

INSTYTUT AUTOMATYKI
POLSKIEJ AKADEMII NAUK



PRACE
INSTYTUTU AUTOMATYKI PAN

Zeszyt 35

JACEK KARPIŃSKI, RYSZARD MICHAŁSKI

PERCEPTRON DLA ZNAKÓW ALFANUMERYCZNYCH

OPIS KONCEPCJI PROJEKTU
ORAZ MODELU NA MASZYNIE CYFROWEJ



WARSZAWA

1966

Н. Карпински Р. Михальски

Перцептрон для ознакования букв и цифр
концепция проекта и описание модели перцептрона
на вычислительной машине

Резюме

Приводится описание устройства опознающего буквы и цифры. Устройство состоит из двух частей :

1. - рецепторной части, определяющей некоторые геометрические свойства знаков,
2. - ассоциирующей части, которая обучается и опознает знаки методом вычисления определенного коэффициента сходства опознаваемого знака с эталонным образом.

Описаны результаты экспериментов проведенных с моделью устройства на вычислительной машине.

J. Karpiński, R. Michalski

Perceptron recognizing alphanumeric characters. The idea of a construction and a description of the model realized using a digital computer

Summary

The paper describes a certain idea of the system recognizing alphanumeric characters. The receptive block of this system determines certain geometric properties of characters the categorizer block is a learning system which recognizes characters by computation of an adequately defined coefficient of proximity of a sequence representing character to be recognized to standard sequences. The latter ones are determined in the learning period. A description of the realization of this system using a digital computer and obtained results are also presented.

Na prawach rękopisu

Jacek Karpiński, Ryszard Michalski

PERCEPTRON DLA ZNAKÓW ALFANUMERYCZNYCH OPIS KONCEPCJI PROJEKTU ORAZ MODELU NA MASZYNIE CYFROWEJ

Podano opis pewnej koncepcji urządzenia rozpoznającego dla symboli alfanumerycznych, którego część recepcyjną służy do określania pewnych wybranych własności symboli, a układ asocjacyjny (klasyfikator) stanowi układ uszający się i rozpoznający symbole przez obliczenie odpowiednio określonego współczynnika bliskości (miary podobieństwa) symbolu rozpoznawanego do poszczególnych symboli wzorcowych, określonych w okresie uczenia. Ponadto podano opis programu i wyniki badań modelu urządzenia na maszynie cyfrowej.

1. UWAGI WSTĘPNE

Ostatnio obserwuje się wzmożone zainteresowanie problemami teorii i techniki rozpoznawania. Wynika to z tego, że dalszy rozwój automatyzacji wielu dziedzin wymaga zastosowania nowego typu urządzeń - urządzeń rozpoznających (perceptronów), których zadaniem, ogólnie biorąc, jest podział zadanych obiektów rozpoznawania na określone klasy. Obiektami rozpoznawania mogą być znaki pisma, przedmioty fizyczne, dźwięki mowy, gatunki zwierząt, stany chorobowe organizmów żywych, stany procesów przemysłowych przekroju warstw geologicznych itp.

Znane dotychczas rozwiązania techniczne perceptronów (perceptron Rosenblatta, perceptron "Alfa", "Pandemonium Selfridge'a", perceptron Mark I, kibertron K-100 i inne), modele na maszynach cyfrowych (np. Adapt II Roberta) oraz liczne prace teoretyczne nie wskazują jednak dotychczas na możliwość zbudowania prostego technicznie, niezawodnego i jednocześnie uniwersalnego (przy zmianie urządzenia wejściowego) urządzenia rozpoznającego. Zagadnienie mo-

ina jednak dość znacznie uprościć, jeśli założy się zbiór obiektów rozpoznawania i narzuci ograniczenia co do zakresu ich przekształceń izomorficznych¹⁾. Trudność bowiem polega na zbudowaniu układu recepcyjnego (receptora), którego zadaniem jest pomiar informacji wejściowej o obiekcie i przekształcenie jej w zbiór sygnałów²⁾ (nazwanych często cechami obiektu) możliwie mało zależnych od przekształceń izomorficznych obiektów. Wydaje się, że najbardziej perspektywiczna i uniwersalna jest tu koncepcja budowy odpowiednich układów samoorganizujących się, które same będą określać poszukiwane istotne własności (cechy klas obiektów), przy czym proces przetwarzania informacji może być nawet bliżej nieznanym konstruktorowi. Prostota jednak do realizacji technicznej, przy założonym zbiorze obiektów rozpoznawania, wydaje się koncepcja wymagająca, że by poszukiwaniem cech obiektów zajął się sam człowiek i następnie budował układy, które w sposób ściśle zdeterminowany określają wybrany przez niego zbiór cech.

W niniejszej pracy podano opis pewnej koncepcji budowy urządzenia rozpoznającego dla symboli alfanumerycznych, którego część recepcyjna jest ściśle zdeterminowana, a układ asocjacyjny (klasyfikator) stanowi układ uczący się i rozpoznający nowe symbole przez obliczenie odpowiednio zdefiniowanego współczynnika bliskości (miary podobieństwa) symbolu rozpoznawanego do symbolu wzorcowego określonego w okresie nauczania. Podano ponadto opis programu i wyniki badań modelu urządzenia rozpoznającego na maszynie cyfrowej.

Zasadniczym założeniem, które decydowało o wyborze koncepcji i rozwiązaniu całości układu, było wymaganie jak najprostszej realizacji technicznej na podstawie półprzewodnikowej techniki progowej. Ze względów praktycznych przyjęto stosunkowo gruboziarnistą siatkówkę (8×5), ogólna jednak koncepcja układu (a także program modelu na maszynie cyfrowej) umożliwia stosunkowo łatwe przejście

1)

Przez przekształcenie izomorficzne obiektu rozumie się takie jego przekształcenie, które według subiektywnej oceny człowieka lub pewnych określonych kryteriów, np. technicznych, nie powoduje wyjścia obiektu z określonej klasy.

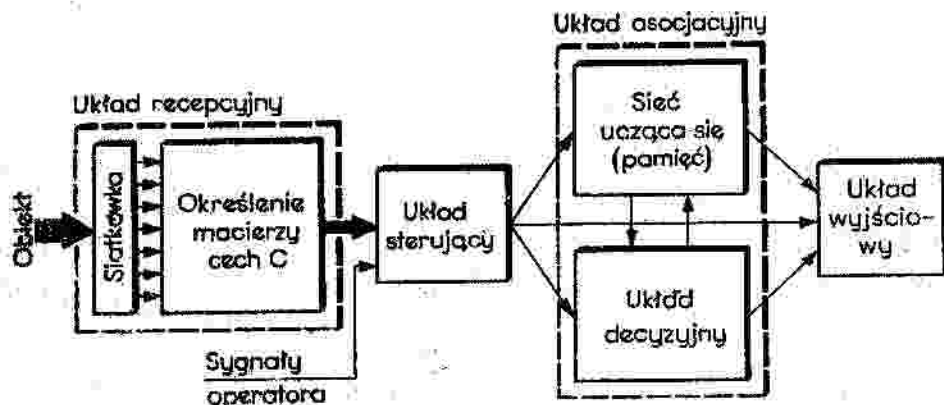
2) Zbiór tego rodzaju sygnałów nazywa się w niniejszej pracy obrazem badanym obiektu w odróżnieniu od obrazu fizycznego powstającego na tzw. siatkówce perceptronu.

na siatkówkę o większej liczbie elementów.

2. OGÓLNY OPIS KONCEPCJI URZĄDZENIA ROZPOZNAJĄCEGO ORAZ PRZYJĘTE ZAŁOŻENIA POCZĄTKOWE

2.1. Schemat blokowy urządzenia oraz opis działania i współpracy bloków

W schemacie blokowym urządzenia (rys. 1) można wyróżnić cztery zasadnicze bloki: układ recepcyjny, układ sterujący, układ asocjacyjny i układ wyjściowy.



Rys. 1. Schemat blokowy perceptronu

Informację wejściową o obiekcie stanowi jego obraz fizyczny na siatkówce – pierwszej warstwie układu recepcyjnego. Siatkówka przedstawia układ elementów światłoczułych i zadaniem jej jest przekształcenie informacji ciągłej – obrazu fizycznego – w informację dyskretną – zbiór sygnałów elektrycznych opisywany pewną macierzą $A = [a_{ij}]$ $i=1, \dots, k$, $a_{ij} \in \{0, 1\}$. Przekształcenie to polega na

tym, że zaciemnionym elementom siatkówki przypisuje się wartość 1, elementom oświetlonym zaś wartość 0. W dalszej części układu recepcyjnego odbywają się określone operacje na elementach macierzy A , w wyniku których powstaje macierz $C = [c_i]$ $i=1, \dots, n$, $c_i \in \{1, 0\}$ która w dalszym ciągu nazywa się obrazem badanym albo macierzą cech, jej elementy zaś – cechami obiektu.

Algorytm określenia macierzy C opiera się na założeniu, że jej elementy powinny stanowić taki opis obiektu, żeby ich wartości były możliwie mało zależne od jego przekształceń izomorficznych. Są to w przypadku symboli alfanumerycznych takie przekształcenia jak: zmiany rozmiarów, przesunięcie pionowe lub poziome na siatkówce, skręcenie, pewne zniekształcenia przebiegu linii itp., przy czym przyjmuje się pewne granice dopuszczalnych przekształceń. Granice te zależą od liczby elementów siatkówki oraz od rozwiązywania układu recepcyjnego.

Obraz badany poprzez układ sterujący jest dostarczany do układu asocjacyjnego, który składa się z sieci urzającej się i układu decyzyjnego. Działanie układu asocjacyjnego zależy od tego, czy urządzenie znajduje się w okresie uczenia, czy w okresie rozpoznawania. Decyduje o tym sygnał operatora, który nastawia odpowiednio układ sterujący. W okresie uczenia układ sterujący posyła kolejny obraz badany do sieci uczącej się wraz z sygnałami operatora o nazwie danych obiektów (symboli).

W sieci uczącej się w każdej klasie symboli zostaje przyporządkowany pewien ciąg liczb będący funkcją wszystkich dostarczonych układowi obrazów badanych tej klasy. Ciąg ten po normalizacji (zobacz pkt 4.3) może być traktowany jako punkt w przestrzeni k -wy-miarowej, uogólniający własności obiektów danej klasy i jest nazywany obrazem wzorcowym obiektu.

Obrazy wzorcowe są zapamiętywane (w postaci nieznormalizowanej) jako wiersze pewnej macierzy Y - macierzy stanu pamięci urządzenia i są modyfikowane w czasie każdej operacji uczącej. Operacją uczącą nazywa się cykl działań wykonywanych każdorazowo w okresie uczenia z chwilą przedstawienia urządzeniu określonego obiektu i po podaniu nazwy klasy, do której obiekt ten należy.

W okresie rozpoznawania obraz badany jest posyłany do układu decyzyjnego, który oblicza współczynnik podobieństwa p_i obrazu badanego do każdego obrazu wzorcowego.

Jeżeli dla pewnego obrazu wzorcowego i współczynnik podobieństwa p_i osiąga wartość maksymalną p_{maks} i spełniającą warunek $p_{maks} > p_l$, gdzie p_l jest to pewien parametr o wartości ustalonej doświadczalnie, to symbol odpowiadający temu obrazowi badanemu jest zaliczany do i -tej klasy.

Na okres badań przewidziano możliwość sterowania urządzeniem

przez operatora w zależności od jego oceny poprawności rozpoznawania.

Jeśli rozpoznanie jest poprawne, doświadczenie zostaje utrwalone w sieci urzającej się; jeśli jest błędne, następuje proces uczenia, w którym dany obraz badany zostaje zakwalifikowany do właściwej klasy.

Wszystkie odpowiedzi urządzenia są wydawane przez układ wyjściowy, którym może być tablica sygnalizacyjna, drukarka, maszyna do pisania.

2.2. Przyjęte założenia początkowe

Poniżej podano przyjęte założenia początkowe, które są istotne dla konkretnego rozwiązania układu recepcyjnego. Należy jednak podkreślić, że nie mają one wpływu na ogólną koncepcję algorytmu uczenia i rozpoznawania, a także na zasadniczą ideę budowy układu recepcyjnego.

