                                   C  14
       DATA CH/ANCORPTS,6HPHOPTS,6HALTPCH,6HSTOPNT,6HPATTAP,6HPATERN,6HPN  C  15
      1OUT1,6HFLTING,6HPHOUT2,6HNPUNC4,6HTERMNL,6HDIAGON,6HMLFNWF,6HARSON  C  16
      $7/                                                               C  17
       DATA BLK/AH     /                                                C  18
       DATA STP/AHSTOP  /                                               C  19
       DATA NOR/AHNORMAL/                                               C  20
C                          ***                                          C  21
C      READ IN PARAMETERS ON /INPUT AND /OUTPUT CARDS                   C  22
C                          ***                                          C  23
       READ (F,21) PAR                                                  C  24
       WRITE(A,71) PAR                                                  C  25
       IF (PAR(1).EQ.STP) GO TO 3                                       C  26
C                          ***                                          C  27
C      INITIALIZE LL                                                    C  28
C                          ***                                          C  29
       DO 1 I=1,14                                                      C  30
     1 LL(I)=.TRUE.                                                     C  31
C                          ***                                          C  32
C      SEPA PARAMETERS IN LIST                                          C  33
C                          ***                                          C  34
       DO 2 I=1,12                                                      C  35
       IP=I                                                             C  36
       SS=.TRUE.                                                        C  37
       IF(PAR(I).EQ.BLK) GO TO 2                                        C  38
       DO 200 J=1,14                                                    C  39
       IF(PAR(I).EQ.CH(J)) GO TO 200                                    C  40
       LL(J)=.FALSE.                                                    C  41
       SS=.FALSE.                                                       C  42
  200  CONTINUE                                                         C  43
       IF(SS) GO TO 300                                                 C  44
     2 CONTINUE                                                         C  45
       IF (.NOT.(LL(1).AND.LL(2))) GO TO 4                              C  46
C                          ***                                          C  47
C      IF NEITHER PHOPTS OR CORPTS IS SPECIFIED PRINT ERROR AND ABORT   C  48
C                          ***                                          C  49
       WRITE (6,27)                                                     C  50
     3 STOP                                                             C  51
C                          ***                                          C  52
C      INITIALIZE L                                                     C  53
C                          ***                                          C  54
     4 L1=.FALSE.                                                       C  55
       L2=.TRUE.                                                        C  56
       L3=.TRUE.                                                        C  57
       L4=.TRUE.                                                        C  58
       L5=.TRUE.                                                        C  59
       L6=.TRUE.                                                        C  60
       L8=.FALSE                                                        C  61
       L9=.TRUE.                                                        C  62
       L10=.TRUE.                                                       C  63
       L11=LL(11)                                                       C  64
       L12=.FALSE.                                                      C  65
       L13=.FALSE.                                                      C  66
C                          ***                                          C  67
C      TRANSFER LL TO L                                                 C  68
C                          ***                                          C  69
       L14=.FALSE.                                                      C  70
       IF (.NOT.LL(14)) L14=.TRUE.                                      C  71
       IF (.NOT.LL(13)) L12=.TRUE.                                      C  72
C  LOAD FORMATS FOR EITHER TERMINAL OR COMPUTER UNIT DEPENDING ON LL(11)
       CALL FRMT(LL(11))
       IF (LL(10)) L10=.FALSE.                                          C  73
       IF (LL(8)) L4=.TRUE.                                             C  74
       IF (LL(9)) L9=.FALSE.                                            C  75
C      IF PATERN IS SPECIFIED CONTINUE                                  C  76
       IF (LL(9)) GO TO 5                                               C  77
       L3=.FALSE                                                        C  78
       L2=.FALSE.                                                       C  79
       L6=.FALSE                                                        C  80
     5 IF (LL(4)) GO TO 6                                               C  81
       L5=.FALSE.                                                       C  82
     6 IF (LL(3)) GO TO 7                                               C  83
       L4=.FALSE                                                        C  84
       L3=.FALSE                                                        C  85
C      LL(3) FALSE MEANS CORPTS WAS SPECIFIED                           C  86
C      LL TRUE MEANS CORPTS WAS  SPECIFIED                              C  87
     7 IF (LL(1)) GO TO 8                                               C  88
       L1=.TRUE.                                                        C  89
C                          ***                                          C  90
C      READ NO. OF PHOTOS, NO. POINTS PER PHOTO,                        C  91
C                                   DESIRED LEVEL OF QUANTIZATION.      C  92
C        AND NUMBER OF TRAINING REGIONS                                 C  93
C                          ***                                          C  94
     8 READ (F,23) NP,M,LEVEL,KR                                        C  95
       WRITE(A,23) NP,M,LEVEL,KR                                        C  96
       MPB1=M                                                           C  97
C      L7 FALSE MEANS PHOUT1 WAS SPECIFIED                              C  98
       L7=.FALSE.                                                       C  99
       IF (LL(7)) L7=.TRUE.                                             C 100
```

```
      IF (L.(7)) GO TO 12
C                                   ***                                     C  86
C     READ IF PHOTO SIZE HR-HORIZONTAL  HR-VERTICAL                         C  87
C                                   ***                                     C  88
      READ (5,26) HX,HY                                                     C  89
      WRITE(6,26) HX,HY                                                     C  90
      HXHX=
      HYHY=                                                                 C  91
      WRITE(6,1)                                                            C  92
      IF(HXHY.EQ.0) GO TO 31
   31 CONTINUE
C      IF PATERN IS SPECIFIED GO TO 12
      IF (.NOT.LL(A)) GO TO 12
      MC=0                                                                  C  93
C                                   ***                                     C  94
C     IF PATERN NOT USED READ IN NUMBERS AND CLASSIFICATIONS                C  95
C     STARTING BY BIGHTS ON FILE 20                                         C  96
C                                   ***                                     C  97
      READ (5,29) NCLAS                                                     C  98
      REWIND 20                                                             C  99
      JONCLAS=0                                                             C 100
      IF (J.EQ.0) GO TO 10                                                  C 101
      DO 9 I=1,J                                                            C 102
      READ                                                                  C 103
      WRITE (20) K                                                          C 104
      MC=MC+1                                                               C 105
      READ (5,26) (A(K),B(K),K=1,K)                                         C 106
      WRITE (20) (A(K),K=1,K)                                               C 107
    9 WRITE (20) (B(K),K=1,K)                                               C 108
   10 JONCLAS=JONCLAS+J                                                     C 109
      IF (J.EQ.0) GO TO 11                                                  C 110
      WRITE (20) J                                                          C 111
      MC=MC+1                                                               C 112
      READ (5,26) (A(K),B(K),K=1,J)                                         C 113
      WRITE (20) (A(K),K=1,J)                                               C 114
      WRITE (20) (B(K),K=1,J)                                               C 115
   11 JON                                                                   C 116
      WRITE (20) J                                                          C 117
C                                   ***                                     C 118
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   C 119
C                                   ***                                     C 120
      REWIND 20                                                             C 121
C                                   ***                                     C 122
C     IF STORMT IS SPECIFIED , READ IN THE PHOTO SIZE (HX,HY)               C 123
C                                   ***                                     C 124
   12 IF (L9) GO TO 14                                                      C 125
      IF (L7) GO TO 13                                                      C 126
      HXHX=                                                                 C 127
      HYHY=                                                                 C 128
      GO TO 14                                                              C 129
   13 READ (F,27) HX,HY                                                     C 130
      WRITE(6,27) HX,HY                                                     C 131
      HPR(NF)=2
      IF(HXHY.NE.0) GO TO 30
   32 CONTINUE
C                                   ***                                     C 132
C     IF PATERN IS SPECIFIED, READ IN THE LOSS MATRIX TO BE USED            C 133
C       (KK X KK )                                                          C 134
C                                   ***                                     C 135
   14 IF (L6) GO TO 20                                                      C 136
      KKKKKKK=2                                                             C 137
C     KKK IS THE NUMBER OF TRAINING REGIONS SQUARED.
C     IF DIAGON IS SPECIFIED SET UP DIAGONAL GAIN MATRIX
      IF (LL(1P)) GO TO 17
      DO 15 I=1,625                                                         C 138
   15 AMAT(I)=0.0                                                           C 139
      DO 16 I=1,KK                                                          C 140
      II=(I-1)*KK+I                                                         C 141
   16 AMAT(II)=1.00                                                         C 142
      GO TO 18                                                              C 143
   17 READ (F,28) (AMAT(I),I=1,KK*                                          C 144
   18 DO 19 I=1,KKK                                                         C 145
   19 AMAT(I)=-AMAT(I)                                                      C 146
      LT=.FALSE.                                                            C 147
C                                   ***                                     C 148
C     READ IN NORMALIZED OR UNNORMALIZED                                    C 149
C     LT TRUE SIGNALS NORMALIZED                                            C 150
C                                   ***                                     C 151
      READ (F,29) NORM                                                      C 152
      IF (NORM.EQ.NO*) LT=.TRUE.                                            C 153
   20 CALL CHART                                                            C 154
      STOP                                                                  C 155
C                                   ***                                     C 156
C                                   ***                                     C 157
C                                   ***                                     C 158
   21 FORMAT (1X,6(10.4X)/1X,6(10.4X))                                      C 159
   22 FORMAT (1X,20HINCORRECT PARAMETER CARD)                               C 160
   23 FORMAT (9I10)                                                         C 161
   24 FORMAT (2I10)                                                         C 162
   25 FORMAT (I10)                                                          C 163
   26 FORMAT (7X,10A,I10X)                                                  C 164
   27 FORMAT (2I10)                                                         C 165
   28 FORMAT (8F10.5)                                                       C 166
   29 FORMAT (12A6)                                                         C 167
   30 CONTINUE                                                              C 168
      HPRPRPRPRPR=
      WRITE(6,100) N,NS,NY,NPROB,NPRTB
  100 FORMAT(1X,20HTOTAL NUMBER OF POINTS WAS PURPORTED TO BE ,I10/
     1I2H BUT NEMPTY ,I5,1H(,I5,3H ( ,I10/31H SET TOTAL NUMBER OF POINTS
     2 AT ,I10)
      NUMPOS=0
      GO TO (131,321),NPRINT
  300 WRITE(6,301) PARIIP)
  301 FORMAT(10,6A1,9H IS A BAD PARAMETER ON /INPUT OR /OUTPUT CARDS)
      STOP
      END                                                                   - END.
```

```
17964 01  05-02-69   10.163

 1            SUBROUTINE FFRM(L)                                                  A   1
 2            COMMON MPO1,FORMT1(12)                                              A   2
 3            COMMON F                                                            B   3
 4            COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2  B   4
 5           1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)                      A   5
 6            DIMENSION O(64), T(64)                                              B   6
 7            LOGICAL L                                                           B   7
 8            INTEGER F                                                           A   8
 9            DATA T/6H(//30X,6H,11HLA.6HBLE TA.6HBLE //.6H 16X,2.6H0MPLOT,6H SV  A   9
10           1MBO.6HLS VS .6HIMPUT .6HSYMBOL.6HS  //2.6H6X,A1,.6H12X,A6.6H)      A  10
11           2.6H       .6H(/26X,.6HA1,12X.6H.A6)  .6H(27X.1.6H0NTRAI.6HNING R.6H A  11
12           3EGIONS.6H //)  .6H(24X.2.6H0MPHOT.6HO CLAS.6HSIFICA.6HTION /.6H/)  B  12
13           4   .6H(//15X.6H.29HTR.6HAINING.6H REGIO.6HN CLAS.6HSIFIED.6H AS /)  A  13
14           5.6H(5X.8(.6H5X.I2).6H/)     .6H(1H+.2.6HX.A1) .6H(3X.A1.6H)     .6H A  14
15           6(1H1.2.6H5X.3A6.6H//)   .6H(24X.2.6H2HTHE .6HPROBAB.6HILITY .6HHAT A  15
16           7RIX.6H)     .6H(6X.I2.6H.8(2X.,6HF4.2)).6H      .6H(/////.6H//21X. B  16
17           8.6H17HTOT.6HAL PRO.6HBABILI.6HTY //2.6H3X.1E1.6H2.4)  /            A  17
18            DATA O/6H(//60X,6H.11HLA.6HBLE TA.6HBLE //.6H51X.31.6HMPLOT .6HSYM A  18
19           1BOL.6HS  VS .6HIMPUT .6HSYMBOL.6HS //56.6HX,A1.1.6H2X.A6).6H      B  19
20           2.6H       .6H(/56X,.6HA1,12X.6H.A6)  .6H(57X.1.6H0NTRAI.6HNING R.6H B  20
21           3EGIONS.6H //)  .6H(54X.2.6H0MPHOT.6HO CLAS.6HSIFICA.6HTION /.6H/)  B  21
22           4   .6H(//58X.6H.29HTR.6HAINING.6H REGIO.6HN CLAS.6HSIFIED.6H AS /)  B  22
23           5.6H(4X.8(.6H13X.I2.6H)/)   .6H(1H+.1.6HX.A1) .6H (2X.A.6H1)    .6H B  23
24           6(1H1.5.6H5X.3A6.6H//)   .6H(54X.2.6H2HTHE .6HPROBAB.6HILITY .6HHAT B  24
25           7RIX.6H )    .6H(5X.I2.6H.8(3X.,6HE12.4).6H)     .6H(/////.6H//51X. B  25
26           8.6H17HTOT.6HAL PRO.6HBABILI.6HTY  ./.6H/53X.1.6HE12.4)/            B  26
27      C                          ***                                           A  27
28      C     SETUP FORMATS FOR USE IN THE PROGRAM                               A  28
29      C                          ***                                           B  29
30            IF (L) GO TO 2                                                     A  30
31            DO 1 I=1.64                                                        B  31    4
32          1 FMT1(I)=T(I)                                                       B  32    5
33            RETURN                                                             B  33    7
34          2 DO 3 I=1.64                                                        B  34    8
35          3 FMT1(I)=O(I)                                                       B  35    9
36            RETURN                                                             A  36   11
37            END                                                                B  37-  12