Obiekty rozpoznawania: symbole alfanumeryczne (przede wszystkim duże litery alfabetu łacińskiego).

Rozmiary macierzy A : $k = 8$, $l = 5$.

Rozmiary macierzy C : $n = 30$.

Granice dopuszczalnych przekształceń:

gabaryty pionowe obrazu fizycznego symbolu (największa rozpiętość w linii pionowej) nie mniejsze niż ok. 85% ($7/8$ długości siatkówki);

gabaryty poziome - nie mniejsze niż około 80% ($4/5$ szerokości siatkówki).

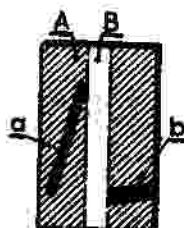
Kąt skręcenia - nie większy niż ok. $15-20^\circ$ w lewo lub w prawo od kierunku osi pionowej siatkówki.

3. UKŁAD RECEPCYJNY

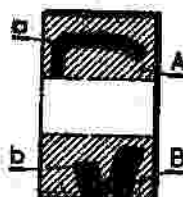
3.1. Krótki opis podstawowej idei budowy układu

Wartości elementów macierzy C będą niezależne od dowolnych przekształceń izomorficznych, jeśli będą charakteryzowały tylko te własności obiektu, które decydują o przynależności obiektu do danej klasy w wyniku założonego podziału obiektów na klasy. W przypadku symboli alfanumerycznych określenie tych istotnych własności

obiektów można sprowadzić na przykład do wykrywania pewnych prostych cech o własnościach niezależnych od przyjętych dopuszczalnych przekształceń. Mogą to być na przykład takie cechy jak: istnienie w określonym obszarze siatkówki linii ciągłej długiej (ze względu na pewne przyjęte kryterium długości) linii ciągłej krótkiej, dwu odcinków linii bez względu na ich długość, trzech odcinków linii itp. (rys. 2, 3, 4).



Rys. 2. a - linia ciągła długa w obszarze A, b - linia ciągła krótka w obszarze B



Rys. 3. a - linia ciągła długa w obszarze A, b - dwa odcinki linii w obszarze B



Rys. 4. Trzy odcinki linii w obszarze A

Przy odpowiednim doborze obszarów siatkówki zbiór cech podobnych może wystarczająco dokładnie opisywać różne znaki alfanumeryczne. Okazuje się ponadto, że wykrywanie tego rodzaju cech wymaga procesu obliczeniowego, który można stosunkowo łatwo zrealizować w oparciu o półprzewodnikową technikę progową.

3.2. Oznaczenia operacji logicznych. Definicje wprowadzonych operatorów

Macierze, które biorą udział w przetwarzaniu informacji wejściowej na obraz badany są z wyjątkiem macierzy sumujących S^1 (zo-

bacz pkt 3.1.5) macierzami o elementach przyjmujących wartości ze zbioru $\{0, 1\}$. Elementy tych macierzy będziemy traktować jako zmienne boolowskie. Ponieważ opis matematyczny układu wykracza jednak częściowo poza ramy algebry Boole'a, dlatego wprowadza się pewien operator logiczny I na funkcjach zdaniowych, który umożliwiał jednocześnie operowanie funkcjami zdaniowymi i funkcjami boolowskimi:

$$a = I(f),$$

gdzie: a - zmienna boolowska, f - funkcja zdaniowa, I - operator logiczny, którego działanie określa się następująco:

$$a = 1, \text{ jeśli } f - \text{prawda}$$

$$a = 0, \text{ jeśli } f - \text{fałsz}$$

Stosuje się następujące oznaczenia funkcji logicznych:

$$f = x_1 \vee x_2 - \text{alternatywa}$$

$$f = x_1 x_2 \text{ lub } x_1 \wedge x_2 - \text{koniunkcja}$$

$$f = x_1 \oplus x_2 - \text{różnica symetryczna (alternatywa wyłączająca)}$$

$$x_1 \oplus x_2 = x_1 \bar{x}_2 \vee \bar{x}_1 x_2$$

W celu ułatwienia opisu matematycznego operacji na macierzach wprowadza się operatory pomocnicze: b_k , d_k , s_k . Działanie danym operatorem na dowolną macierz sprowadza się ogólnie do określonych operacji na jej elementach, w wyniku których otrzymuje się pewną macierz przekształconą.

Niech będzie dana macierz P

$$P = [p_{ij}] \begin{matrix} i=1, \dots, n \\ j=1, \dots, m \end{matrix} = \begin{bmatrix} p_{11} & \dots & p_{1m} \\ \vdots & & \vdots \\ p_{n1} & \dots & p_{nm} \end{bmatrix}$$

Działanie operatorów b_k , d_k , s_k na macierz P określa się następująco:

b_k - operator alternatywy względem k , gdzie k - wskaźnik macierzy $k \in \{1, j\}$

$$b_k^o P = B^o = \begin{matrix} p_{ij}^o & \begin{matrix} i=1, 2, \dots, n \\ j=1, 2, \dots, m-1 \end{matrix} \end{matrix} \left. \begin{matrix} \end{matrix} \right\} \text{ jeśli } k = j$$

$$\begin{matrix} i=1, \dots, n-1 \\ j=1, \dots, m \end{matrix} \left. \begin{matrix} \end{matrix} \right\} \text{ jeśli } k = i$$

gdzie

$$p_{ij}^o = \begin{cases} p_{ij} \vee p_{i, j+1} & - \text{ jeśli } k = j \\ p_{ij} \vee p_{i+1, j} & - \text{ jeśli } k = i \end{cases}$$

d_k - operator różnicy symetrycznej (różnicowy) względem k ,
 $k \in \{i, j\}$
 $d_k P = D^0 = \begin{cases} d_{ij}^0 & \begin{matrix} i=1, \dots, n \\ j=1, \dots, m-1 \end{matrix} \end{cases} \text{ jeśli } k = j$
 $\begin{cases} i=1, \dots, n-1 \\ j=1, \dots, m \end{cases} \text{ jeśli } k = i$

gdzie

$$d_{ij}^0 = \begin{cases} p_{ij} \oplus p_{ij+1} & \text{jeśli } k = j \\ p_{ij} \oplus p_{i+1,j} & \text{jeśli } k = i \end{cases}$$

s_k - operator sumy algebraicznej (sumujący) względem k ,
 $k \in \{i, j\}$
 $s_k P = S^0 = \begin{cases} s_{ij}^0 & \begin{matrix} i=1, \dots, n \\ j=1 \end{matrix} \end{cases} \text{ jeśli } k = j$
 $\begin{cases} i=1 \\ j=1, \dots, m \end{cases} \text{ jeśli } k = i$

gdzie

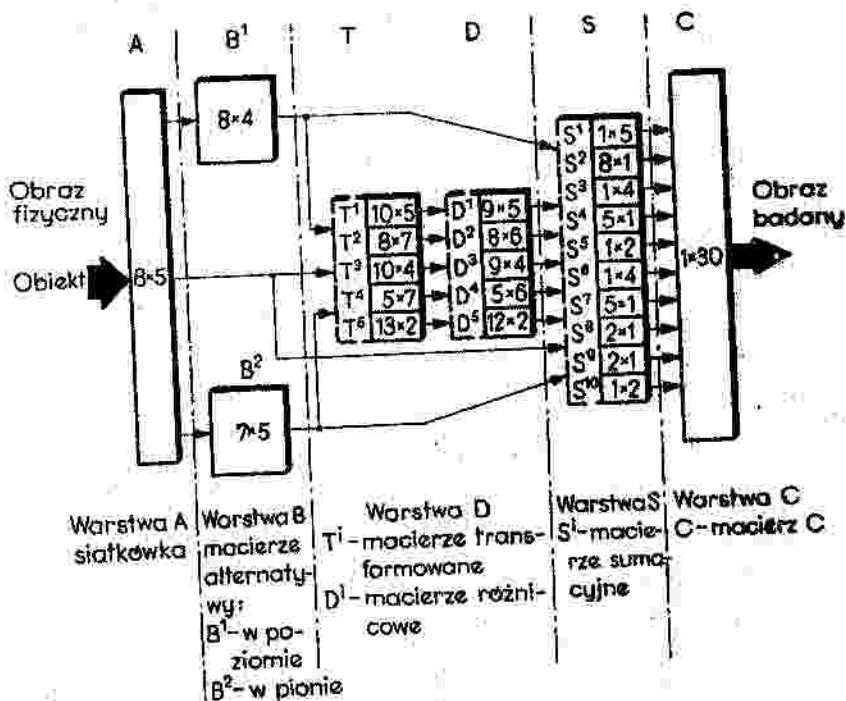
$$s_{ij}^0 = \begin{cases} \sum_{j=1}^m p_{ij} & \text{jeśli } k = j \\ \sum_{i=1}^n p_{ij} & \text{jeśli } k = i \end{cases}$$

Elementy macierzy S^0 traktuje się w tym przypadku jako zmienne algebraiczne.

3.3. Opis matematyczny układu recepcyjnego

3.3.1. Schemat blokowy

Na rys. 5 przedstawiono schemat blokowy układu recepcyjnego. Każdy prostokąt narysowany linią ciągłą przedstawia blok, którego wyjście opisuje się macierzą o nazwie napisanej obok prostokąta oraz rozmiarach wpisanych w środku. Nazwa macierzy stanowi jednocześnie nazwę bloku. Ze względu na analogiczny charakter operacji bloki $D^1, \dots, D^5$ oraz $S^1, \dots, S^{10}$ zostały połączone w układy bloków D i S . Linie pogrubione zakończone strzałką wskazują drogi przechodzenia informacji i pokazują schematycznie, z których bloków czerpie się informację potrzebną do wykonania działań w bloku lub w ukła-



Rys. 5. Schemat blokowy układu recepcyjnego

dzie bloków, do którego dochodzą strzałki.

Cały układ recepcyjny podzielono ze względu na kolejność wykonywanych operacji na warstwy A, B, D, S, C.

3.3.2. Warstwa A

Warstwa A (tzw. siatkówka) składa się z bloku A i stanowi układ techniczny, w którym dokonuje się przekształcenia sygnałów świetlnych na skwantowane sygnały elektryczne. Przyporządkowanie tym ostatnim wartości ze zbioru $\{0, 1\}$ umożliwia opisywanie wyjścia warstwy A w postaci macierzy $A = [a_{ij}]$ $i=1, \dots, k$ $j=1, \dots, l$. W ni-

niejszej pracy przyjęto $k = 8$, $l = 5$ (zob. pkt 2.2).

Warstwa A stanowi więc wyłącznie przetwornik informacji ciągłej na informację dyskretną.

3.3.3. Warstwa B

Warstwa B składa się z dwu bloków B^1 i B^2 . Operacje w blokach tych są określone działaniem operatora b_k na macierz A . W wyniku powstają macierze B^1 oraz B^2 określone następująco:

$$B^1 = b_j A = \begin{bmatrix} b_{1,j}^1 \\ \vdots \\ b_{10,j}^1 \end{bmatrix} \begin{matrix} i=1, \dots, 8 \\ j=1, \dots, 4 \end{matrix}$$

$$B^2 = b_i A = \begin{bmatrix} b_{i,1}^2 \\ \vdots \\ b_{i,5}^2 \end{bmatrix} \begin{matrix} i=1, \dots, 7 \\ j=1, \dots, 5 \end{matrix}$$

Macierz B^1 nazywa się macierzą alternatywy w poziomie, macierz B^2 macierzą alternatywy w pionie.

3.3.4. Warstwa D

Warstwa D składa się z układów bloków T i D zawierających bloki $T^1, \dots, T^5$ oraz $D^1, \dots, D^5$.