<><><>
```

```
$       FORTRAN BLOCK,COMMON
CINT          INPUT LINEA2                                                  INPUT
      SUBROUTINE INPUT
C       LINEA2                                                              F   1
        DIMENSION LARGE(20), LSL(20), ACC(20)
        DIMENSION TRAIN(8000), STO(36), KCOUNT(36), DATA(8000), NCOUNT(37)  F   3
       1 , STA(8000),FORM(12)                                               F   4
        COMMON MPO1,FORM(12)
        COMMON F                                                            F  10
        COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(8),FMT12(7),FMT13(5),FMT14(2  F  11
       1),FMT16(7),FMT22(3),FMT5A(4),FMT57(4),FMT58(4)                      F  12
        COMMON LA,L10,L11,L12,L13,L14                                       F  13
        COMMON N,M,LEVEL,M71,NCOUNT,ITAPE,NSAT(6079),L7,L1,L2,L3,L4,L5,L6,M F  14
       1P,MS,MT,MDUMY,L7,MT1,MT2,MT3,ZZ(74000)                              F  15
        EQUIVALENCE (ZZ(1),STA(1)),(ZZ(8001),DATA(1)),(ZZ(16001),TRAIN(1))  F  16
        LOGICAL LOG,L7,L1,L2,L3,L4,L5,L6,L7
        LOGICAL LA,L10,L11,L12,L13                                          F  20
        LOGICAL L14                                                         F  21
        DIMENSION IPRIME(20)
        INTEGER F
        INTEGER DATA,STA                                                    F  22
        DATA BLANK/1H /                                                     F  23
        DATA IPRIME/1,2,4,5,8,10,12,16,20,24,28,30,32,34,40,44,48,49,42,96/ F  24
C                          ***
C                                                                           F  25
C       TAPE 11 STORES THE SECOND HALF OF THE TRAINING REGION AND THE       F  26
C       SECOND HALF OF THE DATA FOR M.EQ.MT
C       TAPE 9 STORES THE ORIGINAL DATA SET BEING PRESENTLY WORKED ON
C       THIS IS THE ENTIRE DATA SET IF AN ENTIRE CLASSIFICATION IS DONE
C       OR IT IS THE FIRST HALF OF THE DATA IF MLEQMT
C
C       LEVEL= LEVEL OF QUANTIZATION                                        F  27
C       MZ1=NO. OF SIGNIFICENT FIG. IN INPUT DATA M71 .LE. 9                F  28
C       M=NO. OF DATA POINTS PER PHOTO                                      F  29
C       N=NO. OF PHOTOS                                                     F  33
C                                                                           F  31
C                                                                           F  32
C                          ***
        IF (.NOT.L4) WRITE (43,84)                                          F  33
C                          ***                                              F  34
C       INITILIZE KCOUNT ,MPO,K,KK,ITAPE                                    F  35
C                          ***                                              F  36
        DO 1 I=1,36                                                         F  37
      1 KCOUNT(I)=0                                                         F  38
        IF (L13.OR.L12) REWIND 11                                           F  39
        MPO=M                                                               F  40
        K=1                                                                 F  41
        KK=1                                                                F  42
        ITAPE=1                                                             F  43
        MM1=MS                                                              F  44
        MM2=MT
C                          ***
C       JUMP TO INPUT SECTIONS ON TYPE OF DATA LIKTRUP SIGNALS CORRYG        F  45
C                          ***                                              F  46
        IF (L1) GO TO 30                                                    F  47
C                          ***                                              F  48
C       IF NEITHER PATERN OR ALTERNATE PUNCH  GO TO 41                      F  49
C                          ***                                              F  50
        IF (L5.AND.L2) GO TO 9                                              F  51
C                          ***                                              F  52
C       L13 =TRUE SIGNALS SECOND HALF DATA                                  F  53
C                          ***                                              F  54
C       L12 IS TRUE WHEN MLEQMT IS NOSCIFIED AND THE FIRST HALF OF DATA IS  F  55
C       RUN.   FOR THE SECOND HALF, L12 IS FALSE
        IF (.NOT.L13) GO TO 2
        READ (11) (TRAIN(I),I=1,M)                                          F  56
        GO TO 600                                                           F  57
C                          ***
C       READ IN FORMAT AND DATA                                             F  58
C                          ***                                              F  59
      2 READ (F,95) FORM                                                    F  60
        WRITE(6,101) FORM                                                   F  61
  101 FORMAT(20H FORMAT FOR TRAINING DATA IS ,12A6)
        READ (F,FORM) (TRAIN(I),I=1,M)
        IF (L12) WRITE (11) (TRAIN(I),I=1,M)                                F  63
        IF (.NOT.L13) GO TO 3                                               F  64
  600 CONTINUE                                                              F  65
C       SECOND HALF
        MM1=MS-MS/2
        MM2=MT
        IF(MM1-MM2.GE.MM3) GO TO 3
        DO 200 JAC=1,20
        IF(MM3/IPRIME(JAC).LE.IPRIME(JAC).AND.MM3) GO TO 200
        INDEX=IPRIME(JAC)
        INDEX=MM3/IPRIME(JAC)
  200 CONTINUE
        MM1=INDEX
        MM2=INDEX
      3 IF (.NOT.L12) GO TO 9
C       FIRST HALF                                                          F  70
        J=1
        DO 4 I=3,M,2                                                        F  71
        J=J+1                                                               F  72
      4 TRAIN(J)=TRAIN(I)                                                   F  73
        MM1=M/2                                                             F  74
        IF (2*MM1.NE.M) MM1=MM1+1                                           F  75
        MM1=MS/2
        MM2=MT
        IF(MM1-MM2.GE.MM3) GO TO 9
        DO 300 JAC=1,20
        IF(MM3/IPRIME(JAC).LE.IPRIME(JAC).AND.MM3) GO TO 300
        INDEX=IPRIME(JAC)
        INDEX=MM3/IPRIME(JAC)
  300 CONTINUE
        MM1=INDEX
```

```
      MM2=INTEGR
    9 NCOUNT(1)=1                                                       E  81
      STO(1)=TRAIN(1)                                                   E  82
C      STO IS THE ARRAY OF UNIQUE SYMBOLS USED IN TRAINING REGION
C      NCOUNT(I) IS THE NUMBER OF TIMES THE ITH UNIQUE SYMBOL OCCURRED
C      IN TRAINING REGION
      DO 8 I=2,M                                                        E  83
    6 IF (TRAIN(I).EQ.STO(K)) GO TO 7                                   E  84
      K=K+1                                                             E  85
      IF (K.LE.KK) GO TO 6                                              E  86
      STO(K)=TRAIN(I)                                                   E  87
      NCOUNT(K)=1                                                       E  88
      KKKK=K                                                            E  89
      GO TO 8                                                           E  90
    7 NCOUNT(K)=NCOUNT(K)+1                                             E  91
    8 K=1                                                               E  92
C                          ***                                          E  93
C     NCOUNT CONTAINS THE NUMBER OF POINTS IN EACH TRAINING REGION      E  94
C       KK IS THE NO. OF TRAINING REGIONS                               E  95
C     L3=FALSE SIGNALS PATERN OR LINEAR                                 E  96
C                          ***                                          E  97
    9 IF (L3) GO TO 76                                                  E  98
      NCOUNT(1)=0                                                       E  99
      MMP1=KKK+1                                                        E 100
C                          ***                                          E 101
C     NCOUNT CONTAINS CUMULATIVE TRAINING REGIONS                       E 102
C                          ***                                          E 103
      DO 10 I=2,MMP1                                                    E 104
   10 NCOUNT(I)=NCOUNT(I-1)+NCOUNT(I-1)                                 E 105
      REWIND 3                                                          E 106
      IF(L3) GO TO 11
C                          ***                                          E 108
C     READ FORMAT FOR DATA                                              E 109
C                          ***                                          E 110
      READ (F,A9) FORM                                                  E 111
C                          ***                                          E 112
C     SET UP OUTPUT FORMAT                                              E 113
C                          ***                                          E 114
      CALL CHANGE (FORM,FORM1)                                          E 115
C      WRITE OUT NAME OF RUN
   11 CALL HNAME                                                        E 116
C      WRITE OUT ORIGINAL DATA
      WRITE (6,8A)                                                      E 117
C                          ***                                          E 118
C       *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *   E 119
      IF(.NOT.L13) WRITE(6,100) FORM,FORM1
  100 FORMAT(/27H INPUT FORMAT FOR DATA WAS ,12A6/
     1 27H OUTPUT FORMAT FOR DATA IS ,12A6)
C     READ IN DATA AND DETERMINE LARGEST AND SMALLEST DATA VALUES       E 120
C                          ***                                          E 121
      DO 12 I=1,N                                                       E 122
      LARGE(I)=0                                                        E 123
   12 LSL(I)=1000000000                                                 E 124
      REWIND 9                                                          E 125
      DO 18 J=1,N                                                       E 126
      WRITE(6,777) J
  777 FORMAT(1H1,10X,6HPHOTO ,I9)
C      L13 IS TRUE IF MLFNHF IS SPECIFIED AND PROGRAM IS WORKING ON
C      SECOND HALF OF DATA
      IF (.NOT.L13) GO TO 13                                            E 127
      READ (11) (DATA(I),I=1,M)                                         E 128
      GO TO 15                                                          E 129
   13 READ (F,FORM) (DATA(I),I=1,M)                                     E 130
      IF (.NOT.L12) GO TO 15                                            E 131
      WRITE (11) (DATA(I),I=2,M,2)                                      E 132
      JWRONG=1
      DO 14 I=3,M,2                                                     E 134
      JWRONG=JWRONG+1
      DATA(JWRONG)=DATA(I)
   14 CONTINUE                                                          E 137
   15 DO 666 I=1,MM
      MIS=(I-1)*MM1+1
      MIF=I*MM1
  666 WRITE(6,FORM1) (DATA(M),M=MIS,MIF)
C      L8 FALSE MEANS THAT FLTING WAS SPECIFIED
      IF (.NOT.L8) GO TO 17                                             E 140
      DO 16 I=1,MPQ
      IF (DATA(I).GT.LARGE(J)) LARGE(J)=DATA(I)                         E 142
   16 IF (DATA(I).LT.LSL(J)) LSL(J)=DATA(I)                             E 143
   17 WRITE (9) (DATA(I),I=1,M)                                         E 144
   18 CONTINUE                                                          E 145
      MMPQ                                                              E 146
       L8 IS TRUE IF FLTING WAS NOT SPECIFIED
      IF(L8) GO TO 190
C                          ***                                          E 147
C     IF DATA IS FLOATING USE FPR TO DETERMINE LARGEST AND SMALLEST VALU E 148
C                          ***                                          E 149
      CALL FPR (LSL,LARGE,N,M)                                          E 150
      GO TO 19
C                          ***                                          E 151
C       *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *   E 152
C     QUANTIZIZE DATA AND STORE ON TAPE 3                               E 153
C                          ***                                          E 154
  190 IF(.NOT.L14) GO TO 19
C      DATA IS TO BE ABSOLUTELY QUANTIZED
      LLARGE=LARGE(1)
      LLSL=LSL(1)
      DO 667 J=2,N
      IF(LLARGE .L. LARGE(J)) LLARGE=LARGE(J)
      IF(LLSL.GT.LSL(J)) LLSL=LSL(J)
  667 CONTINUE
      DO 668 J=1,N
      LARGE(J)=LLARGE
  668 LSL(J)=LLSL
```