Macierze transformowane $T^1, \dots, T^5$ powstają w wyniku przekształcenia macierzy A , B^1 , B^2 według następujących zależności:

$$T^1 = \begin{bmatrix} t_{1,1}^1 & t_{1,2}^1 & \dots & t_{1,5}^1 \\ \vdots & \vdots & \ddots & \vdots \\ t_{2,1}^1 & t_{2,2}^1 & \dots & t_{2,5}^1 \\ \vdots & \vdots & \ddots & \vdots \\ t_{10,1}^1 & \dots & \dots & t_{10,5}^1 \end{bmatrix} A = \begin{bmatrix} t_{i,j}^1 \\ \vdots \\ t_{i,5}^1 \end{bmatrix} \begin{matrix} i=1, \dots, 10 \\ j=1, \dots, 5 \end{matrix}$$

gdzie

$$t_{i,j}^1 = \begin{cases} 0 & \text{dla } i = 1, 10 \\ & j = 1, \dots, 5 \\ a_{i-1,j} & \text{dla } i = 2, \dots, 9 \\ & j = 1, \dots, 5 \end{cases}$$

$$T^2 = \begin{bmatrix} t_{1,1}^2 & t_{1,2}^2, \dots, t_{1,6}^2 & t_{1,7}^2 \\ t_{2,1}^2 & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ t_{8,1}^2 & t_{8,2}^2, \dots, t_{8,6}^2 & t_{8,7}^2 \end{bmatrix} = [t_{i,j}^2]_{\substack{i=1,\dots,8 \\ j=1,\dots,7}}$$

gdzie

$$t_{i,j}^2 = \begin{cases} 0 & \text{dla } \begin{matrix} i = 1, 2, \dots, 7, 8 \\ j = 1, 7 \end{matrix} \\ a_{i,j-1} & \text{dla } \begin{matrix} i = 1, 2, \dots, 7, 8 \\ j = 2, \dots, 6 \end{matrix} \end{cases}$$

$$T^3 = \begin{bmatrix} t_{1,1}^3, \dots, t_{1,4}^3 \\ t_{2,1}^3, \dots, t_{2,4}^3 \\ \vdots \\ \vdots \\ \vdots \\ t_{9,1}^3, \dots, t_{9,4}^3 \\ \vdots \\ t_{10,1}^3, \dots, t_{10,4}^3 \end{bmatrix} = [t_{i,j}^3]_{\substack{i=1,\dots,10 \\ j=1,\dots,4}} \quad B^1$$

gdzie

$$t_{i,j}^3 = \begin{cases} 0 & \text{dla } \begin{matrix} i = 1, 6 \\ j = 1, \dots, 4 \end{matrix} \\ b_{i-1,j}^1 & \text{dla } \begin{matrix} i = 2, 9 \\ j = 1, \dots, 4 \end{matrix} \end{cases}$$

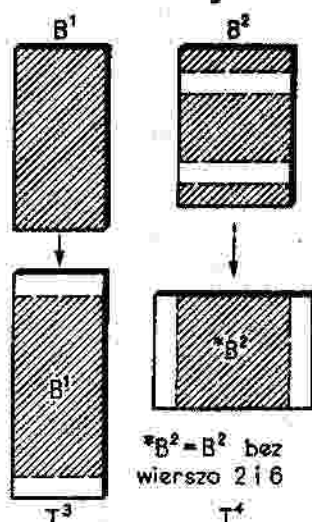
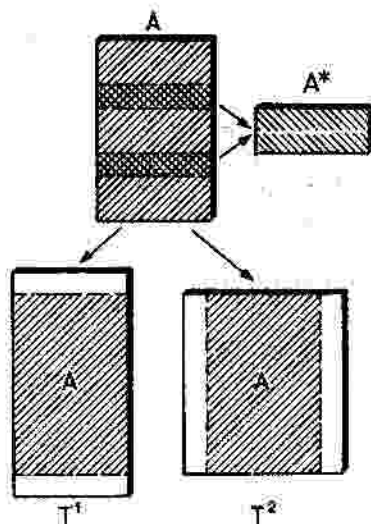
$$T^4 = \begin{bmatrix} t_{1,1}^4 & t_{1,2}^4 & \dots & t_{1,6}^4 & t_{1,7}^4 \\ \vdots & \vdots & & \vdots & \vdots \\ t_{5,1}^4 & t_{5,2}^4 & \dots & t_{5,6}^4 & t_{5,7}^4 \end{bmatrix} = [t_{i,j}^4]_{\substack{i=1,\dots,5 \\ j=1,\dots,7}}$$

$*B^2 \rightarrow$

gdzie

$$t_{i,j}^4 = 0 \quad \text{dla} \quad i = 1, \dots, 5 \\ j = 1, 7$$

$$[t_{i,j}^4]_{\substack{i=1,\dots,5 \\ j=2,\dots,6}} = \begin{bmatrix} b_{1,1}^2 & b_{1,2}^2 & \dots & b_{1,5}^2 \\ b_{2,1}^2 & b_{2,2}^2 & \dots & b_{2,5}^2 \\ b_{3,1}^2 & b_{3,2}^2 & \dots & b_{3,5}^2 \\ b_{4,1}^2 & \dots & \dots & b_{4,5}^2 \\ b_{5,1}^2 & \dots & \dots & b_{5,5}^2 \\ b_{7,1}^2 & b_{7,2}^2 & \dots & b_{7,5}^2 \end{bmatrix}$$



Rys. 6. Przekształcenie macierzy A w macierze T^1 , T^2 i A^*

Rys. 7. Przekształcenie macierzy B^1 i B^2 w macierze T^3 i T^4

$$T^5 = [t_{i,j}^5]_{\substack{i=1,\dots,13 \\ j=1,2}}$$

gdzie

$$t_{i,j}^5 = 0 \quad \text{dla} \quad \begin{matrix} i = 1, 13 \\ j = 1, 2 \end{matrix}$$

$$[t_{i,j}^5]_{\substack{i=2,\dots,12 \\ j=1,2}} =$$

$$\begin{bmatrix} b_{11}^2 & b_{15}^2 \\ b_{21}^1 & b_{24}^1 \\ b_{22}^2 & b_{24}^2 \\ b_{32}^2 & b_{34}^2 \\ b_{42}^1 & b_{43}^1 \\ b_{43}^2 & b_{43}^2 \\ b_{53}^1 & b_{52}^1 \\ b_{54}^2 & b_{52}^2 \\ b_{64}^2 & b_{62}^2 \\ b_{74}^1 & b_{71}^1 \\ b_{75}^2 & b_{71}^2 \end{bmatrix}$$

Przekształcanie macierzy A , B^1 i B^2 na macierze $T^1, \dots, T^5$ przedstawiono na rys. 6, 7 i 8.

W blokach $D^1, \dots, D^5$ dokonuje się operacji określonych działaniem operatora d_k na macierze odpowiednio $T^1, \dots, T^5$. Macierze D^1 nazywane macierzami różnicowymi są określone następująco:

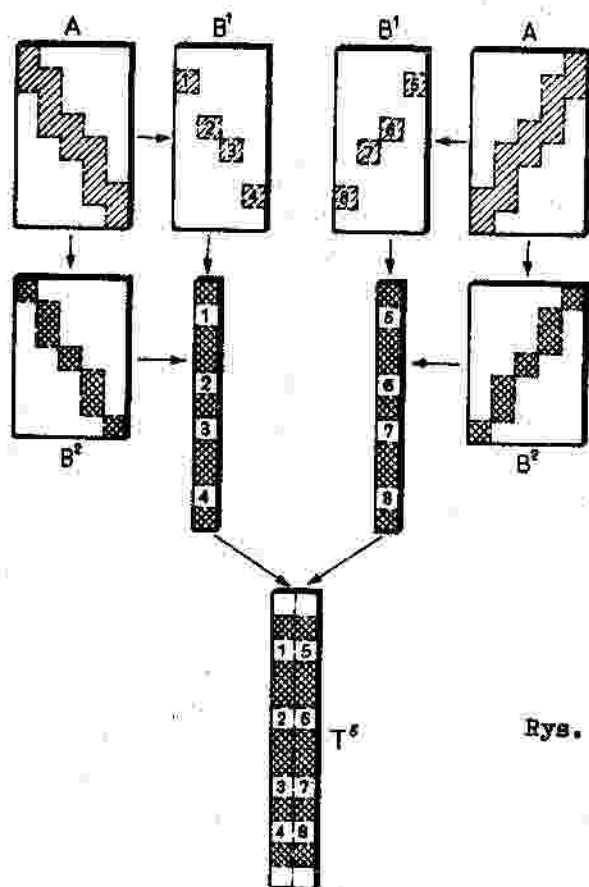
$$D^1 = d_1 T^1 = [d_{ij}^1]_{\substack{i=1,\dots,9 \\ j=1,\dots,5}}$$

$$D^2 = d_2 T^2 = [d_{ij}^2]_{\substack{i=1,\dots,8 \\ j=1,\dots,6}}$$

$$D^3 = d_3 T^3 = [d_{ij}^3]_{\substack{i=1,\dots,9 \\ j=1,\dots,4}}$$

$$D^4 = d_j T^4 = \begin{bmatrix} d_{1j}^4 \end{bmatrix} \begin{matrix} i=1, \dots, 5 \\ j=1, \dots, 6 \end{matrix}$$

$$D^5 = d_1 T^5 = \begin{bmatrix} d_{1j}^5 \end{bmatrix} \begin{matrix} i=1, \dots, 12 \\ j=1, 2 \end{matrix}$$



Rys. 8. Tworzenie macierzy T^5

3.3.5. Warstwa S

Warstwa S stanowi układ bloków S składający się z bloków $S^1, \dots, S^{10}$.

Macierze sumacyjne $S^1, \dots, S^{10}$ powstają w wyniku działania operatora s_k na odpowiednie macierze $D^1, \dots, D^5$ oraz pewne pod-

macierze macierzy A , B^1 i B^2 .

$$\begin{aligned} s^1 &= s_i D^1 = \begin{bmatrix} s_{1j}^1 \\ s_{2j}^1 \end{bmatrix} \quad j=1, \dots, 5 \\ s^2 &= s_j D^2 = \begin{bmatrix} s_{11}^2 \\ s_{21}^2 \end{bmatrix} \quad i=1, \dots, 8 \\ s^3 &= s_i D^3 = \begin{bmatrix} s_{1j}^3 \\ s_{2j}^3 \end{bmatrix} \quad j=1, \dots, 4 \\ s^4 &= s_j D^4 = \begin{bmatrix} s_{11}^4 \\ s_{21}^4 \end{bmatrix} \quad i=1, \dots, 5 \\ s^5 &= s_i D^5 = \begin{bmatrix} s_{1j}^5 \\ s_{2j}^5 \end{bmatrix} \quad j=1, 2 \\ s^6 &= s_i B^1 = \begin{bmatrix} s_{1j}^6 \\ s_{2j}^6 \end{bmatrix} \quad j=1, \dots, 4 \\ s^7 &= s_j^* B^2 = \begin{bmatrix} s_{11}^7 \\ s_{21}^7 \end{bmatrix} \quad i=1, \dots, 5 \end{aligned}$$

gdzie

$${}^*B^2 = \begin{bmatrix} b_{1j}^2 \\ b_{2j}^2 \end{bmatrix}_{\substack{i=1, \dots, 5 \\ j=1, \dots, 5}} = \begin{bmatrix} b_{1,1}^2, \dots, b_{1,5}^2 \\ b_{2,1}^2, \dots, b_{2,5}^2 \\ b_{3,1}^2, \dots, b_{3,5}^2 \\ b_{4,1}^2, \dots, b_{4,5}^2 \\ b_{5,1}^2, \dots, b_{5,5}^2 \\ b_{6,1}^2, \dots, b_{6,5}^2 \\ b_{7,1}^2, \dots, b_{7,5}^2 \end{bmatrix} \quad (\text{zob. rys. 7})$$

$$s^8 = s_j B^3 = \begin{bmatrix} s_{11}^8 \\ s_{21}^8 \end{bmatrix} \quad i=1, 2$$

gdzie

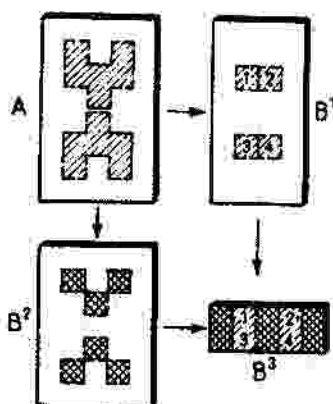
$$B^3 = \begin{bmatrix} b_{2,2}^2, b_{3,2}^1, b_{3,3}^2, b_{3,3}^1, b_{2,4}^2 \\ b_{6,2}^2, b_{6,2}^1, b_{5,3}^2, b_{6,3}^1, b_{6,4}^2 \end{bmatrix} \quad (\text{zob. rys. 9})$$

$$s^9 = s_j A^* = \begin{bmatrix} s_{11}^9 \\ s_{21}^9 \end{bmatrix} \quad i=1, 2$$

gdzie

$$A^* = \begin{bmatrix} a_{1j}^* \\ a_{2j}^* \end{bmatrix}_{\substack{i=1,2 \\ j=1, \dots, 5}} = \begin{bmatrix} a_{3,1}, \dots, a_{3,5} \\ a_{6,1}, \dots, a_{6,5} \end{bmatrix} \quad (\text{zob. rys. 6})$$

$$s^{10} = s_1 T^5 = \begin{bmatrix} s_{1j}^{10} \\ s_{2j}^{10} \end{bmatrix} \quad j=1, 2$$



Rys. 9
Tworzenie macierzy B^3

3.3.6. Warstwa C

W warstwie C dokonuje się określonych operacji logicznych na elementach macierzy $S^1, \dots, S^{10}$. W wyniku otrzymuje się macierz cech $C = [c_i]_{i=1, \dots, k}$ ($k = 30$, zob. pkt 2.2).

Elementy macierzy C określa się następująco:

$$\begin{aligned}
 c_1 &= I(s_{1,1}^3 = 2 \wedge s_{1,1}^6 \geq 5) \\
 c_2 &= I(s_{1,1}^3 = 2 \wedge s_{1,1}^6 < 5) \\
 c_3 &= I(s_{1,1}^3 = 4 \vee s_{1,1}^1 = 4) \\
 c_4 &= I(s_{1,4}^3 = 2 \wedge s_{1,4}^6 \geq 5) \\
 c_5 &= I(s_{1,4}^3 = 2 \wedge s_{1,4}^6 < 5) \\
 c_6 &= I(s_{1,4}^3 = 4 \vee s_{1,4}^1 = 4) \\
 c_7 &= I(s_{1,2}^3 = 2 \wedge s_{1,2}^6 \geq 5 \vee s_{1,3}^3 = 2 \wedge s_{1,3}^6 \geq 5) \\
 c_8 &= I(s_{1,3}^1 = 4) \\
 c_9 &= I(s_{1,3}^1 = 6) \\
 c_{10} &= I(s_{1,1}^4 = 2 \wedge s_{1,1}^2 \neq 4 \wedge s_{1,1}^7 \geq 3) \\
 c_{11} &= I(s_{1,1}^4 = 2 \wedge s_{1,1}^7 < 3)
 \end{aligned}$$

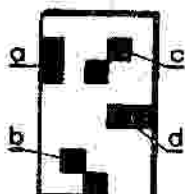
$$\begin{aligned}
c_{12} &= I(s_{1,1}^4) = 4 \vee s_{1,1}^2 = 4) \\
c_{13} &= I(s_{1,1}^8) = 5) \\
c_{14} &= I(s_{3,1}^2) = 2 \wedge s_{1,1}^9 \leq 2) \\
c_{15} &= I(s_{2,1}^2) = 4 \vee s_{3,1}^2 = 4 \vee s_{4,1}^2 = 4) \\
c_{16} &= I(s_{2,1}^8) = 5) \\
c_{17} &= I(s_{6,1}^2) = 2 \wedge s_{2,1}^9 \leq 2) \\
c_{18} &= I(s_{5,1}^2) = 4 \wedge s_{6,1}^2 = 4 \wedge s_{7,1}^2 = 4) \\
c_{19} &= I(s_{5,1}^4) = 2 \wedge s_{8,1}^2 \neq 4 \wedge s_{5,1}^7 \geq 3) \\
c_{20} &= I(s_{5,1}^4) = 2 \wedge s_{5,1}^7 < 3) \\
c_{21} &= I(s_{5,1}^4) = 4 \vee s_{8,1}^2 = 4) \\
c_{22} &= I(s_{2,1}^4) = 2 \wedge s_{2,1}^7 \geq 3 \vee s_{3,1}^4 = 2 \wedge s_{3,1}^7 \geq 3 \\
&\quad s_{4,1}^4 = 2 \wedge s_{4,1}^7 \geq 3) \\
c_{23} &= I(s_{3,1}^4) = 2 \wedge s_{3,1}^7 < 3) \\
c_{24} &= I(s_{3,1}^4) = 4 \vee s_{3,1}^4 = 6) \\
c_{25} &= I(s_{1,1}^5) = 2 \wedge s_{1,1}^{10} \geq 9) \\
c_{26} &= I(s_{1,1}^5) = 2 \wedge s_{1,1}^{10} < 9) \\
c_{27} &= I(s_{1,1}^5) = 4) \\
c_{28} &= I(s_{1,2}^5) = 2 \wedge s_{1,2}^{10} \geq 9) \\
c_{29} &= I(s_{1,2}^5) = 2 \wedge s_{1,2}^{10} < 9) \\
c_{30} &= I(s_{1,2}^5) = 4)
\end{aligned}$$

3.4. Interpretacja elementów macierzy cech C

Wartości elementów macierzy cech charakteryzują określone własności obrazu fizycznego. Wartość 1 danego elementu macierzy oznacza występowanie danej własności, wartość 0 oznacza niewystępowanie jej w obrazie fizycznym.

Poniżej zestawiono opis własności odpowiadających poszczególnym elementom (cechom) macierzy C. Używa się tu pojęcia linii ciągłej w danym obszarze siatkówki oraz pojęcia długości linii 1. Pojęcia te są określone następująco:

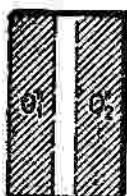
Mówi się, że w danym obszarze siatkówki występuje linia ciągła (jedna), jeśli wszystkie pobudzone elementy siatkówki w danym obszarze tworzą ciąg elementów sąsiadujących ze sobą, przy czym za sąsiednie uważa się zarówno elementy sąsiadujące w linii poziomej, pionowej, jak i na w skos (rys. 10). Analogicznie określa się pojęcie dwóch linii ciągłych, trzech itd. w danym obszarze.



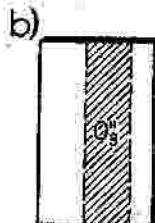
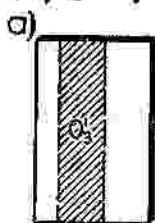
Rys. 10. Pary elementów a, b, c, d są parami elementów sąsiadujących ze sobą

Długość linii zależy od ilości pobudzonych elementów w wydzielonym obszarze siatkówki: każdy obszar siatkówki przekształca się do postaci wiersza lub kolumny odpowiedniej macierzy B^1 , B^2 , B^3 i A^* lub T^5 i długość linii jest określona ilością elementów o wartości 1 w danym wierszu lub w kolumnie.

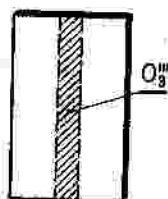
Długość linii w wydzielonych obszarach siatkówki wyrażają elementy macierzy S^6 , ..., S^{10} .



Rys. 11



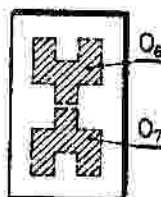
Rys. 12



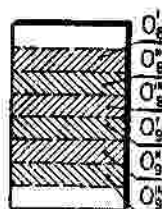
Rys. 13



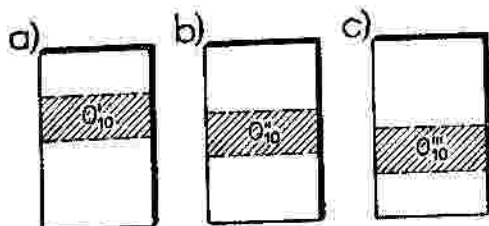
Rys. 14



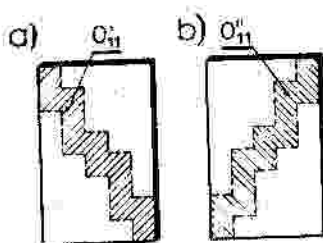
Rys. 15



Rys. 16



Rys. 17



Rys. 18

Opis własności obrazu fizycznego odpowiadających poszczególnym cechom jest następujący:

- c_1 - linia ciągła długa ($1 \geq 5$) w obszarze O'_1 (rys. 11)
- c_2 - linia ciągła krótka ($1 < 5$) w obszarze O'_1
- c_3 - dwie linie ciągłe w obszarze O'_1 lub O''_1
- c_4 - linia ciągła długa ($1 \geq 5$) w obszarze O'_2 (rys. 11)
- c_5 - linia ciągła krótka ($1 < 5$) w obszarze O'_2
- c_6 - dwie linie ciągłe w obszarze O'_2 lub O''_2
- c_7 - linia ciągła długa ($1 \geq 5$) w obszarze O'_3 lub O''_3 (rys. 12)
- c_8 - dwie linie ciągłe w obszarze O''_3
- c_9 - trzy linie ciągłe w obszarze O''_3 (rys. 13)
- c_{10} - linia ciągła długa ($1 \geq 3$) w obszarze O'_4 i brak dwóch linii ciągłych w obszarze O''_4 (rys. 14)
- c_{11} - linia ciągła krótka ($1 < 3$) w obszarze O'_4
- c_{12} - dwie linie ciągłe w obszarze O'_4 lub O''_4
- c_{13} - linia ciągła w obszarze O_6 (rys. 15), przy czym jest spełniony warunek $b_{22}^2 + b_{32}^1 + b_{33}^2 + b_{33}^1 + b_{24}^2 = 5$
- c_{14} - linia ciągła krótka ($1 \leq 2$) w obszarze O_8'' (rys. 16)
- c_{15} - dwie linie ciągłe w obszarze O_8' lub O_8'' lub O_8''' (rys. 16)
- c_{16} - linia ciągła w obszarze O_7 (rys. 15), przy czym jest spełniony warunek $b_{62}^2 + b_{62}^1 + b_{53}^2 + b_{63}^1 + b_{64}^2 = 5$
- c_{17} - linia ciągła krótka ($1 \leq 2$) w obszarze O_9'' (rys. 16)
- c_{18} - dwie linie ciągłe w obszarze O_9' lub O_9'' lub O_9'''
- c_{19} - linia ciągła długa ($1 \geq 3$) w obszarze O'_5 i brak dwóch linii ciągłych w obszarze O''_5 (rys. 14)
- c_{20} - linia ciągła krótka ($1 < 3$) w obszarze O'_5
- c_{21} - dwie linie ciągłe w obszarze O'_5 lub O''_5

- c_{22} - linia ciągła długa ($1 > 3$) w obszarze O'_{10} lub O''_{10} lub O'''_{10}
(rys. 17)
- c_{23} - linia ciągła krótka ($1 < 3$) w obszarze O''_{10}
- c_{24} - dwie lub trzy linie ciągłe w obszarze O''_{10}
- c_{25} - linia ciągła długa ($1 \geq 9$) w obszarze O'_{11} (rys. 18a)
- c_{26} - linia ciągła krótka ($1 < 9$) w obszarze O'_{11}
- c_{27} - dwie linie ciągłe w obszarze O'_{11}
- c_{28} - linia ciągła długa ($1 \geq 9$) w obszarze O''_{11} (rys. 18b)
- c_{29} - linia ciągła krótka ($1 < 9$) w obszarze O''_{11}
- c_{30} - dwie linie ciągłe w obszarze O''_{11}

3.5. Charakterystyka ogólna układu

Opisany układ recepcyjny powstał w wyniku analizy kształtu obrazu fizycznego różnych symboli na siatkówce 8×5 . Jako symbole rozważano przede wszystkim duże litery alfabetu łacińskiego, mimo że brano również pod uwagę możliwość rozpoznawania małych liter alfabetu łacińskiego oraz małych i dużych liter alfabetu rosyjskiego, greckiego i różnych znaków specjalnych. kierowano się przy tym zasadą, żeby uzyskać układ wykrywający możliwie najistotniejsze cechy symboli, ale jednocześnie najprostszy do realizacji technicznej w oparciu o półprzewodnikową technikę progową.