```
   19 REWIND 9
      DO 20 J=1,N
   20 FCC(J)=FL(DATA(LEVEL-1)/FL(DATA(ARBE(J)-    (J))
      DO 22 J=1,N
      READ (9) (DATA(I),I=1,M)
      DO 21 I=1,N
      DATA(I)=FL(DATA(DATA(I))-LSL(J))*FCC(J)+.5
   21 CONTINUE
C      TAPE 3 HAS QUANTIZED DATA
      WRITE (3) (DATA(I),I=1,M)
   22 CONTINUE
C                          ***
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C                           **
      IC=1
      C=1.0
      REWIND 1
      REWIND 2
C                          ***
C     L2 FALSE SIGNALS TAPE 1 ST BE CREATED FOR USE IN PATERN
C                          ***
      IF (L2) GO TO 52
C                          ***
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C                          ***
   23 DO 24 I=1,KK
   24 KCOUNT(I)=0
      READ (3) (DATA(I),I=1,M)
      DO 27 I=1,N
      DO 25 K=1,KK
      IF (TRAIN(I).EQ.TTO(K)) GO TO 26
   25 CONTINUE
   26 KCOUNT(K)=KCOUNT(K)+1
      JONCOUNT(K)=KCOUNT(K)
   27 STA(J)=DATA(I)
      JONCOUNT(K)=1
      WRITE (1) C,(STA(I),I=1,J),C
      IC=IC+1
      IF (IC.LE.N) GO TO 23
      REWIND 2
      REWIND 1
      REWIND 3
      DO 29 II=1,KK
      DO 28 LL=1,N
      JONCOUNT(II)=1
      LONCOUNT(II)
      KONCOUNT(KK+1)=2-J-L
      READ (1) (SS,I=1,J),(DATA(I),I=1,L),(SS,I=1,K)
   28 WRITE (2) (DATA(I),I=1,L)
   29 REWIND 1
      GO TO 48
C                          ***
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C     CORPTS INPUT
C                          ***
   30 REWIND 3
      REWIND 9
      DO 31 I=1,N
      LABEL(I)=0
   31 LSL(I)=1000000000
      IF (.NOT.L13) GO TO 32
C     SECOND HALF
      MM1=MM-MM/2
      MM2=MM
      IF(MM1.GE.MM2.AND.MPG) GO TO 32
      DO 400 JACK=1,20
      IF(MPG/IPRIME(JACK)*IPRIME(JACK).NE.MPG) GO TO 400
      INDEXX=IPRIME(JACK)
      INDEXV=MPG/IPRIME(JACK)
  400 CONTINUE
      MM1=INDEXX
      MM2=INDEXV
   32 IF (.NOT.L12) GO TO 33
      MPG=M/2
      IF (MPG*2.NE.M) MPG=MPG+1
      MM1=MM/2
      MM2=MM
      IF(MM1.GE.MM2.AND.MPG) GO TO 33
      DO 900 JACK=1,20
      IF(MPG/IPRIME(JACK)*IPRIME(JACK).NE.MPG) GO TO 900
      INDEXX=IPRIME(JACK)
      INDEXV=MPG/IPRIME(JACK)
  900 CONTINUE
      MM1=INDEXX
      MM2=INDEXV
   33 IF (L13) GO TO 34
C                          ***
C     READ IN FORMAT FROM DATA
C                          ***
      READ (F,05) FORM
      CALL CHANGE (FORM,FORM1)
   34 CALL NAME
      WRITE (6,06)
C                          ***
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
C     OBTAIN DATA AND FIND MAX AND MIN
C                          ***
      IF(.NOT.L13) WRITE(6,100) FORM,FORM1
      DO 30 J=1,M
      IF (.NOT.L12) GO TO 39
      READ (11) TRAIN(J),(DATA(I),I=1,N)
      GO TO 36
   39 READ (F,FORM) TRAIN(J),(DATA(I),I=1,N)
      IF (.NOT.L12.OR.J/2*2.NE.J) GO TO 36
```

```
C       USE ALL 800 DISS FOR FIRST HALF
        WRITE (13) TRAIN(J),(DATA(I),I=1,N)                                     E 244
        GO TO 38
   36   WRITE (8,FORMAT) TRAIN(J),(DATA(I),I=1,N)                               E 246
        IF (.NOT.LA) GO TO 45                                                   E 247
C       DATA IS IN LARGER FORMAT
        DO 37 I=1,N                                                             E 248
        IF (DATA(I).LT.LSL(I)) LSL(I)=DATA(I)                                   E 249
   37   IF (DATA(I).GT.LARGE(I)) LARGE(I)=DATA(I)                               E 250
   38   WRITE (8) (DATA(I),I=1,N)                                               E 251
        IF(.NOT.LT) GO TO 800
        JWRONG=1
        DO 801 I=1,N,2
        JWRONG=JWRONG+1
  801   TRAIN(JWRONG)=TRAIN(I)
  800   CONTINUE
        REWIND                                                                  E 257
C       LA IF TURN IF DATA IS IN INTEGER FORM
        IF(LA) GO TO 390
        CALL FIRT (LSL,LARGE,N,M)                                               E 254
        GO TO 39
  390   IF(LAD.LE.1) GO TO 39
C       DATA IS IN ABSOLUTELY QUANTIZED
        LLARGE=LARGE(1)
        LLSL=LSL(1)
        DO 700 J=2,N
        IF(LLARGE.LT.LARGE(J)) LLARGE=LARGE(J)
        IF(LLSL.GT.LSL(J)) LLSL=LSL(J)
  700   CONTINUE
        DO 701 J=1,N
        LARGE(J)=LLARGE
  701   LSL(J)=LLSL
C                                   ***                                         E 255
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     E 256
C       QUANTIZE DATA AND BUILD TRAINING REGION ARRAY WITH NCOUNT [illegible] NTD  E 257
C                                   ***                                         E 258
   39   REWIND 8                                                                E 259
        DO 50 J=1,N                                                             E 260
   40   ACC(J)=(F(DATA(LEVEL-1))/F(DATA(LARGE(J))-LSL(J)))                      E 261
        DO 42 J=1,N                                                             E 262
        READ (8) (DATA(I),I=1,N)                                                E 263
        DO 41 I=1,N                                                             E 264
        DATA(I)=F(DATA(DATA(I)-LSL(I))*ACC(I))+1.5                              E 265
   41   CONTINUE                                                                E 266
        WRITE (3) (DATA(I),I=1,N)                                               E 267
   42   CONTINUE                                                                E 268
        REWIND 2                                                                E 269
        STOR(1)=TRAIN(1)                                                        E 270
        NCOUNT(1)=1                                                             E 271
        DO 45 K=2,M                                                             E 272
   43   IF (TRAIN(K).EQ.STOR(K)) GO TO 44                                       E 273
        KKKK=K                                                                  E 274
        IF (K.LE.M) GO TO 43                                                    E 275
        STOR(KKKK)=TRAIN(K)                                                     E 276
        NCOUNT(KKKK)=1                                                          E 277
        KKKK=KKKK+1                                                             E 278
        GO TO 45                                                                E 279
   44   NCOUNT(KKKK)=NCOUNT(KKKK)+1                                             E 280
   45   KK=1                                                                    E 281
        REWIND 3                                                                E 282
        IF (L.EQ.1) GO TO 52                                                    E 283
        DO 47 KK=1,KK                                                           E 284
        DO 47 I=1,N                                                             E 285
        REWIND 3                                                                E 286
        II=1                                                                    E 287
C                                   ***                                         E 288
C       STORE        ING REGIONS IN TAPE 3                                      E 289
C                                   ***                                         E 290
        DO 46 J=1,N                                                             E 291
        READ (3) (DATA(I),I=1,N)                                                E 292
        IF (TRAIN(J).NE.STOR(KK)) GO TO 46                                      E 293
        DATA(II)=DATA(I)                                                        E 294
        II=II+1                                                                 E 295
   46   CONTINUE                                                                E 296
C                                   ***                                         E 297
C       STORE DATA ON TAPE 2                                                    E 298
C                                   ***                                         E 299
   47   WRITE (2) (DATA(I),I=1,II)                                              E 300
C                                   ***                                         E 301
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     E 302
C       QUANTIZE DATA IN INTEGER LEVELS FROM 1 THROUGH LEVEL                    E 303
C       PLACE RESULTS ON TAPE ONE DIMENSION BY DIMENSION                        E 304
C                                   ***                                         E 305
   48   REWIND 3                                                                E 306
        REWIND 1                                                                E 307
        REWIND 2                                                                E 308
        IF (F(DATA(LEVEL)).GT.1.E11) STOP                                       E 309
        DO 51 I=1,M                                                             E 310
        NNCOUNT(I)=                                                             E 311
        DO 49 J=1,N                                                             E 312
   49   STOR(J)=0                                                               E 313
        DO 50 I=1,N                                                             E 314
        READ (2) (DATA(J),J=1,N)                                                E 315
C                                   ***                                         E 316
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     E 317
C                                   ***                                         E 318
        DO 50 J=1,N                                                             E 319
   50   STOR(J)=STOR(J)+DATA(J)                                                 E 320
   51   WRITE (1) (STOR(J),J=1,N)                                               E 321
C                                   ***                                         E 322
C       PLACE DATA IN ALTERNATE FORM ON INPUT IF ALT IS SPECIFIED ON            E 323
C       THE PARAM TEN CARD                                                      E 324
C                                   ***                                         E 325
   52   IF (L.N AND.[illegible]) GO TO 15                                       E 326
```

```
      REWIND 2
      REWIND 4
      REWIND 3
      IF (L1) GO TO 72
C                          ***
C     PROMPTS TO COMPUTE SECTION
C                          ***
      NPHO=N
      LOGE.FALSE.
      IC=0
      NTRN
   53 IMAXPHO=M
      IF (.NOT.(IM.GT.1000)) GO TO 54
      NPHO=(NT*(IM-10*)1)/10+1)
      IMAXPHO=M
      LOGE.TRUE.
   54 DO 55 I=1,NPHO
      J=(I-1)*N+1
      K=J+P
   55 READ (3) (STA(I),L=J,K)
      IF (LOG) GO TO 59
      DO 56 I=1,P
      IF (.NOT.(L4) WRITE (43,87) TRAIN(I),(STA(K),K=J,IM,M)
   56 IF (.NOT.(L7) WRITE (4) (STA(K),K=J,IM,M)
      GO TO 76
   57 DO 58 I=1,P
   58 WRITE (2) (STA(K),K=J,IM,M)
      IC=IC+1
      NT=NT-NPHO
      NCOUNT(IC)=NPHO
      NPHO=NT
      IF (NT.EQ.0) GO TO 59
      GO TO 53
   59 REWIND 2
      K=1000/N
      L=M-P
      JJJ=0
      DO 61 LL=1,IC
      DO 60 I=1,P
      JJJ=JJJ+1
      JJJ=JJJ+NCOUNT(LL)-1
   60 READ (2) (STA(I),I=JJJ,JJJ)
      DO 61 J=1,L
   61 READ (2) (A,I=JJJ,JJJ)
      DO 62 I=1,P
      CALL DEF (NCOUNT,STA,NCOUNT,1,IC,M)
      IF (.NOT.(L4) WRITE (43,87) TRAIN(I),(NCOUNT(K1),K1=1,N)
   62 IF (.NOT.(L7) WRITE (4) (NCOUNT(K1),K1=1,N)
      REWIND 2
      MULT=1
   63 IF (M.LT.L) GO TO 67
      K1=MULT*M
      JJJ=0
      DO 65 LL=1,IC
      J=NCOUNT(LL)
      DO 64 I=1,P
   64 READ (2) (A,I=1,J)
      DO 65 J=1,L
      JJJ=JJJ+1
      JJJ=JJJ+NCOUNT(LL)-1
   65 READ (2) (STA(I),I=JJJ,JJJ)
      DO 66 I=1,L
      CALL DEF (NCOUNT,STA,NCOUNT,1,IC,L)
      K1=L*K*MULT
      IF (.NOT.(L4) WRITE (43,87) TRAIN(K1),(NCOUNT(K1),K1=1,N)
   66 IF (.NOT.(L7) WRITE (4) (NCOUNT(K1),K1=1,N)
      GO TO 76
   67 L=L-M
      K1=MULT*M
      JJJ=0
      DO 70 LL=1,IC
      J=NCOUNT(LL)
      DO 69 I=1,P
      JJJ=JJJ+1
      JJJ=JJJ+NCOUNT(LL)-1
   69 READ (2) (STA(I),I=JJJ,JJJ)
      DO 70 J=1,L
   70 READ (2) (A,I=1,JJJ)
      DO 71 I=1,P
      CALL DEF (NCOUNT,STA,NCOUNT,1,IC,M)
      K1=L*K*MULT
      IF (.NOT.(L4) WRITE (43,87) TRAIN(K1),(NCOUNT(K1),K1=1,N)
   71 IF (.NOT.(L7) WRITE (4) (NCOUNT(K1),K1=1,N)
      MULT=MULT+1
      REWIND 2
      GO TO 63
C                          ***
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     COMPUTE TM PROMPTS SECTION
C                          ***
   72 STOP
      CALL TRANSE(TRAIN,M,DATA)
      WRITE (43,86) (TRAIN(I),I=1,N)
      JJJ=0
   73 K=1000/M
      IF (K.EQ.0) GO TO
      DO 74 I=1,P
      READ (3) (NCOUNT(I),I=1,N)
      DO 74 I=1,P
      J=I-1*M+1
   74 STA(J)=NCOUNT(I)
      K1=1
```

```
      K2=M                                                              E 431
      JJ=JJJ+1                                                          E 432
      JJJ=JJJ+K                                                         E 433
      DO 75 I=JJ,JJJ                                                    E 434
      WRITE (43,87) I                                                   E 435
      WRITE(43,99) (STA(J),J=K1,K2)
   99 FORMAT(13(A)
      K1=K2+1                                                           E 437
   75 K2=K2+M                                                           E 438
      NT=NT-K                                                           E 439
      IF (NT.GT.0) GO TO 73                                             E 440
C                          ***                                          E 441
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  E 442
C     WRITE OUT TRAINING REGIONS CENTERED ON PAGE                       E 443
C                          ***                                          E 444
   74 IF (L5) GO TO 78                                                  E 445
      CALL TRAME (TRAIN,M,DATA)                                         E 446
      WRITE (1) (DATA(I),I=1,M)                                         E 447
      J=1                                                               E 448
      JJ=MM1
      L=(130-MM1)/2
      IF(.NOT.L11) L=(70-MM1)/2
      CALL HNAME                                                        E 452
      WRITE (6,FOT9)                                                    E 453
      DO 77 I=1,MM2
      WRITE (6,91) (BLANK,II=1,L),(TRAIN(II),II=J,JJ)                   E 455
      J=JJ+1                                                            E 456
   77 JJ=JJ+MM1
      REWIND 3                                                          E 458
   78 IF (L10) WRITE (43,94)                                            E 459
      CALL HNAME                                                        E 460
C                          ***                                          E 461
C     PTINT AND PUNCH QUANTITIZED DATA IF DESINED                       E 462
C                          ***                                          E 463
      WRITE (6,93)                                                      E 464
C      IF L1 IS TRUE THEN CORPTS WAS SPECIFIED
      IF (L1) GO TO 80                                                  E 465
      DO 79 I=1,M                                                       E 466
      WRITE(6,777) I
      READ (3) (DATA(J),J=1,M)                                          E 467
      IF (L10) WRITE (43,95) (DATA(J),J=1,M)                            E 468
   79 WRITE (6,92) (DATA(J),J=1,M)                                      E 469
      GO TO 82                                                          E 470
   80 DO 81 I=1,M                                                       E 471
      READ (3) (DATA(J),J=1, )                                          E 472
      IF (L10) WRITE (43,95) (DATA( ),J=1,M)                            E 473
   81 WRITE (6,92) (DA[illegible],J=1,M)                                E 474
   82 IF (.NOT.L6.AND.K [illegible]) GO TO [illegible]                  E 475
      KR=MAXR(KR,KK)                                                    E 476
      REWIND 1                                                          E 477
      CALL CHNXT                                                        E 478
   83 WRITE (6,96)                                                      E 479
      WRITE (6,96)                                                      E 480
      WRITE (6,96)                                                      E 481
      WRITE (6,97) KW,KR                                                E 482
      WRITE (6,98)                                                      E 483
      WRITE (6,96)                                                      E 484
      WRITE (6,96)                                                      E 485
      STOP                                                              E 486
C                          ***                                          E 487
C                                                                       E 488
C                          ***                                          E 489
   84 FORMAT (15HALTERNATE PUNCH)                                       E 490
   85 FORMAT (12A4)                                                     E 491
   86 FORMAT (1X,12HORIGINAL DATA/)                                     E 492
   87 FORMAT (A6,3X,10(1X,I10))                                         E 493
   88 FORMAT (80A1)                                                     E 494
   89 FORMAT (5HPHOTO,24,I3)                                            E 495
   90 FORMAT (15I6)                                                     E 496
   91 FORMAT (1X,131A1)                                                 E 497
   92 FORMAT (10(1X,I10))                                               E 498
   93 FORMAT (1X,14HQUANTIZED DATA/)                                    E 499
   94 FORMAT (27HQUANTIZED DATA FORMAT(8I10))                           E 500
   95 FORMAT (8I10)                                                     E 501
   96 FORMAT (100H ****************************************************** E 502
     1*****************************                                    E 503
   97 FORMAT (44H ONE OF US CAN NOT COUNT  I FIND THAT I4H96 ARE ,I3,27H E 504
     1TRAINING REGIONS  YOU FOUND,I3)                                   E 505
   98 FORMAT (//)                                                       E 506
      END                                                               E 507
```

```
 1          SUBROUTINE FPR (LSL,LARGE,M,N)                                  F   1
 2          DIMENSION SS(20), BB(20), LSL(20), LARGE(20), IDATA(13)         F   2
 3          COMMON PP01,FORMT1(12)                                          F   3
 4          COMMON                                                          F   4
 5          COMMON FMT1(19),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2  F   5
 6         1),FMT1A(2),FMT22(3),FMT50(6),FMT57(4),FMT90(8)                  F   6
 7          COMMON L8,L9,L10,L11,L12,L13,L14                                F   7
 8          COMMON N,L,B(8679),DATA(8000)                                   F   8
 9          EQUIVALENCE (IDATA(1),DATA(1))                                  F   9
10          INTEGER F                                                       F  10
11          LOGICAL L8,L9,L10,L11,L12,L13,L14                               F  11
12          REWIND 9                                                        F  12
13          REWIND 21                                                       F  13   2
14    C                          ***                                        F  14
15    C      L14 IS TRUE IF ABSOLUTE QUANTIZATION
16    C      PHOTO FORM
17    C      N IS THE NUMBER OF PHOTOGRAPHS
18    C      M IS THE NUMBER OF DATA POINTS
19    C     READ IN DATA                                                     F  19
20    C                          ***                                        F  20
21          S=1.E20                                                         F  21
22          B=-1.E20                                                        F  22   3
23    C      GO THRU EACH PHOTOGRAPH                                        F  23   4
24          DO 3 I=1,N                                                      F  24   5
25          READ (9) (DATA(J),J=1,M)                                        F  25   6
26    C                          ***                                        F  26
27    C     SEARCH FOR MAX AND MIN VALUES IN DATA                            F  27
28    C                          ***                                        F  28
29          DO 1 J=1,M                                                      F  29
30          IF (S.GT.DATA(J)) S=DATA(J)                                     F  30   11
31        1 IF (B.LT.DATA(J)) B=DATA(J)                                     F  31   12
32          IF (L14) GO TO 3                                                F  32   13
33          R=1./(B-S)*1.E10                                                F  33   14
34    C                          ***                                        F  34
35    C     QUANTITIZE AND STORE DATA ON 21                                 F  35
36    C      QUANTIZE RELATIVELY                                            F  36
37    C                          ***                                        F  37
38          DO 2 J=1,M                                                      F  38
39        2 IDATA(J)=(DATA(J)-S)*R                                          F  39   23
40          S=1.E20                                                         F  40   24
41          B=-1.E20                                                        F  41   25
42        3 WRITE (21) (IDATA(J),J=1,M)                                     F  42   26
43          REWIND 9                                                        F  43   27
44          REWIND 21                                                       F  44   28
45    C                          ***                                        F  45
46    C     STORE DATA BACK ON 9                                            F  46
47    C      FOR EACH PHOTOGRAPH                                            F  47
48    C                          ***                                        F  48
49          DO 5 I=1,N                                                      F  49
50          READ (21) (IDATA(J),J=1,M)                                      F  50   36
51          IF (.NOT.L14) GO TO 5                                           F  51   37
52    C      QUANTIZE ABSOLUTELY                                            F  52   42
53          R=1./(B-S)*1.E10                                                F  53
54          DO 4 J=1,M                                                      F  54   45
55        4 IDATA(J)=(DATA(J)-S)*R                                          F  55   46
56        5 WRITE (9) (IDATA(J),J=1,M)                                      F  56   47
57        6 DO 7 I=1,10                                                     F  57   48
58          LSL(I)=0                                                        F  58   49
59        7 LARGE(I)=10000000000                                            F  59   55
60          RETURN                                                          F  60   96
61          ENTRY FPR1(LSL,LARGE,M,N)                                       F  61   97
62          REWIND 9                                                        F  62   99
63          REWIND 21                                                       F  63   60
64    C                          ***                                        F  64   61
65    C      CORPTS FORM                                                    F  65
66    C      M IS THE NUMBER OF PHOTOGRAPHS
67    C      N IS THE NUMBER OF DATA POINTS
68    C      L14 IS TRUE FOR ABSOLUTE QUANTIZATION
69    C     INITIALIZE                                                       F  69
70    C                          ***                                        F  70
71          DO 8 I=1,M                                                      F  71
72          SS(I)=1.E20                                                     F  72   62
73        8 BB(I)=-1.E20                                                    F  73   63
74    C                          ***                                        F  74   64
75    C     SEARCH FOR EXTREME VALUES                                       F  75
76    C                          ***                                        F  76
77    C      GO THRU EACH POINT                                             F  77
78          DO 9 I=1,N                                                      F  78
79    C      READ IN ALL THE COMPONENTS FOR THAT POINT                      F  79   66
80          READ (9) (DATA(J),J=1,M)                                        F  80   67
81          WRITE (21) (DATA(J),J=1,M)                                      F  81   72
82          DO 9 J=1,M                                                      F  82   73
83          IF (SS(J).GT.DATA(J)) SS(J)=DATA(J)                             F  83   76
84        9 IF (BB(J).LT.DATA(J)) BB(J)=DATA(J)                             F  84   81
85          IF (.NOT.L14) GO TO 12                                          F  85   86
86          S=1.E20                                                         F  86   89
87          B=-1.E20                                                        F  87   90
88          DO 10 I=1,M                                                     F  88   91
89          IF (S.GT.SS(I)) S=SS(I)                                         F  89   92
90       10 IF (B.LT.BB(I)) B=BB(I)                                         F  90   95
91          DO 11 I=1,M                                                     F  91   96
92          SS(I)=S                                                         F  92   100
93       11 BB(I)=B                                                         F  93   101
94       12 REWIND 9                                                        F  94   103
95          REWIND 21                                                       F  95   104
96    C                          ***                                        F  96
97    C      QUANTIZE                                                       F  97
98    C                          ***                                        F  98
99          DO 13 I=1,M                                                     F  99
100      13 BB(I)=1./(BB(I)-SS(I))*1.E10                                    F  100  105
101         DO 19 I=1,N                                                     F  101  106
102         READ (21) (DATA(J),J=1,M)                                       F  102  108
103   C                          ***                                        F  103  109
104   C     STORE DATA ON 9                                                 F  104
105   C                          ***                                        F  105
106         DO 14 J=1,M                                                     F  106
107      14 IDATA(J)=(DATA(J)-SS(J))*BB(J)                                  F  107  114
108      19 WRITE (9) (IDATA(J),J=1,M)                                      F  108  117
109         GO TO 6                                                         F  109  123
110         END                                                             F  110  124

<>><>><>>
```

```
17964 01  09-02-69   10.269

      SUBROUTINE DEF (L,M,N,I,IC,K)
      DIMENSION L(2), M(2), N(2)
      JT=(I-1)*N(1)
C                                   ***
C     SET SINGLE CHARACTER TRAINING REGION
C     DEFINE IN ALPHABETICAL ORDER
C                                   ***
      J9=0
      KC=0
      DO 2 J=1,IC
      JJ=M(J)
      DO 1 II=1,JJ
      JT=JT+1
      J9=J9+1
    1 L(J9)=N(JT)
      KC=KC+JJ*K
    2 JT=JT+(I-1)*N(J-1)
      RETURN
      END