Przetwarzanie informacji w warstwach B, D, S ma charakter uniwersalny, niezależny od rodzaju obrazu fizycznego powstającego na siatkówce. Wpływ rodzaju rozpoznawanych symboli uwidocznia się dopiero w budowie warstwy C. Umożliwia to zmianę ilości i jakości cech bez jakichkolwiek zmian we wspomnianych warstwach. Zmiana rozmiarów siatkówki również jest możliwa bez dokonania większych zmian w układzie oprócz zwiększenia ilości operacji. Koncepcja budowy warstwy C - jak zresztą i całego układu recepcyjnego - jest koncepcją próbną. Jest rzeczą oczywistą, że ściśle jej uzasadnienie matematyczne byłoby nader kłopotliwe. Dlatego jej przydatność lub nieprzydatność oraz celowość zmian w niej mogą wykazać odpowiednie badania eksperymentalne.

4. UKŁAD ASOCJACYJNY

4.1. Sieć ucząca się. Opis algorytmu uczenia

Sieć uczącą się można traktować jako pewnego rodzaju urządzenie pamięciowe, w którym gromadzi się informacja o nauczonych symbolach. Można ją opisać za pomocą tablicy Y zbudowanej w sposób następujący:

$$\text{Tablica } Y = \begin{bmatrix} Y_1 & (\text{nazwa 1 klasy symboli}) & [y_{10}, y_{11}, \dots, y_{1n}] \\ Y_2 & \cdot & [y_{20}, y_{21}, \dots, y_{2n}] \\ Y_3 & \cdot & \cdot \\ \vdots & \vdots & \vdots \\ Y_m & (\text{nazwa } m \text{ klasy symboli}) & [y_{m0}, y_{m1}, \dots, y_{mn}] \end{bmatrix} \quad (1)$$

gdzie: m - ilość nauczonych klas obiektów, $n = 30$ (ilość elementów macierzy C).

Każdy wiersz tablicy ma numer porządkowy i ; zawiera też nazwę klasy symboli związanej z tym numerem (krótko: klasy i) oraz macierz wierszową $Y_i = [y_{i0}, \dots, y_{in}]$ nazwaną nie znormalizowanym obrazem wzorcowym klasy i . Część tablicy Y , w której są ułożone wierszami macierze Y_i , $i = 1, \dots, m$, traktuje się jako pewną macierz Y_s - macierz stanu pamięci urządzenia

$$Y_s = [y_{ij}] \quad \begin{matrix} i=1, \dots, m \\ j=1, \dots, n \end{matrix} \quad (2)$$

Elementy y_{ij} są określone następująco:

$$y_{10} = L_1 \quad (3)$$

$$y_{ij} = \sum_{k=1}^{L_1} c_{ij}^k, \quad j = 1, \dots, n \quad (4)$$

gdzie: c_{ij}^k - element j -ty macierzy $C_1^k = [c_{11}^k, c_{12}^k, \dots, c_{1n}^k]$, przy czym macierz C_1^k stanowi k -ty wyraz ciągu macierzy $C_1^1, C_1^2, \dots, C_1^{L_1}$, które w okresie uczenia były przez uczącego przypisywane klasie i ; L_1 - długość (ilość wyrazów) ciągu uczącego $C_1^1, C_1^2, \dots, C_1^{L_1}$ klasy i .

Załóżmy, że aktualny stan sieci uczącej się opisuje się tablicą

Y^0 , której odpowiada macierz

$$Y_s^0 = [y_{ij}^0] \begin{matrix} i=1, \dots, m \\ j=0, \dots, n \end{matrix}$$

Jeżeli nastąpi teraz okres uczenia na przykład symbolu o macierzy $C_r = [c_{rj}]$ $j=1, \dots, n$, któremu zostanie przypisana nazwa klasy r , to będą wykonane działania:

1. Sprawdzenie, czy nazwa klasy r jest jedną z nazw zawartych w tablicy Y^0 , t.j. czy numerowi r można przypisać wartość ze zbioru $\{1, \dots, m\}$. W zależności od wyniku sprawdzenia będą wykonane działania określone w pktcie 2a lub 2b.

2a. $r \in \{1, \dots, m\}$. Zostanie odszukana macierz wierszowa $Y_r^0 = [y_{r0}^0, \dots, y_{rn}^0]$ i będą wykonane działania

$$y_{r0}^1 = y_{r0}^0 + 1$$

$$y_{rj}^1 = y_{rj}^0 + c_{rj}^1, \quad j = 1, \dots, n$$

W tablicy Y^0 na miejsce macierzy Y_r^0 zostanie zapisana macierz $Y_r^1 = [y_{r0}^1, \dots, y_{rn}^1]$.

2b. $r \notin \{1, \dots, m\}$. Urządzenie jest uczone nieznanego dotychczas symbolu, r przyjmie wartość $(m+1)$, a macierz Y_s^0 powiększy się o nowy wiersz

$$Y_{m+1} = [y_{m+1,0}^0, y_{m+1,1}^0, \dots, y_{m+1,n}^0]$$

gdzie

$$y_{m+1,0}^0 = 1$$

$$y_{m+1,j}^0 = c_{rj}^0, \quad j = 1, \dots, n$$

W tablicy Y^0 nowemu wierszowi $Y_{m+1} = Y_r$ zostanie przypisana nazwa klasy r .

Jeśli urządzenie nie było jeszcze uczone, $m=0$, to pierwszy proces uczenia będzie polegał na wykonaniu działań opisanych w punkcie 2b. Następny proces nauczania odbędzie się jak opisano powyżej.

Posługiwanie się macierzą Y_s (wzór (2)) jest wygodne do realizacji praktycznej algorytmu uczenia. Natomiast do jego interpretacji geometrycznej oraz do opisu algorytmu rozpoznawania jest wy-

godniej używać macierzy W , stanowiącej znormalizowaną postać macierzy Y

$$W = [w_{ij}]_{\substack{i=1, \dots, n \\ j=1, \dots, m}} \quad (6)$$

gdzie

$$w_{ij} = y_{ij}/y_{i0}$$

Ponieważ dla dowolnego $i \in \{1, \dots, m\}$ oraz $j \in \{1, \dots, n\}$

$$y_{ij} \leq y_{i0}$$

przeto

$$\bigwedge_{i \in \{1, \dots, m\}} \bigwedge_{j \in \{1, \dots, n\}} (0 \leq w_{ij} \leq 1) \quad (7)$$

Macierz wierszową $W_i = [w_{i1}, \dots, w_{in}]$, $i \in \{1, \dots, m\}$ stanowiącą i -ty wiersz macierzy W nazywamy obrazem wzorcowym i -tej klasy symboli.

Elementy w_{ij} można interpretować jako częstości występowania cechy c_j w obrazach badanych i -tej klasy lub jako wagi cechy c_j w obrazie badanym ze względu na i -tą klasę.

4.2. Układ decyzyjny. Opis algorytmu rozpoznawania

W układzie decyzyjnym jest realizowany algorytm rozpoznawania, który kończy się powzięciem decyzji o tym, do której klasy zalicza się obraz badany. Można w nim wyróżnić następujące etapy:

1. Określenie współczynników bliskości (podobieństwa) $p_i(C, W_i)$ obrazu badanego C do kolejnych obrazów wzorcowych W_i , $i = 1, \dots, m$.

Elementy macierzy $C = [c_j]_{j=1, \dots, n}$ są tu traktowane jako zwykle zmienne algebraiczne c_j 0, 1.

Współczynnik $p_i(C, W_i)$ definiuje się następująco:

$$p_i(C, W_i) = \frac{1}{n} \sum_j [c_j w_{ij} + (1 - c_j)(1 - w_{ij})] \quad (8)$$

gdzie j przebiega wartości ze zbioru $\{1, \dots, n\}$, przy których $w_{ij} \leq g_d$ lub $w_{ij} \geq g_g$, przy czym g_g oraz g_d są to parametry

nazwane odpowiednio: progiem górnym i progiem dolnym i spełniające warunki:

$$0,5 \leq gg \leq 1, \quad 0 \leq gd \leq 0,5 \quad (8a)$$

$$n' = \sum_j 1_j, \quad \text{gdzie} \quad \bigwedge_j (1_j = 1) \quad (8b)$$

Parametry gg oraz gd określają zakresy wartości elementów w_{1j} , przy których wagi cech w_{1j} biorą udział w obliczaniu p_1 . Jeśli bowiem np. $w_{1j} = 0,5$, a więc częstość występowania cechy c_j w obrazach badanych i-tej klasy stanowi połowę wszystkich przypadków, to oznacza, że cecha c_j jest mało istotna dla symboli tej klasy i dlatego może nie być brana pod uwagę. Jest więc rzeczą naturalną przyjęcie pewnych granic wartości w_{1j} , które będą brały udział przy określaniu współczynnika podobieństwa p_1 . Wartości parametrów gg oraz gd można ustalać doświadczalnie.

Tak określony współczynnik bliskości spełnia relację (dowód w pktcie 4.3)

$$\bigwedge_{i \in \{1, \dots, m\}} (0 \leq p_1(C, W_1) \leq 1) \quad (9)$$

W pktcie 4.3 będzie wykazane, że p_1 jest liniową (malejącą) funkcją miary odległości w liniowej unormowanej przestrzeni n' -wymiarowej.

Do realizacji technicznej wygodniejsza jest następująca postać

$$p_1(C, Y_1) = \frac{1}{n' y_{10}} \sum_j [c_j y_{1j} + (1 - c_j)(y_{10} - y_{1j})] \quad (10)$$

Zamiast macierzy W_1 stosuje się tu macierz Y_1 .

2. Określenie P_{maks} oraz P'_{maks} . Obliczone współczynniki p_i $i = 1, \dots, m$ można ustawić w ciąg o wzrastających wartościach. Ostatni element tego ciągu nazywamy P_{maks} , a przedostatni P'_{maks} . Załóżmy, że

$$P_{maks} = P_r, \quad P'_{maks} = P_s$$

Współczynniki p_r odpowiada r -ta, a współczynniki p_s -

s-ta klasa symboli.

Symbol, dla którego $P_m = P_{maks}$, można więc odnieść do klasy m , ale pod warunkiem, że jest spełniony dodatkowy warunek

$$P_{maks} \geq p_1 \quad (11)$$

przy czym p_1 - pewien parametr ustalony doświadczalnie. Może bowiem zdarzyć się, że dany symbol należy do klasy nie znanej urządzeniu lub że jest zbyt odkształcony, żeby mógł być zakwalifikowany do określonej klasy. W takim przypadku można jedynie powiedzieć że dany symbol jest najbardziej podobny do danego obrazu wzorcowego w sensie przyjętej miary podobieństwa, oczekuje się jednak, że wartość P_{maks} będzie stosunkowo mała.

W związku z tym w zależności od zakresu wartości, w którym mieści się P_{maks} , jest celowe wyróżnienie pewnych kategorii rozpoznawania. Granice tych zakresów, np. $p_1, p_2, \dots$, mogą być ustalone doświadczalnie.

Przykładowo wybrane kategorie rozpoznawania podano przy opisie modelu urządzenia rozpoznającego na maszynie cyfrowej (pkt 5).

Z powodu dużej gruboziarnistości siatekówki 8×5 można spodziewać się, że pewne symbole, np. 0 oraz D, nie będą rozróżniane. Aby uniknąć w takich przypadkach odpowiedzi błędnych, jest celowe zastosowanie rozpoznawania alternatywnego, polegającego na tym, że urządzenie daje odpowiedź w postaci np. "Jest to symbol 0 lub D". O tym, czy rozpoznawanie ma być jednoznaczne, czy alternatywne, może decydować wartość

$$\Delta = P_{maks} - P'_{maks} \quad (12)$$

Jeśli $\Delta \leq \Delta_0$, gdzie Δ_0 - pewna wartość graniczna, to stosuje się rozpoznawanie alternatywne.