<<
```

```
      SUBROUTINE PATERN                                                 J   1
      COMMON NP31,FORMT1(12)                                            J   2
      COMMON F                                                          J   3
      COMMON FMT1(15),FMT2(3),FMT8(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2 J   4
     11,FMT1A(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)                    J   5
      COMMON LR,L9,L10,L11,L12,L13,L14                                  J   6
      COMMON NP,MM,LEVEL,MZ1,NPOINT,ITAPE,AAA,L7,L1,L2,L3,L4,L5,L6,N,M1, J   7
     1M2,MDUPY,L7,M3,ME,KC,ZZZ                                          J   8
C                                 ***                                   J   9
C     N=NUMBER OF TRAINING REGIONS                                      J  10
C     LEVEL=NUMBER OF LEVELS OF QUANTITIZATION                          J  11
C     NP=NUMBER OF PHOTOS                                               J  12
C     MZ1=NUMBER OF SIGNIFICANT FIGURES PER DATA POINT                  J  13
C     NPOINT=ORDERED LIST OF POINTS PER TRAINING REGION                 J  14
C      INPUT IS EXPECTED ON TAPE UNIT 'ONE'                             J  15
C     L7=LOGICAL TRUE FOR NORMALIZED AND FALSE FOR UNNORMALIZED         J  16
C                                 ***                                   J  17
      DIMENSION AAA(625)                                                J  18
      DIMENSION ZZZ(24000), IPT(2), IREM(2), NREG(2), NPOINT(36), NPT(36 J  19
     1), RESULT(625), SIGMA(36), FORM(12), BLANK(4)                     J  20
      EQUIVALENCE (ZZZ(1),IPT(1)), (ZZZ(1),IREM(1)), (ZZZ(1),NREG(1))   J  21
      LOGICAL L7,L1,L2,L3,L4,L5,L6,L7                                   J  22
      LOGICAL L8,L9,L10,L11,L12,L13                                     J  23
      INTEGER F                                                         J  24
      DATA BLANK/6HUNORMA,5HLIZED,6HNORMAL,4HIZED/                      J  25
      REWIND 1                                                          J  26
      REWIND 2                                                          J  27   2
      REWIND 8                                                          J  28   3
      REWIND 10                                                         J  29   4
      REWIND 20                                                         J  30   5
C                                 ***                                   J  31
C     PRINT HEADING FOR PUNCH                                           J  32
C     NHIGH IS ABSOLUTE MAX NUMBER OF POINTS IN A SINGLE TRAINING REGION J  33
C                                 ***                                   J  34
      WRITE (10) N                                                      J  35   6
      WRITE (43,32)                                                     J  36   9
      NHIGH=0                                                           J  37  11
      NTOTAL=0                                                          J  38  12
C                                 ***                                   J  39
C     NPOINT CONTAINS NO. OF POINTS PER TRAINING REGION                 J  40
C     COUNT TOTAL PTS.     FIND MAX IN NPOINT                           J  41
C                                 ***                                   J  42
      DO 1 I=1,N                                                        J  43  13
      NTOTAL=NTOTAL+NPOINT(I)                                           J  44  14
      IF (NPOINT(I).LT.NHIGH) GO TO 1                                   J  45  15
      NHIGH=NPOINT(I)                                                   J  46  18
    1 CONTINUE                                                          J  47  19
      J=1                                                               J  48  21
      IF (LT) J=3                                                       J  49  22
      K=J+1                                                             J  50  25
      CALL MNAME                                                        J  51  26
      WRITE (6,33) (BLANK(I),I=J,K)                                     J  52  27
      WRITE (6,34)                                                      J  53  32
C                                 ***                                   J  54
C     PRINT PARAMETERS                                                  J  55
C                                 ***                                   J  56
      WRITE (6,35) N,LEVEL,NP                                           J  57  34
C                                 ***                                   J  58
C     INITIALIZE                                                        J  59
C                                 ***                                   J  60
      DO 2 I=1,100                                                      J  61  37
    2 RESULT(I)=0.                                                      J  62  38
C                                 ***                                   J  63
C     RESULT IS PROBABILITY MATRIX                                      J  64
C     CALCULATE NO. OF LOCATIONS FOR USE IN IREM AND IPT                J  65
C     NEVER MORE THAN 1 TR IN BOTTOM OF ZZZ                             J  66
C                                 ***                                   J  67
      NG=20000-NHIGH                                                    J  68  40
      IF (NG.LT.10000) NG=10000                                         J  69  41
      KBC=0                                                             J  70  44
      NC=0                                                              J  71  45
      IP=3                                                              J  72  46
      SP=0.                                                             J  73  47
C                                 ***                                   J  74
C     CALCULATE NORMALIZING FACTORS                                     J  75
C                                 ***                                   J  76
      DO 3 I=1,N                                                        J  77  48
      SIGMA(I)=1./FLOAT(NPOINT(I))                                      J  78  49
    3 SP=SP+FLOAT(NPOINT(I))                                            J  79  50
C                                 ***                                   J  80
C     KCOUNT IS A FUNCTION OF THE POSITION IN THE TOP PART OF ZZZ       J  81
C     INDICATION WHERE THE PRESENT UNIQUE N-TUPLE(UNIQUE TO TR)         J  82
C     WILL BE STORED                                                    J  83
C     LICN IS THE BEGINNING OF THE BLOCK MI IN THE TOP HALF OF ZZZ      J  84
C     IPOI IS NUMBER OF POINTS IN MI TRAINING REGION                    J  85
C     AND LOAD IT UP IN THE BOTTOM PART OF ZZZ                          J  86
C     STARTING AT THE TOP AND COMMING DOWN                              J  87
C                                 ***                                   J  88
      KCOUNT=1                                                          J  89  52
      LICN=1                                                            J  90  53
C                                 ***                                   J  91
C     MI RANGES OVER NO. OF TRAINING REGIONS                            J  92
C                                 ***                                   J  93
      DO 8 MI=1,N                                                       J  94  54
      IPOI=NPOINT(MI)                                                   J  95  55
C                                 ***                                   J  96
C     READ IN SINGLE TRAINING REGION                                    J  97
C                                 ***                                   J  98
      READ (1) (NREG(I),I=1,IPOI)                                       J  99  56
C                                 ***                                   J 100
C     SET N-TUPLE IN 6NBN LOCATIONS                                     J 101
C     PUT NO. TIMES OCCURED IN 6BB LOCATIONS                            J 102
C     TRAINING REGION IS STORED IN LOWER HALF OF 6BB LOCATIONS          J 103
```

```
C                                   ***
      DO 7 LJ=1,IPOI
      ****MPG(LJ)
      IF (LICK.EQ.NCOUNT) GO TO 9
      II=NCOUNT-1
      DO 4 I=LICK,II
      I1=20002-I=2
      IF (NNN.+3.IREM(I1)) GO TO 6
    4 CONTINUE
C                                   ***
C     IF OVERLAP OCCURES IN SCRATCH AREA GO TO 777 AND STORE RESULTS
C     NMEG(LJ) HAS NOT OCCURED BEFOR  IN THE TRAINING REGION, NAME
C     POSITION TO STORE IT
C                                   ***
    9 IF (NCOUNT+2.GT.NS) GO TO 97
      NCOUNT=20002-NCOUNT+2
      IREM=IMCOUNT+***
      IPT(NCOUNT-1)=10000+KI=100
      NCOUNT=NCOUNT+1
      GO TO 7
C                                   ***
C     NMEG(LJ) HAS OCCURED BEFOR IN TO N1
C                                   ***
    6 I1=20001-I=2
      IPT(I1)=IPT(I1)+10000
    7 CONTINUE
      LICK=NCOUNT
    8 NNN=NCOUNT-1
C                                   ***
C     N2 IS THE LOWEST LOCATION ON THE TOP PART WHICH IS USED
C                                   ***
      I=2***
      N2=20001-I
C                                   ***
C     STORE RESULTS ON TAPE 10
C                                   ***
      WRITE (10) I
      WRITE (10) (IREM(I),I=N2,20000)
    9 IF (NRC.EQ.0) GO TO 11
      JTAPE=2
      LTAPE=3
      RE=INS 2
      RE=INS 9
      LL=I
   10 READ (JTAPE) NCOUNT
      N=20001-NCOUNT
      NM1=N2-1
      ***N2-4
      READ (JTAPE) (IREM(I),I=N,NM1)
      ****
   11 L***-1
C                                   ***
C     SEQUENCE N-TUPLES IN DECENDING ORDER
C                                   ***
      DO 12 I=1,L
      IP1=I+1
      DO 12 J=IP1,***
      I1=20002-I=2
      J1=20002-J=2
      IF (IREM(I1).GE.IREM(J1)) GO TO 12
      IPRI=IREM(I1)
      IREM(I1)=IREM(J1)
      IREM(J1)=IPRI
      IPRI=IPT(I1-1)
      IPT(I1-1)=IPT(J1-1)
      IPT(J1-1)=IPRI
   12 CONTINUE
      IF (NRC.EQ.0) GO TO 19
      WRITE (LTAPE) (IREM(I),I=N2,NM1)
      LL=LL+1
      IF (LL.LE.NRC) GO TO 10
      NP1=***+3
      ****IREM(***+1)
      DO 13 I=NP1,20000,2
      IF (IREM(I).NE.***) GO TO 1
   13 CONTINUE
   14 ***I/2
   15 CONTINUE
      NCOUNT=0
      I=1
C                                   ***
C     LOOK FOR STARTING AND END POINTS OF EACH N-TUPLE
C     LOOK FROM PRESENT N-TUPLE DOWN
C                                   ***
   16 DO 17 J=I,***
      J=IJ
      I1=20002-I=2
      J1=20002-J=2
      IF (IREM(I1).NE.IREM(J1)) GO TO 18
   17 CONTINUE
      J=***+1
   18 J=J-1
C                                   ***
C     J IS A FUNCTION OF THE LAST N-TUPLE LIKE THE I TO N-TUPLE TOP BOT
C                                   ***
      K =1
      IF (I.EQ.J) GO TO 19
C                                   ***
C     N-TUPLE OCCURS IF MORE THAN ONE TO
C                                   ***
      GO TO 20
C                                   ***
C     IF ONLY ONE T.O. IN N-TUPLE CLASSIFY N-TUPLE
C                                   ***
```

```
      10 II=20001-I*2                                                   J 208  177
C                                   ***                                 J 209
C     I1 IS THE LOCATION OF THE CONT FOR THE I TH N-TUPLE               J 210
C                                   ***                                 J 211
      N=IPT(II)/10000                                                   J 212  178
      M=(IPT(II)-N*10000)/100                                           J 213  179
C                                   ***                                 J 214
C     M IS NUMBER OF TIMES THAT THE N-TUPLE IN II+1 OCCURED             J 215
C                                   ***                                 J 216
      GO TO 21                                                          J 217  180
C                                   ***                                 J 218
C     IF TWO OR MORE T.O. IN N-TUPLE LOOK FOR PREDOMINANT T.O.          J 219
C     AND CLASSIFY                                                      J 220
C                                   ***                                 J 221
   20 CALL DIGION (I1,J,IPT,STORE,N,M,AAA,LT,NTOTAL,NPRED)              J 222  181
   21 DO 22 L=1,J                                                       J 223  182
      M1=IABS(L)                                                        J 224  183
   22 IPT(M1)=IPT(M1)+4                                                 J 225  184
      NCOUN=NCOUN+1                                                     J 226  185
C                                   ***                                 J 227
C     STORE IN MORE N-TUPLE AND CLASSIFICATION                          J 228
C     IF MORE FULL GO TO 23 AND STORE MORE                              J 229
C                                   ***                                 J 230
   23 IF (NCOUN.GT.MAXN) GO TO 30                                       J 231  187
      II=20002-I*2                                                      J 232  188
      MORE(NCOUN)=IPT(II)                                               J 233  189
      WRITE (9) N,NPRED                                                 J 234  190
      I=J+1                                                             J 235  191
      IF (I.LE.NN) GO TO 10                                             J 236  192
      NCOUN=NCOUN+1                                                     J 237  193
      IF (N.NE.0) GO TO 30                                              J 238  194
C                                   ***                                 J 239
C     CLASSIFY ALL N-TUPLES                                             J 240
C     NN IS TO FOR THE N-TUPLE                                          J 241
C     NNN IS THE CLASSIFICATION BY BAYES' RULE                          J 242
C     LJ IS POSITION IN RESULT ARRAY FOR (NN,NNN)                       J 243
C                                   ***                                 J 244
      DO 24 I=1,NN                                                      J 245  195
      II=20001-I*2                                                      J 246  196
      N=IPT(II)/10000                                                   J 247  197
      MM=(IPT(II)-N*10000)/100                                          J 248  198
      MMM=(IPT(II)-N*10000-MM*100                                       J 249  199
      LJ=(NN-1)*NM+MMM                                                  J 250  200
C                                   ***                                 J 251
C     IF DATA WAS STORED GO FETCH IT                                    J 252
C                                   ***                                 J 253
   24 RESULT(LJ)=RESULT(LJ)+FLOAT(N)                                    J 254  208
      IF (NBC.GT.0) GO TO 30                                            J 255  211
      SUM=0.                                                            J 256  212
C                                   ***                                 J 257
C     CALCULATE PROBABILITIES AND PRINT OUT RESULTS                     J 258
C                                   ***                                 J 259
      NP1=N+1                                                           J 260  215
      MM1=M+1                                                           J 261  216
      DO 25 I=1,MM1,NP1                                                 J 262  217
   25 SUM=SUM+RESULT(I)                                                 J 263  218
C                                   ***                                 J 264
C     SUM IS TOTAL PROB OF CORRECT CLASSIFICATIONS TO CONDITIONAL PROB  J 265
C                                   ***                                 J 266
      SUM=SUM/SP                                                        J 267  220
      DO 26 I=1,M                                                       J 268  221
      LJ=(I-1)*NP1                                                      J 269  222
      RSUM=0.                                                           J 270  223
      DO 26 J=LJ,M                                                      J 271  224
   26 RESULT(J)=RESULT(J)/RSUM                                          J 272  225
      CALL OUTP (RESULT,M)                                              J 273  226
      WRITE (8,FMT50) SUM                                               J 274  229
C                                   ***                                 J 275
C     WRITE END OF FILE MARKS ON TAPES                                  J 276
C                                   ***                                 J 277
      I=0                                                               J 278  232
      WRITE (10) I                                                      J 279  233
      WRITE (20) I                                                      J 280  234
      REWIND 10                                                         J 281  235
      CALL CHMAT                                                        J 282  240
C                                   ***                                 J 283
C     STORE RESULTS AND INITIALIZE NCOUNT                               J 284
C     NCOUNT IS NOW THE NUMBER OF LOCATIONS ACTUALLY FILLED             J 285
C     NCOUNT IS THE BOTTOM MOST POSITION FILLED IN THE TOP PART OF ZZZ  J 286
C     BUMP TYP MATH OF ZZZ BUTP Z                                       J 287
C     NBC GIVES NUMBER OF BUMPS IN Q                                    J 288
C                                   ***                                 J 289
   27 NCOUN=(NCOUN-1)*2                                                 J 290  241
      NCOUN=20001-NCOUN                                                 J 291  242
      WRITE (2) NCOUN                                                   J 292  243
      WRITE (2) ((IPT(I),I=N+1,I1),I1=NCOUN,20000,2)                    J 293  246
      NCOUN=1                                                           J 294  251
      LICK=1                                                            J 295  252
      NBC=NBC+1                                                         J 296  253
      GO TO 9                                                           J 297  254
   30 L=20001-NCOUN                                                     J 298  255
      NCOUN=0                                                           J 299  256
      I=20001-L*2                                                       J 300  257
      IP1=1                                                             J 301  258
      DO 20 J=NN,L                                                      J 302  259
      I=I+1                                                             J 303  260
   20 IPT(I)=IPT(J)                                                     J 304  261
      JOLTAPE                                                           J 305  263
      LTAPE=LTAPE                                                       J 306  264
      LTAPE=J                                                           J 307  265
      ABC=ABC+1                                                         J 308  266
      NNN=NNN+1                                                         J 309  267
      GO TO 9                                                           J 310  268
C                                   ***                                 J 311
C     STORE, PUNCH AND INITIALIZE N-TUPLES                              J 312
C                                   ***                                 J 313
   40 NCOUN=NCOUN+1                                                     J 314  269
      WRITE (20) NCOUN                                                  J 315  270
      REWIND 9                                                          J 316  271
      WRITE (20) (MORE(I),I=1,NCOUN)                                    J 317  272
      DO 31 I=1,NCOUN                                                   J 318  273
      READ (9) N1,NPRED                                                 J 319  280
      WRITE (43,36) MORE(I),N1,NPRED                                    J 320  284
   31 NBC=NBC+N1                                                        J 321  287
      WRITE (20) (MORE(I),I=1,NCOUN)                                    J 322  289
      REWIND 9                                                          J 323  294
      N1=0                                                              J 324  295
      NCOUN=1                                                           J 325  296
      NCOUN=NCOUN+1                                                     J 326  297
      GO TO 9                                                           J 327  298
C                                   ***                                 J 328
C                                   ***                                 J 329
   50 FORMAT (22HNO.STATIONS CLASSIFICATIONS,5F6.1,5X )                 J 330  300
   35 FORMAT (4X,I4,2X)                                                 J 331  301
   36 FORMAT (2I4,20F6.0, 7HTOTAL NEGATIVE,I4,10HNO.LEVELS .I4,10HNO. O J 332  302
     1NP2)                                                              J 333
   55 FORMAT (F10.0,I10,I5,I5)                                          J 334  303
   60 FORMAT (8I10,I4,I4,F10.5)                                         J 335  304
      END                                                               J 336- 305
```

```
 1          SUBROUTINE OUTP (RESULT,N)                                      *   1
 2          DIMENSION RESULT(N,N), IDL(25)                                  *   2
 3          COMMON HP31,FORMT1(12)                                          *   3
 4          COMMON F                                                        *   4
 5          COMMON FMT1(19),FMT2(5),FMT9(5),FMT11(8),FMT12(7),FMT13(3),FMT14(2  *   5
 6         11),FMT1A(2),FMT22(3),FMT5A(8),FMT57(4),FMT58(8)                 *   6
 7          COMMON L8,L9,L10,L11,L12,L13,L14                                *   7
 8          INTEGER F                                                       *   8
 9          DATA IDL/1HT,1HR,1HU,1HE,1H ,1HT,1HR,1HA,1HI,1HN,1HG,1H ,1  *   9
10         1HR,1HE,1HG,1HI,1HO,1HN/                                         *  10
11    C                                  ***                                *  11
12    C     PRINT OUT PROBABILITY MATRIX IN PUBLISHABLE FASHION             *  12
13    C                                  ***                                *  13
14          II=1                                                            *  14
15          III=8                                                           *  15    2
16          NN=N                                                            *  16    3
17        1 CALL NNAME                                                      *  17    4
18          N=NN                                                            *  18    5
19          IF (N.LT.8) GO TO 9                                             *  19    6
20          LINE=0                                                          *  20    9
21          N=8                                                             *  21   10
22          NS=25-N                                                         *  22   11
23          IF (NS.EQ.0) GO TO 3                                            *  23   12
24          DO 2 I=1,NS                                                     *  24   15
25          LINE=LINE+1                                                     *  25   16
26        2 WRITE (6,15)                                                    *  26   17
27        3 LINE=LINE+7                                                     *  27   20
28          WRITE (6,FMT56)                                                 *  28   21
29          WRITE (6,FMT12)                                                 *  29   23
30          WRITE (6,FMT13) (I,I=II,III)                                    *  30   25
31          IF (LINE.NE.24) GO TO 4                                         *  31   30
32          WRITE (6,FMT14) IDL(1)                                          *  32   33
33        4 DO 6 J=1,N                                                      *  33   34
34          LINE=LINE+1                                                     *  34   37
35          WRITE (6,FMT57) J,(RESULT(I,J),I=II,I                           *  35   38
36          IF (44-LINE.LE.0) GO TO 5                                       *  36   44
37          L=LINE-23                                                       *  37   47
38          IF (L.LE.0) GO TO 5                                             *  38   48
39          WRITE (6,FMT14) IDL(L)                                          *  39   51
40        5 LINE=LINE+1                                                     *  40   54
41          WRITE (6,15)                                                    *  41   55
42          IF (44-LINE.LT.0) GO TO 6                                       *  42   57
43          L=LINE-23                                                       *  43   60
44          IF (L.LE.0) GO TO 6                                             *  44   61
45          WRITE (6,FMT14) IDL(L)                                          *  45   64
46        6 CONTINUE                                                        *  46   67
47        7 LINE=LINE+1                                                     *  47   69
48          L=LINE-23                                                       *  48   70
49          IF (L.GT.25) GO TO 8                                            *  49   71
50          WRITE (6,FMT14) IDL(L)                                          *  50   74
51          GO TO 7                                                         *  51   77
52        8 N=N-8                                                           *  52   78
53          II=II+8                                                         *  53   79
54          III=III+8                                                       *  54   80
55          IF (N.GT.0) GO TO 1                                             *  55   81
56          RETURN                                                          *  56   84
57        9 DO 10 I=1,23                                                    *  57   85
58       10 WRITE (6,15)                                                    *  58   86
59          WRITE (6,FMT56)                                                 *  59   89
60          WRITE (6,FMT12)                                                 *  60   91
61          III=III+N-1                                                     *  61   93
62          WRITE (6,FMT13) (I,I=II,III)                                    *  62   94
63          WRITE (6,FMT14) IDL(1)                                          *  63   99
64          LINE=24                                                         *  64  102
65          DO 12 J=1,N                                                     *  65  103
66          LINE=LINE+1                                                     *  66  104
67          WRITE (6,FMT57) J,(RESULT(I,J),I=II,III)                        *  67  105
68          IF (44-LINE.LE.0) GO TO 11                                      *  68  111
69          L=LINE-23                                                       *  69  114
70          IF (L.LE.0) GO TO 11                                            *  70  115
71          WRITE (6,FMT14) IDL(L)                                          *  71  118
72       11 LINE=LINE+1                                                     *  72  121
73          WRITE (6,15)                                                    *  73  122
74          IF (44-LINE.LE.0) GO TO 12                                      *  74  124
75          L=LINE-23                                                       *  75  127
76          IF (L.LE.0) GO TO 12                                            *  76  128
77          WRITE (6,FMT14) IDL(L)                                          *  77  131
78       12 CONTINUE                                                        *  78  134
79       13 LINE=LINE+1                                                     *  79  136
80          L=LINE-23                                                       *  80  137
81          IF (L.GT.25) GO TO 14                                           *  81  138
82          WRITE (6,FMT14) IDL(L)                                          *  82  141
83          GO TO 13                                                        *  83  144
84       14 RETURN                                                          *  84  149
85    C                                  ***                                *  85
86    C                                                                     *  86
87    C                                  ***                                *  87
88       19 FORMAT (/)                                                      *  88  148
89          END                                                             *  89- 148
```

```
<<
```

```
17564 03  05-02-A9   18.316

 1            SUBROUTINE DECSON (I,J,M1,SIGMA,K,N,A,LT,NTOTAL,XPNOM)
 2            DIMENSION M1(2), SIGMA(2)
 3            DIMENSION A(2), C(30)
 4            LOGICAL LT
 5            M=0
 6            DO 1 K=1,N
 7          1 C(K)=0.
 8            XPNOB=0.
 9            ZLARGE=.1E+21
10            DO 7 K1=I,J
11            K=20001->*K1
12            LL=M1(K)/10000
13            L=(M1(K)-LL*10000)/100
14          7 C(L)=C(L)+FLOAT(LL)
15            XPROB=FLOAT(M)/FLOAT(NTOTAL)
16            M=1
17      C                          ***
18      C     IF NORMALIZED IS SPECIFIED MULT BY NORMALIZING FACTORS
19      C                          ***
20            IF (.NOT.LT) GO TO 4
21            DO 3 L=1,N
22            IF (C(L).LT.1.E-10) C(L)=0.
23          3 C(L)=C(L)*SIGMA(L)
24          4 DO 6 L=1,N
25            SMALL=0.
26      C                          ***
27      C     A IS THE LOSS MATRIX
28      C                          ***
29            DO 5 K=1,N
30            SMALL=SMALL+A(M)*C(K)
31          5 M=M+1
32            SMALL=SMALL-10000.
33      C                          ***
34      C     LOOK FOR PREDOMINT [illegible] REGION
35      C
36            IF (SMALL.GT.ZLA[illegible]
37            ZLARGE=SMALL
38            K=L
39          6 CONTINUE
40      C                          ***
41      C     K IS THE RESULTING TRAINING REGION
42      C                          ***
43            RETURN
44            END