4.3. Interpretacja geometryczna algorytmu uczenia i rozpoznawania

Jeśli elementy macierzy $A = [a_{ij}] \begin{matrix} i=1, \dots, k \\ j=1, \dots, l \end{matrix}$ traktować jako współrzędne $k \times l$ -wymiarowej przestrzeni, to każdej macierzy A będzie odpowiadał określony punkt w tej przestrzeni. Ponieważ $a_{ij} \in \{0, 1\}$, przeto punkty odpowiadające poszcze-

gólnym macierzom A będą układać się w wierzchołkach kostki jednostkowej.

Przestrzeń tę, w opisywanym przypadku 40-wymiarową: $k = 8$, $l = 5$ nazywamy przestrzenią obrazów fizycznych E_F . Obiektom tej samej klasy, np. klasy 1, będzie odpowiadać w przestrzeni E_F zbiór punktów (wierzchołków) klasy 1.

Załóżmy, że w przestrzeni E_F jest wyprowadzona miara odległości typu

$$\rho(X, Y) = \sum_{i=1}^{k \times l} x_i - y_i \quad (13a)$$

lub

$$\rho(X, Y) = \sqrt{\sum_{i=1}^{k \times l} (x_i - y_i)^2} \quad (13b)$$

gdzie: $X = [x_1, \dots, x_{k \times l}]$, $Y = [y_1, \dots, y_{k \times l}]$.

Rozważając różne przykłady symboli tych samych klas można łatwo wykazać, że punkty tych samych klas mogą być rozrzucone po całej przestrzeni E_F , tzn. że nie będą tworzyły oddzielnych skupień, rozumianych np. w ten sposób, że średnie odległości między punktami jednej klasy będą znacznie mniejsze od średnich odległości między punktami różnych klas.

Układ recepcyjny przekształca powyższą 40-wymiarową przestrzeń w 30-wymiarową przestrzeń obrazów badanych E_b . Każdemu obrazowi badanemu macierzy $C = [c_i]_{i=1, \dots, 30}$ odpowiada w przestrzeni E_b analogicznie punkt - wierzchołek 30-wymiarowej kostki jednostkowej. Dzięki małej zależności macierzy C od przekształceń izomorficznych obiektów można tu już mówić o pewnych skupieniach punktów dla każdej klasy (zakłada się tu tę samą metrykę (13a) lub (13b)). W każdym skupieniu można więc wyróżnić pewien punkt środkowy - środek rozkładu - charakteryzujący daną klasę.

Wykażemy, że przyjętemu algorytmowi uczenia odpowiada interpretacja geometryczna, w której współrzędne środka rozkładu są średnimi arytmetycznymi współrzędnych punktów jednej klasy.

Rozpatrzmy w tym celu np. macierz $W_p = [w_{p1}, \dots, w_{pn}]$ - obraz wzorcowy p -tej klasy obiektów.

Elementy $w_{p1}, \dots, w_{pn}$ można traktować jako współrzędne pewnego punktu S_p w przestrzeni E_b . Ze względu na zależność (7)

punkt ten będzie znajdował się wewnątrz kostki jednostkowej zanurzonej w E_p . Uwzględniając (3) oraz (4) mamy

$$w_{pj} = \frac{y_{pj}}{y_{p0}} = \frac{\sum_{k=1}^{L_p} c_{pj}^k}{L_p}, \quad j = 1, \dots, n \quad (14)$$

Wartości w_{pj} stanowią średnie arytmetyczne wartości c_{pj} - współrzędnych punktów w E_p odpowiadających macierzom $C_p^1, \dots, C_p^{L_p}$. Punkty te tworzą zbiór odpowiadający p -tej klasie symboli, punkt S_p można więc uważać za środek rozkładu. Położenie punktu S_p jest modyfikowane po każdej operacji uczącej symbolu klasy p . Analogiczne zależności zachodzą dla dowolnej klasy $i \in \{1, \dots, m\}$.

W okresie rozpoznawania są obliczane współczynniki p_i określone jako współczynnik bliskości między C a W_i (wzór (8)).

Przekształćmy wyrażenie (8)

$$\begin{aligned} p_1(C, W_1) &= \frac{1}{n} \sum_j [c_j w_{1j} + (1 - c_j)(1 - w_{1j})] \\ &= \frac{1}{n} \sum_j [1 + 2c_j w_{1j} - (c_j + w_{1j})] \\ &= 1 - \frac{1}{n} \sum_j [(c_j + w_{1j}) - 2c_j w_{1j}] \end{aligned}$$

Przy założeniu, że $c_j \in \{0, 1\}$ mamy równość

$$c_j + w_{1j} - 2c_j w_{1j} = |c_j - w_{1j}|$$

a więc

$$p_1(C, W_1) = 1 - \frac{1}{n} \sum_j |c_j - w_{1j}|$$

I ostatecznie

$$p_1(C, W_1) = 1 - \frac{1}{n} \rho_1(C, W_1) \quad (15)$$

gdzie

$$\rho_1(C, W_1) = \sum_j |c_j - w_{1j}| \quad (15a)$$

Określenie ρ_1 pokrywa się z definicją miary odległości w liniowej przestrzeni unormowanej l_n^1 .¹⁾

$$\rho(X, Y) = \|X - Y\|$$

jeśli norma jest określona wzorem

$$\|X\| = \sum_{i=1}^n |x_i| \quad (16)$$

Tak więc współczynnik bliskości jest liniową (malejącą) funkcją odległości ρ w przestrzeni l_n^1 .

Wymiar przestrzeni n' (wzór (8)) zależy - zgodnie z warunkiem (8b) - od przyjętych wartości parametrów gd oraz gg i określa ilość elementów macierzy C (ilość cech) biorących udział przy obliczaniu p_1 . Zmienia się on wraz ze zmianą klasy symboli

$$\begin{aligned} n' &= f(gg, gd, i), \quad i - \text{numer klasy} \\ 0 &\leq n' \leq n \\ n' &= 0, \quad \text{jeśli} \quad gg > 1 \quad \text{oraz} \quad gd < 0 \\ n' &= n, \quad \text{jeśli} \quad gd = gg = 0,5 \end{aligned}$$

Określeniu p_1 , $i = 1, \dots, m$ odpowiada więc badanie odległości między punktem - obrazem badanym a środkami rozkładu zbiorów - obrazami wzorcowymi poszczególnych klas. Proces ten jednak nie odbywa się w n -wymiarowej przestrzeni E_p , lecz w pewnych podprzestrzeniach tej przestrzeni, które powstają przez redukcję określo-

¹⁾ Liniową unormowaną przestrzeń n -wymiarową oznacza się symbolem l_n^p , gdzie p jest wykładnikiem potęgowym we wzorze na odległość

$$\rho = \left\{ \sum |x_i - y_i|^p \right\}^{1/p}$$

W naszych rozważaniach $n = n'$ oraz $p = 1$.

nych współrzędnych odpowiednio dla każdej klasy symboli.
 Udowodnimy teraz relację (9) podaną w pktcie 4.2.

$$\bigwedge_{i \in \{1, \dots, m\}} (0 \leq p_i(C, w_1) \leq 1)$$

Weźmy pewne i ze zbioru $1, \dots, m$. Uwzględniając że przy dowolnym j , $o_j \in \{0, 1\}$ oraz $0 \leq w_{1j} \leq 1$ z (15a) i (8b), wynika, że

$$0 \leq \rho_i \leq n' \quad (16)$$

Biorąc pod uwagę (15) mamy bezpośrednio

$$0 \leq p_i(C, w_1) \leq 1 \quad (17)$$

Ponieważ (17) jest słuszne dla dowolnego $i \in \{1, \dots, m\}$, więc dowodzi to relację (9).

$p_i = 0$, gdy $\rho = n$, tj. gdy punkty - obraz badany i obraz wzorcowy - są przeciwległymi wierzchołkami jednostkowej kostki k' -wymiarowej (największa możliwa odległość).

$p_i = 1$, gdy punkty te pokrywają się.

Rozpoznawanie polega na odszukaniu takiego środka rozkładu S_r , $r \in \{1, \dots, m\}$, do którego $\rho_i = \rho_r$ osiąga minimum (p_i - maksimum)

$$\rho_r = \min_i \rho_i$$

Symbolowi rozpoznawanemu przypisuje się wtedy nazwę r -tej klasy (przy uwzględnieniu dodatkowego warunku (11)).

5. Opis programu modelującego urządzenie rozpoznające na maszynie cyfrowej oraz wyniki badań modelu

Na podstawie opisanej koncepcji ułożono program modelujący urządzenie rozpoznające na maszynie cyfrowej. Program został napisany w języku Gier-Algol III [21] na maszynę cyfrową GIER i zawierał ok. 650 linii algowakich (ponad 2000 instrukcji maszynowych). Program jest umieszczony na końcu niniejszej pracy.

Uproszczony schemat operacyjny programu przedstawiono na rys.19

5.1. Opis przyjętych oznaczeń w schemacie operacyjnym

Komentarz

- komentarz



- tekst wydrukowany przez maszynę na monitorze
(maszynie do pisania)



- opis operacji lub instrukcja bezwarunkowa



- etykiety

a

- sterowanie z monitora ¹⁾ (należy nacisnąć na monitorze klawisz oznaczony literą umieszczoną w kwadracie - w tym przypadku literę a)

tak	nie

- instrukcja warunkowa

kbon

- klucz kb na pulpicie sterującym maszyną włączony (umożliwia on wykonanie skoku w programie ([21] str. 158)

5.2. Opis najważniejszych zmiennych i macierzy używanych w programie

Zmienne:

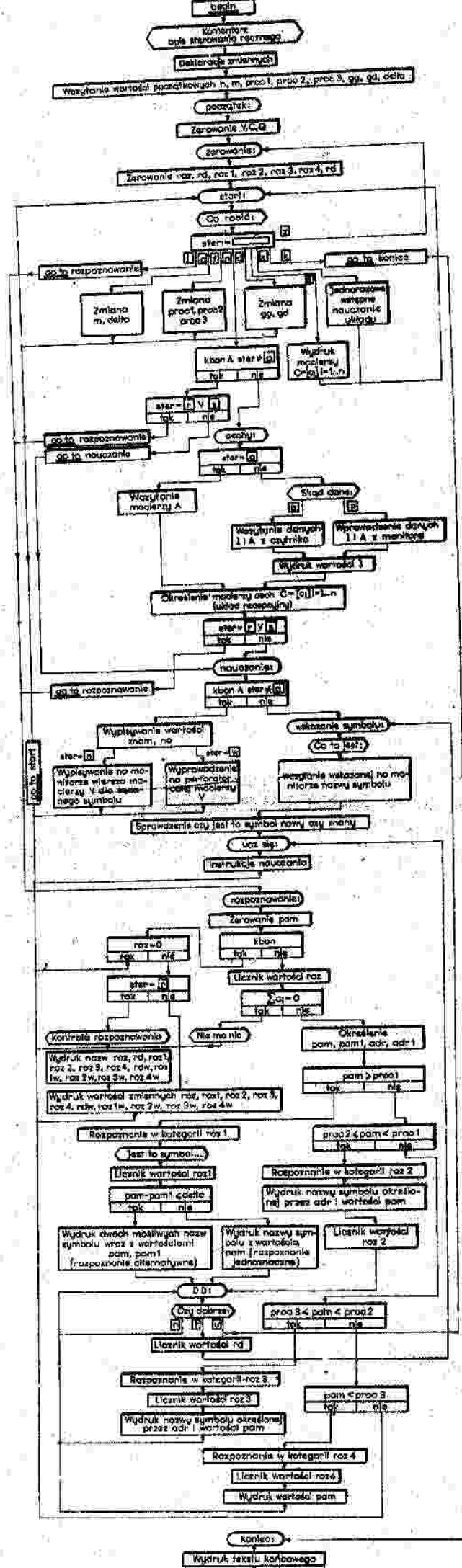
n - ilość cech (ilość elementów w macierzy C)

m - ilość obiektów do rozpoznawania ($m_{maks} \approx 130$)

i - numer porządkowy kolejnego znaku dostarczanego maszynie

ster - zmienna przyjmująca różną wartość w zależności od sterowania ręcznego (pkt 5.3) i sterująca działaniem całego programu