<<
```

```
      1          SUBROUTINE DMP                                          =   1
      2          DIMENSION IA(20), IB(20), IC(20)                        =   2
      3          COMMON MP31,FORMT1(12)                                  =   3
      4          COMMON F                                                =   4
      5          COMMON IQUMY(752),IZZ(240000)                           =   5
      6          EQUIVALENCE (IA(1),IZZ(1)), (IB(1),IZZ(1200))           =   6
      7          INTEGER F                                               =   7
      8          REWIND 9                                                =   8
      9          REWIND 2                                                =   9     2
     10          REWIND 10                                               =  10     3
     11          ITAPE=10                                                =  11     4
     12          KTAPE=9                                                 =  12     5
     13          READ (10) N                                             =  13     6
     14          IF (N.GT.20) GO TO 19                                   =  14     7
     15        1 READ (ITAPE) I                                          =  15    12
     16          IF (I.GT.120000) GO TO 19                               =  16    15
     17          IF (I.LE.0) GO TO 8                                     =  17    18
     18          ITEST=I                                                 =  18    21
     19          READ (ITAPE) (IZZ(J),J=1,I)                             =  19    22
     20        2 READ (ITAPE) M                                          =  20    27
     21          IMP1=I+1                                                   21    30
     22          L=I+M                                                   =  22    31
     23          IF (M.LE.0) GO TO 3                                     =  23    32
     24          READ (ITAPE) (IZZ(J),J=IMP1,L)                          =  24    35
     25        3 L=L-1                                                   =  25    40
     26          DO 4 II=2,L=1,2                                         =  26    41
     27          IP1=II+2                                                =  27    42
     28          DO 4 JJ=IP1,L,2                                         =  28    43
     29          IF (IZZ(II).GT.IZZ(JJ)) GO TO 4                         =  29    44
     30          IS=IZZ(II-1)                                            =  30    47
     31          IZZ(II-1)=IZZ(JJ-1)                                     =  31    48
     32          IZZ(JJ-1)=IS                                            =  32    49
     33          IS=IZZ(II)                                              =  33    50
     34          IZZ(II)=IZZ(JJ)                                         =  34    51
     35          IZZ(JJ)=IS                                              =  35    52
     36        4 CONTINUE                                                =  36    53
     37          WRITE (KTAPE) M                                         =  37    56
     38          IF (M.LE.0) GO TO 5                                     =  38    59
     39          WRITE (KTAPE) (IZZ(J),J=IMP1,L)                         =  39    62
     40          GO TO 2                                                 =  40    67
     41        5 WRITE (2) I                                             =  41    68
     42          WRITE (2) (IZZ(J),J=1,I)                                =  42    71
     43          KK=KTAPE                                                =  43    76
     44          KTAPE=ITAPE                                             =  44    77
     45          ITAPE=KK                                                =  45    78
     46          REWIND 9                                                =  46    79
     47          REWIND 10                                               =  47    80
     48          GO TO 1                                                 =  48    81
     49        6 M1=0                                                    =  49    82
     50          WRITE (2) M1                                            =  50    83
     51          REWIND 2                                                =  51    86
     52          WRITE (5,16)                                            =  52    87
     53          WRITE (6,17) (I,I=1,N)                                  =  53    89
     54        7 DO 8 I=1,20                                             =  54    94
     55        8 IC(I)=0                                                 =  55    95
     56          READ (2) M                                              =  56    97
     57          IF (M.LE.0) GO TO 12                                    =  57   100
     58          READ (2) (IZZ(I),I=1,M)                                 =  58   103
     59          IMP=IZZ(2)                                              =  59   108
     60          M1=IZZ(1)/10000                                         =  60   109
     61          M2=(IZZ(1)-10000*M1)/100                                =  61   110
     62          IC(M2)=M1                                               =  62   111
     63          DO 11 I=3,M,2                                           =  63   112
     64          IF (IMP.NE.IZZ(I+1)) GO TO 9                            =  64   113
     65          M1=IZZ(I)/10000                                         =  65   116
     66          M2=(IZZ(I)-10000*M1)/100                                =  66   117
     67          IC(M2)=M1+IC(M2)                                        =  67   118
     68          GO TO 11                                                =  68   119
     69        9 WRITE (6,18) IMP,(IC(J),J=1,N)                          =  69   120
     70          IMP=IZZ(I+1)                                            =  70   126
     71          DO 10 J=1,20                                            =  71   127
     72       10 IC(J)=0                                                 =  72   128
     73          M1=IZZ(I)/10000                                         =  73   130
     74          M2=(IZZ(I)-10000*M1)/100                                =  74   131
     75          IC(M2)=M1                                               =  75   132
     76       11 CONTINUE                                                =  76   133
     77          WRITE (6,18) IMP,(IC(J),J=1,N)                          =  77   135
     78          GO TO 7                                                 =  78   141
     79       12 WRITE(IQUMY(742))                                       =  79   142
     80          REWIND 20                                               =  80   143
     81       13 READ (20) N                                             =  81   144
     82          IF (N.LE.0) GO TO 14                                    =  82   147
     83          READ (20) (IA(J),J=1,N)                                 =  83   150
     84          READ (20) (IB(J),J=1,N)                                 =  84   155
     85          WRITE (6,19) (IA(J),IB(J),J=1,N)                        =  85   160
     86          GO TO 13                                                =  86   165
     87       14 CALL CHAIN                                              =  87   166
     88       15 WRITE (6,20)                                            =  88   167
     89          CALL CHAIN                                              =  89   169
     90          RETURN                                                  =  90   170
     91    C                          ***                                =  91
     92    C                                                             =  92
     93    C                          ***                                =  93
     94       16 FORMAT (1H1,10X,*MPLES,***,15*TRAINING REGIONS//)       =  94   171
     95       17 FORMAT (/,8X,20I9)                                      =  95   171
     96       18 FORMAT (5X,I10,5X,20I10,I4/)                            =  96   171
     97       19 FORMAT(9I10,I10,10X,I2))                                     171
     98       20 FORMAT (*NO TOO *** TRAINING REGIONS FOR CONVIENT OUTPUT SUBCUTIN  =  98   171
     99          1H CONTINUED     )                                      =  99
    100          END                                                     = 100-  171
```

```
<><><>
```

```
      SUBROUTINE PLOTER                                                 *  1
      DIMENSION CPS(20,130), PLOT(130), DATA(36), APT(10000), RLAS(10000),  *  2
     1 EXT(1000)                                                        *  3
      DIMENSION PLT(130)                                                *  4
      DIMENSION AAA(25), WRDINT(36)                                     *  5
      COMMON /ISTT/ IST                                                 *  6
      COMMON FORT1,FORMT(12)                                            *  7
      COMMON F                                                          *  8
      COMMON FMT1(19),FMT2(3),FMT4(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2  *  9
     1),FMT15(2),FMT22(3),FMT56(6),FMT57(6),FMT58(8)                    *  10
      COMMON LA,L9,L10,L11,L12,L13,L14                                  *  11
      COMMON NP,NN,LEVEL,NA1,NPOINT,ITAPE,AAA,L1,L2,L3,L4,L5,L6,NA,NB   *  12
     1,NC,ND,L7,NI,NY,ATEN,ZPZ(2400)                                    *  13
      EQUIVALENCE (RLAS(1),EXT(1)),(ZBZ(1),APT(1)),(ZBZ(10001),RLAS(1))  *  14
     1 , (ZPZ(23001),CPS(1,1))                                          *  15
      LOGICAL L1,L2,L3,L4,L5,L6,L7                                      *  16
      LOGICAL L9,L9,L10,L11,L12,L13                                     *  17
      INTEGER F                                                         *  18
      INTEGER CPS                                                       *  19
      INTEGER PLT                                                       *  20
      DATA BLANK/1H /                                                   *  21
C      NP IS NUMBER OF PHOTOS
C      NN IS NUMBER OF DATA POINTS PER PHOTO
C      PLOT ARRAY OUTPUTS PHOTO CLASSIFICATION LINE BY LINE
C      NO MORE THAN 20 PHOTOS CAN BE HANDLED HERE
      RESIND 3                                                          *  22
      IST=3543398793                                                    *  23    2
      RESIND 4                                                          *  24    3
      RESIND 20                                                         *  25    4
      IF (L9) WRITE (43,12)                                             *  26    5
      CALL HEADF                                                        *  27    6
      J1=1                                                              *  28   10
      J2=0                                                              *  29   11
    1 READ (20) NCOUNT                                                  *  30   12
      IF (NCOUNT.LE.0) GO TO 2                                          *  31   13
      J2=J2+NCOUNT                                                      *  32   14
C      APT IS THE COMPACTED POINT ARRAY
C      RLAS IS THE CATEGORY WHICH APT IS PUT IN
      READ (20) (APT(J),J=J1,J2)                                        *  33   15
      READ (20) (RLAS(J),J=J1,J2)                                       *  34   16
      J1=J2+1                                                           *  35   17
      GO TO 1                                                           *  36   18
    2 WRITE (6,FMT11)                                                   *  37   19
      J3=(130-NN)/2+1                                                   *  38   20
      J4=J3+79                                                          *  39   21
      RESIND 2                                                          *  40   22
      DO 7 I=1,NY                                                       *  41   23
C      READ A LINE OF UNCOMPACTED N-TUPLES
      DO 3 J=1,NN                                                       *  42   24
      IF (L1) READ (3) (CPS(N,J),N=1,NP)                                *  43   25
    3 IF (.NOT.L1) READ (4) (CPS(N,J),N=1,NP)                           *  44   26
      DO 4 J=1,130                                                      *  45   27
    4 PLOT(J)=BLANK                                                     *  46   28
      JJ=(130-NN)/2                                                     *  47   29
      DO 6 J=1,NN                                                       *  48   30
C      FOR EACH N-TUPLE IN LINE COMPACT IT AND FIND OUT HOW TO CLASSIFY
      JJ=JJ+1                                                           *  49   31
      NK=0                                                              *  50   32
      DO 5 N=1,NP                                                       *  51   33
    5 NK=NK+LEVEL*CPS(N,J)                                              *  52   34
      CALL SEARCH (APT,RLAS,NN,SYMBOL,J2,NP,N,LEVEL)                    *       35
C      PLT CONTAINS NUMERIC CODE FOR SYMBOL
      PLT(J)=N                                                          *  54   36
    6 PLOT(JJ)=SYMBOL                                                   *  55   37
      WRITE (2) (PLT(J),J=1,NN)                                         *  56   38
C      NOW WRITE OUT THE LINE
      IF (L9) WRITE (43,13) (PLOT(J),J=J3,J4)                           *  57   39
      IF (.NOT.L11) WRITE (6,14) (PLOT(J),J=31,100)                     *  58   40
    7 IF (L11) WRITE (6,14) (PLOT(J),J=1,130)                           *  59   41
      ATEN=0                                                            *  60   42
      IF (L2.AND.L9) GO TO 11                                           *  61   43
C      COMPUT. CONTINGENCY TABLE HERE
      RESIND 1                                                          *  62   44
      RESIND 2                                                          *  63   45
      DO 8 I=1,NA                                                       *  64   46
      ZPZ(I)=0.                                                         *  65   47
    8 READ (1) (ZBZ(J),J=1,NN)                                          *  66   48
C      READ IN TRAINING REGION
      NSAMPLE                                                                 49
      I=NA+1                                                                  50
      NTRAIN=NN-NA                                                            51
      READ(1) (APT(J),J=1,NTRAIN)                                             52
      DO 9 I=1,NB                                                       *  69   53
    9 ZBZ(I)=0.                                                         *  70   54
      JJ=NB                                                                   55
C      FOR EACH LINE OF OUTPUT
      DO 10 I=1,NY                                                      *  72   56
C      READ ALL THE POINTS ON THE LINE
      READ (2) (PLT(J),J=1,NN)                                          *  73   57
C      FOR EACH POINT ON THE LINE
      DO 10 J=1,NN                                                      *  74   58
      JJ=JJ+1                                                           *  75   59
      NK=(APT(JJ)-1)*NA+PLT(J)                                                60
    10 ZBZ(NK)=ZBZ(NK)+1.                                               *  77   61
      CALL OUTP (ZBZ,NA)                                                *  78   62
    11 CALL CHRST                                                       *  79   63
      STOP                                                              *  80   64
C                              ***                                      *  81
C                                                                       *  82
C                              ***                                      *  83
   12 FORMAT (25HPHOTO CLASSIFICATION)                                  *  84   65
   13 FORMAT (80A1)                                                     *  85   66
   14 FORMAT (130A1)                                                    *  86   67
      END                                                               *  87   68
```

```
*****************************************************
-    23676 WORDS OF MEMORY USED BY THIS COMPILATION
*****************************************************
```

```
      SUBROUTINE OUTP (RESULT,N)
      DIMENSION RESULT(N,N), IDL(20)
      COMMON MP01,FORMT1(12)
      COMMON F
      COMMON FMT1(15),FMT2(3),FMT9(5),FMT11(6),FMT12(7),FMT13(3),FMT14(2
     1),FMT16(2),FMT22(3),FMT56(6),FMT57(4),FMT58(8)
      COMMON LP,L9,L10,L11,L12,L13,L14
      INTEGER F
      DATA IDL/1HT,1HR,1HU,1HE,1H ,1HT,1HR,1HA,1HI,1HN,1HI,1HN,1HG,1H ,1
     1HR,1HE,1HG,1HI,1HO,1HN/
C                                   ***
C     PRINT OUT PROBABILITY MATRIX IN PUBLISHABLE FASHION
C                                   ***
      II=1
      III=8
      NN=N
    1 CALL NNAME
      K=NN
      IF (K.LT.8) GO TO 9
      LINE=0
      K=8
      NS=25-K
      IF (NS.EQ.0) GO TO 3
      DO 2 I=1,NS
      LINE=LINE+1
    2 WRITE (6,15)
    3 LINE=LINE+7
      WRITE (6,16)
      WRITE (6,FMT12)
      WRITE (6,FMT13) (I,I=II,III)
      IF (LINE.NE.24) GO TO 4
      WRITE (6,FMT14) IDL(1)
    4 DO 6 J=1,N
      LINE=LINE+1
      WRITE (6,FMT57) J,(RESULT(I,J),I=II,III)
      IF (44-LINE.LE.0) GO TO 5
      L=LINE-23
      IF (L.LE.0) GO TO 5
      WRITE (6,FMT14) IDL(L)
    5 LINE=LINE+1
      WRITE (6,15)
      IF (44-LINE.LT.0) GO TO 6
      L=LINE-23
      IF (L.LE.0) GO TO 6
      WRITE (6,FMT14) IDL(L)
    6 CONTINUE
    7 LINE=LINE+1
      L=LINE-23
      IF (L.GT.20) GO TO 8
      WRITE (6,FMT16) IDL(L)
      GO TO 7
    8 NN=NN-K
      II=II+8
      III=III+8
      IF (NN.GT.0) GO TO 1
      RETURN
    9 DO 10 I=1,23
   10 WRITE (6,15)
      WRITE (6,16)
      WRITE (6,FMT12)
      III=II+K-1
      WRITE (6,FMT13) (I,I=II,III)
      WRITE (6,FMT16) IDL(1)
      LINE=24
      DO 12 J=1,N
      LINE=LINE+1
      WRITE (6,FMT57) J,(RESULT(I,J),I=II,III)
      IF (44-LINE.LE.0) GO TO 11
      L=LINE-23
      IF (L.LE.0) GO TO 11
      WRITE (6,FMT14) IDL(L)
   11 LINE=LINE+1
      WRITE (6,15)
      IF (44-LINE.LE.0) GO TO 12
      L=LINE-23
      IF (L.LE.0) GO TO 12
      WRITE (6,FMT14) IDL(L)
   12 CONTINUE
   13 LINE=LINE+1
      L=LINE-23
      IF (L.GT.20) GO TO 14
      WRITE (6,FMT16) IDL(L)
      GO TO 13
   14 RETURN
C                                   ***
C
C                                   ***
   15 FORMAT (/)
   16 FORMAT (57X,17HCONTINGENCY TABLE)
      END

<<
```

```
$       FORTRAN NDECK,COMDK                                              SEARCH
        SUBROUTINE SEARCH (KPT,KLAS,KK,SYMBOL,J2,NP,K,LEVEL)
        DIMENSION KPT(2), KLAS(2), SY(36)                                P   2
        DIMENSION KP(100)
        DIMENSION LIST(2)                                                P   4
        COMMON /ISTT/ IST                                                P   5
        DATA SY/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1H P   6
       10,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H1,1H2,1H3,1H4,1H5, P   7
       21H6,1H7,1H8,1H9,1H0/                                             P   8
C        K IS THE CATEGORY NUMBER
C        SYMBOL IS THE CATEGORY CODE
C        KK IS THE COMPACTED NTUPLE
C        J2 IS UPPER LIMIT FOR KPT AND KLAS ARRAYS
C        LOOK FOR COMPACTED N-TUPLE KK IN LIST
        KS=0                                                             P   9
        KL=J2+1                                                          P  10
        KTRY=(KS+KL)/2                                                   P  11
      1 CONTINUE
        IF(KK-KPT(KTRY)) 3,4,2
      2 IF ((KL-KS).LE.1) GO TO 5                                        P  13
        KL=KTRY                                                          P  14
        KTRY=(KL+KS+1)/2                                                 P  15
        GO TO 1                                                          P  16
      3 IF (KL-KS.LE.1) GO TO 5                                          P  17
        KS=KTRY                                                          P  18
        KTRY=(KL+KS)/2                                                   P  19
        GO TO 1                                                          P  20
      4 K=KLAS(KTRY)                                                     P  21
        SYMBOL=SY(K)                                                     P  22
        RETURN                                                           P  23
      5 CONTINUE                                                         P  24
C        KK CANNOT BE FOUND DO NEAREST NEIGHBOR SEARCH
        I1=IDIST(KK,KPT(1),NP,LEVEL)
        L=1
        KP(1)=1
        DO 7 J=2,J2                                                      P  28
        I2=IDIST(KK,KPT(J),NP,LEVEL)
        IF (I1.LT.I2) GO TO 7                                            P  30
        IF (I1.NE.I2) GO TO 6                                            P  31
        IF(L.GT.99) GO TO 7
        L=L+1                                                            P  32
        KP(L)=J                                                          P  34
        GO TO 7                                                          P  35
      6 CONTINUE                                                         P  36
        I1=I2                                                            P  37
        L=1
        KP(1)=J
      7 CONTINUE                                                         P  40
        X=RCM(IST)                                                       P  44
        K=X*FLOAT(L-1)+1.5
C        KP IS THE ARRAY CONTAINING THE INDEXES FOR ALL THOSE POINTS IN
C        THE KPT ARRAY WHICH ARE CLOSEST TO KK
C        L IS THE UPPER LIMIT OF KP
C        WE WILL CHOOSE ONE POINT FROM THE KP ARRAY AT RANDOM AND
C        IDENTIFY KK WITH THE CATEGORY OF THE POINT IN KPT ASSOCIATED WITH
C        THE RANDOMLY CHOSEN ONE
        K=KP(K)
      8 K=KLAS(K)                                                        P  46
        SYMBOL=SY(K)                                                     P  47
        RETURN                                                           P  48
        END                                                              P  49
```



```
$       FORTRAN NDECK,COMDK                                              IDIST
        FUNCTION IDIST(KG,KP,NP,LEVEL)
        IDIST=0
        KK=KG
        LL=KP
        NP1=NP-1
        DO 1 J=1,NP1
        K=KK/LEVEL
        L1=KK-K*LEVEL
        I=LL/LEVEL
        L2=LL-I*LEVEL
        KK=K
        LL=I
      1 IDIST=IDIST+(L1-L2)*(L1-L2)
        IDIST=IDIST+(K-I)*(K-I)
        RETURN
        END
```

SNUMB = 17564, ACTIVITY # = 04, REPORT CODE = 06, RECORD COUNT = 05082

```
            PHOPTS PATERN DIAGON HLFNMF
            STOPNT PHOUTS
      2      8000        6          2
     28       400
FORMAT FOR TRAINING DATA IS (20A1)
```

TEST DATA

ORIGINAL DATA

INPUT FORMAT FOR DATA WAS (20I1)
OUTPUT FORMAT FOR DATA IS (1X,20I1)

PHOTO 1

```
0352431410
0352431410
0352431410
```

PHOTO									
1									
0	3	5	2	4	3	1	4	1	0

PHOTO									
0	3	5	2	4	3	1	4	1	0

TEST DATA

LABLE TABLE

PLOT SYMBOLS	VS INPUT SYMBOLS
A	A
B	B

TEST DATA

TRAINING REGIONS

```
ABBABBBABA
ABBABBBABA
ABBABBBABA
```

TEST DATA

NO. TRAINING REGIONS	NORMALIZED NO.LEVELS	NO. PHOTOS
2	5	2

TEST DATA

THE PROBABILITY MATRIX

TRAINING REGION CLASSIFIED AS

TRUE TRAINING REGION	1	2
1	0.4545E 00	0.5455E 00
2	0.1111E 00	0.8889E 00

TOTAL PROBABILITY

0.6500E 00

N-TUPLES

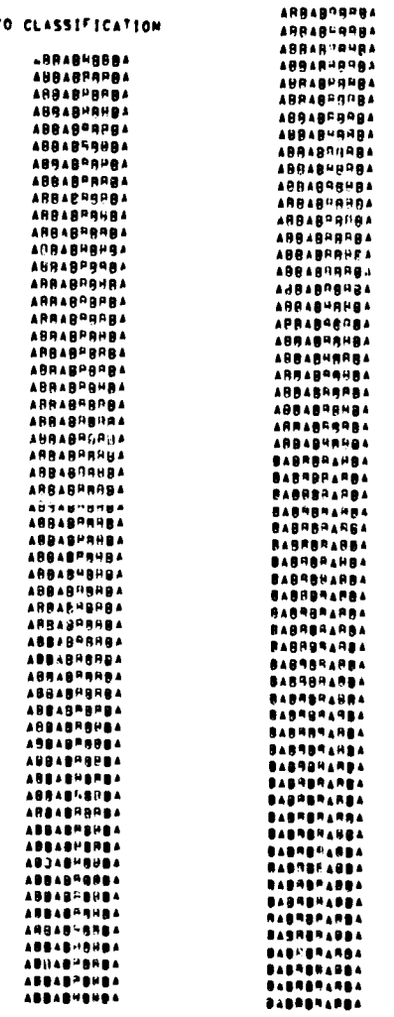
TRAINING REGION

	1	2
35	0	400
28	400	400
21	400	400
14	600	200
7	400	400
0	400	0

35 2 28 2 21 2 14 1 7 2
0 1

TEST DATA

PHOTO CLASSIFICATION



TEST DATA

CONTINGENCY TABLE

TRAINING REGION CLASSIFIED AS

TRUE REGION	1	2
1	0.1000E 04	0.1200E 04
2	0.2000E 03	0.1600E 04

SECOND HALF, NUMBER OF POINTS IS 4000

PHOTO 1

1403552510	1403552510
1403552510	1403552510
1403552510	1403552510

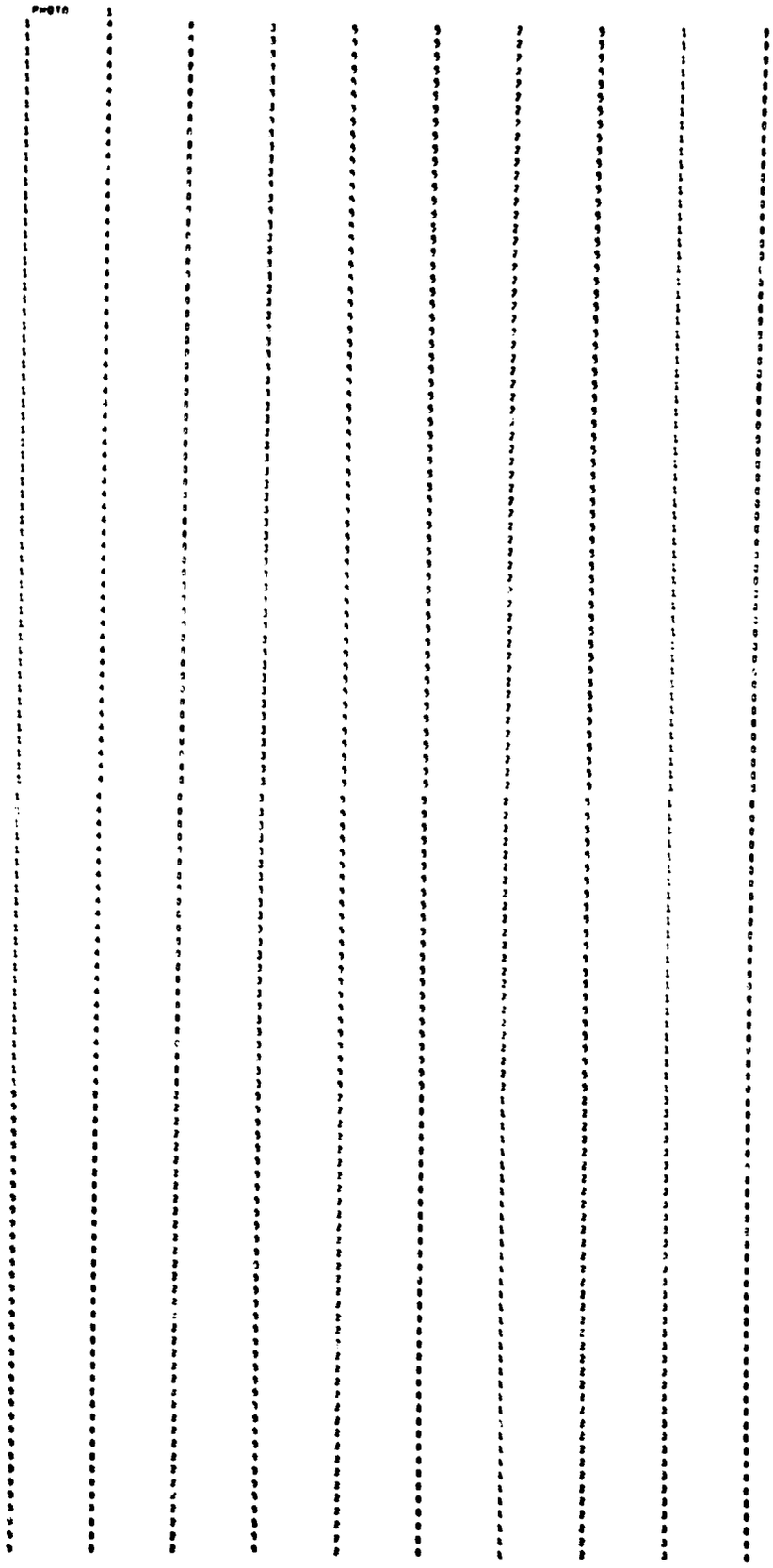


PHOTO	2								
1	4	0	3	5	5	2	5	1	0

TEST DATA

LABLE TABLE

PLOT SYMBOLS	VS INPUT SYMBOLS
A	A
B	B

TEST DATA

TRAINING REGIONS

ABAABBABAA
ABAABBABAA
ABAABBABAA

TEST DATA

PHOTO CLASSIFICATION

```
BBARBBAHBA
BBARBBARBA
BBARBBARBA
BBARBHARBA
BHARBBARBA
BBARBBARBA
BBARBBARBA
BBAHBBARBA
BBARBBAHBA
BHARBHARBA
BBARBBARBA
BBARBBARBA
BBARBB
```

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Center for Research, Inc. The University of Kansas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

The Bayesian Approach to Identification of a Remotely Sensed Environment

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Robert M. Haralick

6. REPORT DATE July 1969	7a. TOTAL NO. OF PAGES 68	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. DAAK02-68-C-0089 b. PROJECT NO. ARPA Order No. 1079 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 133-9 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* Supported	

10. DISTRIBUTION STATEMENT

No Limitation

11. SUPPLEMENTARY NOTES Funded by DOD Project Themis under ARPA Order No. 1079	12. SPONSORING MILITARY ACTIVITY Monitoring: U.S. Army Engineer Topographic Labs Geographic Information Systems Branch Geographic System Division Ft. Belvoir, Virginia

13. ABSTRACT

The first part of this paper provides a brief tutorial introduction of the Bayesian Approach to identification of a remotely sensed environment. The second part describes the input data deck setup for the Fortran IV program which has been written to implement this approach. The third part describes file usage and subroutine organization. The fourth part provides a listing of the program with a simple sample data set.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

CRES LABORATORIES

Chemical Engineering Low Temperature Laboratory

Remote Sensing Laboratory

Electronics Research Laboratory

Chemical Engineering Heat Transfer Laboratory

Nuclear Engineering Laboratory

Environmental Health Engineering Laboratory

Digital Computer Technology Laboratory

Water Resources Institute