¹⁾ Oznaczenia tego używany także w tekście opisującym działanie maszyny.



na - całkowita ilość operacji uczących
 znam - ilość nauczonych symboli (znam = m)
 adr - adres rozpoznanego symbolu (nr porządkowy w tablicy Y)
 adr 1 - adres symbolu odpowiadającego pam 1
 roz - całkowita ilość operacji rozpoznawania
 rd - ilość rozpoznań ocenionych przez operatora jako poprawne
 (sob. str. 64)
 roz 1 - ilość rozpoznań w kategorii roz 1 (sob. pkt 5.3 - działanie 4)
 roz 2 - ilość rozpoznań w kategorii roz 2
 roz 3 - ilość rozpoznań w kategorii roz 3
 roz 4 - ilość rozpoznań w kategorii roz 4
 pum - bieżąca wartość współczynnika bliskości $p_1(G, Y_1)$
 (wzór (10))
 pam - maksymalna wartość współczynnika bliskości p_{maks}
 pam 1 - p_{maks}
 proc 1 - dolna granica rozpoznawania w kategorii roz 1
 proc 2 - dolna granica rozpoznawania w kategorii roz 2
 proc 3 - dolna granica rozpoznawania w kategorii roz 3
 gg - próg górny (sob. pkt 4.2)
 gd - próg dolny
 delta - maksymalna wartość różnicy pam - pam 1, przy której zachodzi rozpoznawanie alternatywne (delta = Δ_0)

Macierze:

A - skwantowany obraz fizyczny $A = [a_{ij}]_{i=1, \dots, k; j=1, \dots, l}$
 (sob. rys. 20)

G - macierz cech $G = [c_1, c_2, \dots, c_n]$

Y - macierz stanu pamięci maszyny

$$Y = \begin{bmatrix} y_{1,0} & y_{1,1} & \dots & y_{1,n-1} & y_{1,n} \\ \vdots & \vdots & & \vdots & \vdots \\ y_{m,0} & y_{m,1} & \dots & y_{m,n-1} & y_{m,n} \end{bmatrix}$$

Przez dodanie do każdego wiersza macierzy Y odpowiedniej nazwy klasy symboli oraz liczby porządkowej klasy otrzymujemy tabl. Y

(wszór (1)). W programie przewidziano możliwość wyprowadzenia tabl. Y jako jednego z działań maszyny (pkt 5.3 - działanie 14).

5.3. Opis działań maszyny

Praca maszyny rozpoczyna się od wczytania zadanych wartości zmiennych a , m , proc 1, proc 2, proc 3, gg, gd, delta. Następnie odbywa się zerowanie zmiennych roz, rd, roz 1, roz 2, roz 3, roz 4 oraz macierzy Y, C i pewnej pomocniczej macierzy Q.

Po wykonaniu tych operacji następuje wydruk na monitorze tekstu "Co robić" i maszyna zatrzymuje się. Dalsze działania maszyny zależą od operatora, który może sterować maszyną przez naciśnięcie odpowiedniego klawisza na monitorze oraz przez włączenie klucza kb na pulpicie sterującym. Poniżej podano opis sterowania i odpowiednich działań maszyny.

1. **[R]** - Nauczanie. Maszyna rozpoczyna działania związane z procesem uczenia. Następuje wydruk tekstu "Skąd dane:". Możliwe są dwie odpowiedzi: **[C]** - następuje wczytanie danych 1 i A z czytnika taśmy dziurkowanej. **[P]** - dane 1 i A wprowadza się z monitora. Po wczytaniu danych maszyna drukuje numer porządkowy danych 1, a następnie pyta: "Co to jest:". Nazwę symbolu pisze się na monitorze kończąc ją przecinkiem, karetką lub średnikiem (maszyna wczytuje nazwy nie dłuższe niż siedmioliterowe). Po wczytaniu nazwy zostają wykonane instrukcje uczenia.
2. **kben** **[R]** - Rozpoczyna się nauczanie automatyczne. Maszyna wczytuje z czytnika pierwszy zestaw danych 1, A wraz z nazwą symbolu i drukuje wczytaną wartość 1. Następnie wykonuje instrukcje uczenia układu, a po ich wykonaniu samoczynnie wczytuje następny zestaw danych i powtarza te same czynności od początku. Proces automatycznego nauczania można przerwać wyłączając klucz kb. Następuje wtedy skok do etykiety "start" i wydruk tekstu "Co robić:". Proces ten przerywa się także samoczynnie, jeśli na końcu danego zestawu danych znajduje się średnik.
3. **[Z]** - Następuje jednorazowe wstępne nauczanie całego układu (organizacja wstępna). Zostaje wczytana wartość m , z_{nm} oraz tablica Y zapisana na uprzednio przygotowanej taś-

nie. Taśmę tę można uzyskać ucząc najpierw maszynę jed-
nym z dwu wymienionych powyżej sposobów, a następnie wy-
prowadzając na perforator wartości na, znam oraz tab-
licę Y (zob. działanie 14). Przy wczytywaniu taśmy na-
stępuje wydruk kontrolny: a) kolejnej liczby porządkowej
symbolu w tablicy Y oraz nazwy symbolu, jeśli pro-
gram był wprowadzony przez O (uwzględnienie odcinka
programu między punch off a punch on) ([21] str.203
b) tylko ostatniej liczby porządkowej symbolu i jego na-
zwy, jeśli program był wprowadzony inaczej.

4. **[F]** - Rozpoczęcie procesu rozpoznawania: a) wczytanie danych
tak jak przy **[H]**, b) wykonanie instrukcji rozpoznawa-
nia. Przewidziano następujące kategorie rozpoznawania:
roz1 - gdy $pam > proc1$. Rozpoznanie w tej kategorii
zaopatruje się w komentarz: « Jest to symbol
"..." p = (wartość pam) ». Jeśli $pam - pam1$
≤ delta, następuje rozpoznanie alternatywne i
wydruk dodatkowo tekstu « lub symbol "..." p =
= (wartość pam) ».
- roz2 - gdy $proc2 \leq pam < proc1$. Zostaje wydrukowany
tekst: « roz2 (symbol) p = (pam) ». Rozpoznanie
w tej kategorii oznacza, że jest stosunkowo duża
możliwość pomyłki i można je scharakteryzować ko-
mentarzem: "Przypomina mi symbol ...".
- roz3 - gdy $proc3 \leq pam < proc2$. Zostaje wydrukowany
tekst: « roz3 (symbol), p = (pam) ». Wartość pam
jest zbyt mała, żeby można ją przyjąć jako miaro-
dajną. Rozpoznanie można skomentować: "nie mogę
dostatecznie dokładnie rozpoznać tego symbolu".
- roz4 - gdy $pam < proc3$. Zostaje wydrukowany tekst: roz4
p = (wartość pam). Przyjmuje się tu, że układ
nie był uczony tego symbolu. Komentarz: "nie znam
tego symbolu".

Ilości odpowiedzi maszyny w każdej z czterech kategorii
są zliczane; określają je zmienne roz1, roz2, roz3 oraz
roz4. Następnie maszyna pyta: "Czy dobrze". Są możliwe
odpowiedzi: a) **[E]** - jeśli odpowiedź jest prawidłowa; na-
stępuje wówczas proces uczenia się tego symbolu i wzrost

wartości zmiennej rd o 1; b) $\boxed{u}$ - jeśli odpowiedź jest błędna, ale chcemy nauczyć układ poprawnej odpowiedzi; maszyna pyta: "Co to jest:"; po wczytaniu podanej nazwy symbolu są wykonywane instrukcje uczenia; c) $\boxed{n}$ (albo dowolna inna litera oprócz u oraz t) - jeśli odpowiedź jest błędna i nie chcemy skorzystać z możliwości b). Po otrzymaniu jednej z powyższych odpowiedzi następuje skok do etykiety "start:" i wydruk tekstu "Co robić:".

5. $\boxed{j}$ - Powtórne rozpoznawanie przy tej samej macierzy C.
6. $\boxed{e}$ - Możliwość wprowadzenia z monitora nowych wartości m i delta.
7. $\boxed{d}$ - To samo co powyżej odnośnie do gg oraz gd.
8. $\boxed{f}$ - To samo co powyżej odnośnie do proc1, proc2 i proc3.
9. $\boxed{a}$ - Wyprowadzenie na monitor aktualnej macierzy C.
10. $\boxed{s}$ - Zerowanie roz, rd, roz1, roz2, roz3, roz4.
11. $\boxed{k}$ - Wydruk tekstu kończącego i zakończenie działań maszyny.
12. kben $\wedge$ $\boxed{l}$ - Wydruk na monitorze wartości znam oraz na.
13. kben $\wedge$ $\boxed{n}$ - To samo co wyżej oraz wydruk wiersza tablicy Y dla symbolu odpowiadającego klawiszowi naciśniętemu po wydrukowaniu przez monitor tekstu: "Y dla:".
14. kben $\wedge$ $\boxed{w}$ - To samo co w pktcie 12 oraz wyprowadzenie na perforator wartości na, znam oraz całej tablicy Y.
15. kben $\wedge$ $\boxed{m}$ - To samo co w pktcie 14 z tym, że wyprowadzona zostanie tablica Y_n , której elementy są określone następująco:

$$y_{jn}^1 = y_j^1 / y_0^1 \times 100$$
16. kben $\wedge$ $\boxed{x}$ - Wydruk w jednym wierszu tekstu z nazwami zmiennych roz, rd, roz1, roz2, roz3, roz4, rdw, roz 1w, roz 2w, roz 3w, roz 4w a pod nimi odpowiednich ich wartości, przy czym rdw = rd/roz, roz 1w = roz1/roz, roz2w = roz2/roz, roz 3w = roz3/roz, roz 4w = roz4/roz.
17. kben $\wedge$ $\boxed{s}$ - To samo co w pktcie 16, lecz bez wydruku nazw zmiennych.

Po wykonaniu każdego z działań 1. - 17. następuje skok do etykiety "start:" i wydruk tekstu "Co robić:". Skok ten może nastąpić także w toku wykonywania działań 13. - 15., jeśli zostanie wyłączony klucz kb.

5.4. Wyniki badań

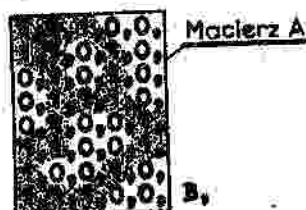
5.4.1. Opis danych do badań

Do badań przygotowano łącznie ok. 900 różnych symboli - dużych i małych liter łacińskich (480 + 270), cyfr (135), małych liter greckich (47), niektórych liter rosyjskich (44) i pewne symbole specjalne (12).

Rys. 20

Przykład zapisu symbolu przygotowanego do badań

2.



Po ręcznym przekształceniu w zero-jedynkową macierz A symbole były zapisywane na 8-kanalowej taśmie dziurkowanej. Sposób zapisu jednego symbolu był następujący: numer porządkowy symbolu, macierz A, nazwa symbolu (rys. 20).

Tak przygotowana taśma z symbolami służyła zarówno do uczenia (proste i automatyczne), jak i do rozpoznawania.

5.4.2. Uczenie modelu

Pojedyncza operacja ucząca przy uczeniu ręcznym (łącznie z czasem wypisania nazwy symbolu na monitorze) trwała 10-13 sek. Przy uczeniu automatycznym - średnio 7 sek. Czas określania macierzy C - od 5 do 6 sek.

W okresie uczenia stosowano przede wszystkim uczenie automatyczne, przy czym po każdym określonym cyklu uczenia (np. po nauczaniu kolejnego alfabetu liter) wyprowadzono tablicę Y z pamięci maszyny na dziurkarkę.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

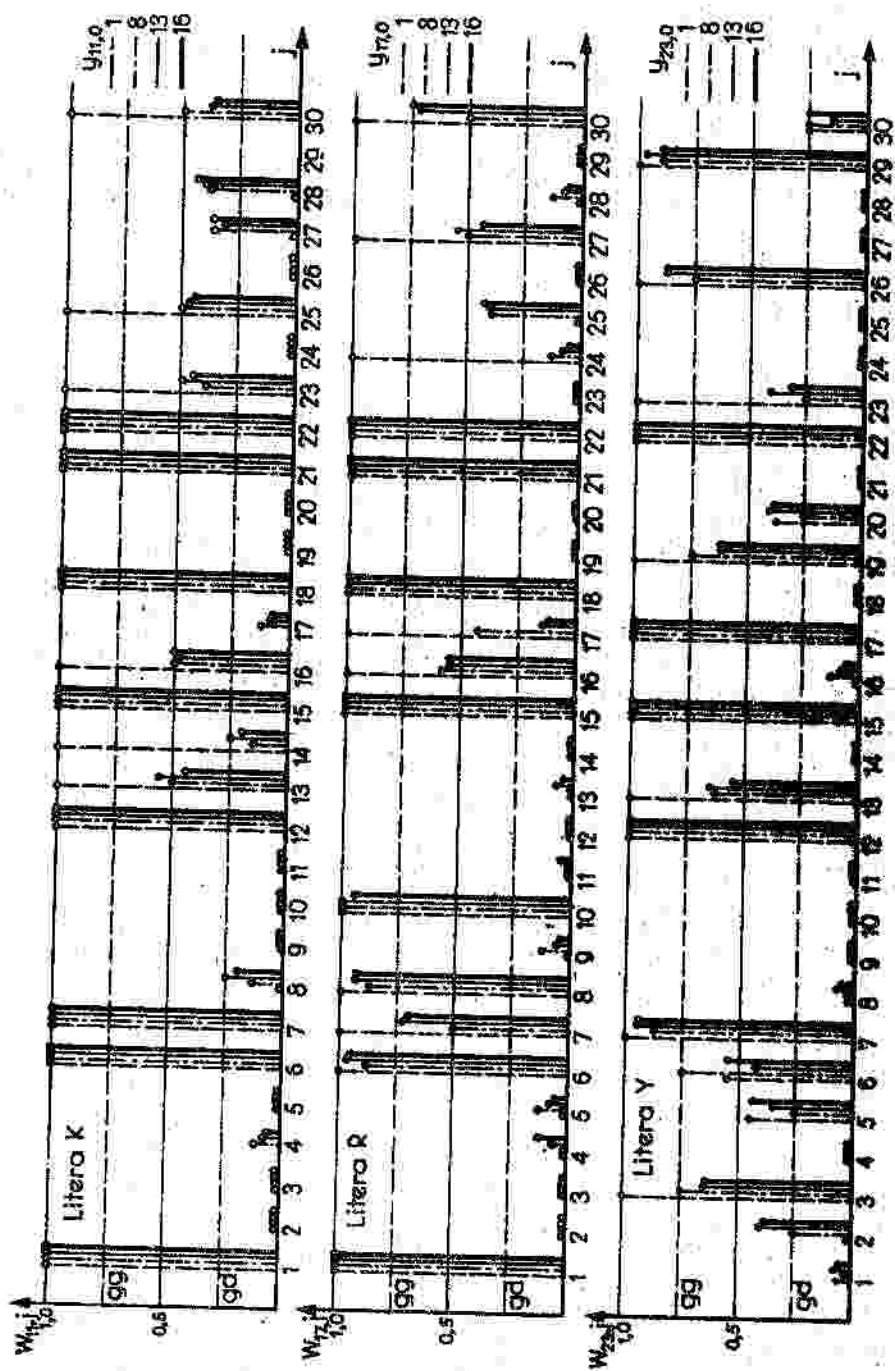
Tablica 7 dla alfabetu innych liter łaciniejszych
po jednokrotnym nauczeniu każdego symbolu (zestaw danych nr 13)

[illegible]

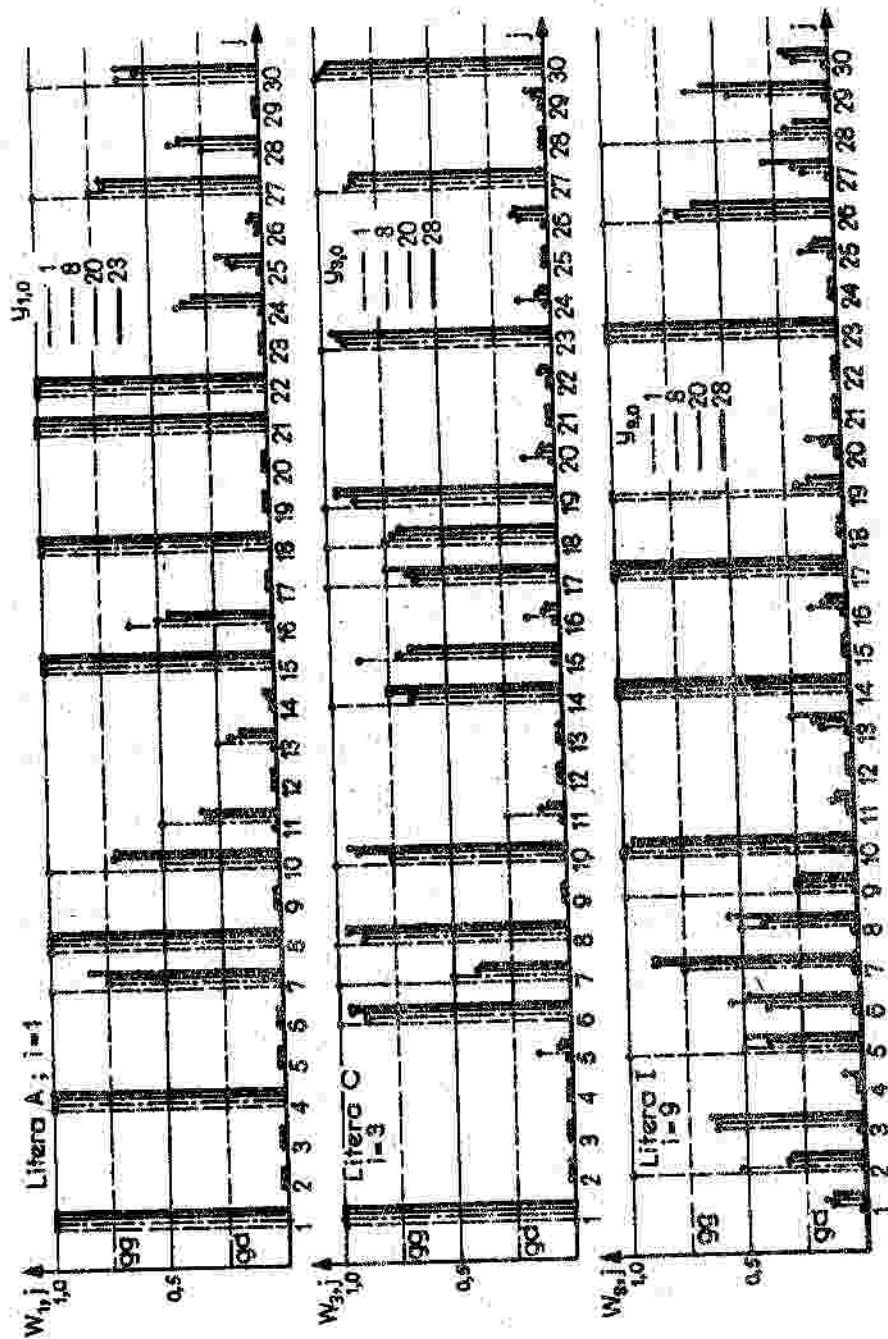
Z tablica Y dla alfabetu dźwięcznych liter łacińskich
po wielokrotnym nauczeniu każdego symbolu

14	7	6	10	7	14	5	11	3	8	11	8	12	3	4	8	0	4	8	3	4	6	2	0	5		
0	6	2	0	0	6	0	8	13	0	9	0	2	3	1	8	1	2	0	6	0	0	0	0	0		
0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	5	0	0	0	7	4	0		
14	6	7	7	9	2	8	12	12	0	8	5	12	9	5	9	10	6	2	6	7	5	5	6	5		
0	0	1	6	0	12	0	0	13	0	0	0	2	15	0	0	0	3	0	0	0	1	1	0	0		
0	1	0	0	0	5	0	0	0	0	0	0	0	0	0	0	3	0	0	0	2	0	0	3	3		
14	3	2	0	1	0	0	5	0	2	8	11	13	1	3	5	2	1	2	1	0	0	0	0	1		
0	0	2	1	0	5	0	4	14	11	0	0	0	15	0	5	0	6	5	0	0	1	2	0	0		
4	13	8	13	12	15	14	0	0	11	2	1	10	5	4	11	15	1	4	9	9	9	9	10	9		
11	0	0	1	0	0	0	14	0	0	12	1	13	0	0	2	0	0	0	0	0	0	0	0	0		
0	0	0	0	0	14	0	0	0	0	0	0	0	0	16	2	0	7	0	0	7	0	1	9	0		
3	13	8	13	0	15	14	13	0	0	10	0	1	12	15			2	9	9	2	9	8	0	10	8	
14	13	7	12	0	15	14	11	13	13	0	13	10	0	5	11	13	1	1	9	0	7	9	0	10	9	
0	0	8	0	10	13	9	0	10	5	6	0	0	16	0	8	12	8	9	9	9	8	0	8	0	8	
0	0	0	2	0	1	7	5	0	0	7	1	0	0	0	0	0	0	1	0	0	0	1	1	1	0	
14	4	2	0	13	6	15	5	0	0	9	6	13	1	16	3	15	5	9	8	9	0	6	1	10	9	
0	10	3	13	0	9	0	10	7	8	0	0	0	2	3	8	0	4	2	5	5	9	8	7	0	0	
4	0	1	0	1	0	0	5	0	0	8	1	1	7	1	2	3	1	0	0	0	0	2	2	0	0	
0	0	0	0	0	0	0	0	0	2	6	0	0	0	6	0	0	0	0	0	5	0	0	0	0	0	
4	13	0	13	1	8	0	13	14	13	10	0	1	0	5	11	0	7	1	1	6	0	2	0	0	0	
9	0	0	0	12	6	15	0	0	0	7	12	0	8	0	1	0	2	8	8	0	9	7	9	10	9	
0	0	0	0	12	2	9	0	0	1	0	0	0	0	0	1	0	0	9	9	1	9	9	0	10	8	
13	13	8	11	1	6	4	0	1	1	3	1	1	10	0	5	15	4	0	0	0	0	8	0	1	1	
12	7	5	9	3	11	3	10	0	5	7	13	0	4	1	10	9	9	0	0	2	0	4	2	5	2	
1	0	8	1	13	4	1	0	0	11	12	0	0	0	0	5	0	0	9	7	1	9	9	1	8	0	
0	9	0	0	0	9	0	7	14	0	0	0	2	4	16	0	7	1	2	0	0	0	0	6	0	1	
14	4	0	12	0	0	15	7	0	1	2	11	6	0	5	5	0	4	0	6	9	0	0	2	9	8	
0	0	0	0	0	4	8	0	13	0	0	0	0	0	1	0	15	1	9	5	0	9	0	8	10	9	
0	0	2	1	0	5	3	0	0	13	0	0	0	5	4	5	2	0	8	0	5	0	2	0	0	0	
14	13	6	2	13	8	4	0	0	13	13	7	1	2	10	1			0	0	0	0	0	0	0	0	
14	13	8	13	13	13	13	13	13	13	13	13	13	13	13	13	13	9	9	9	9	9	9	9	9	9	
x1(a)	x2(b)	x3(c)	x4(d)	x5(e)	x6(f)	x7(g)	x8(h)	x9(i)	x10(j)	x11(k)	x12(m)	x13(n)	x14(o)	x15(r)	x16(s)	x17(t)	x18(y)	x19(jede)	x20(dva)	x21(tray)	x22(cste)	x23(plec)	x24(ases)	x25(sied)	x26(rale)	x27(date)

Tablica Y dla małych liter łacińskich i cyfr



Rys. 21. Ustalenie się wag w_{ij} dla liter A, C, I



Rys. 22. Ustalenie się wag w_{ij} dla liter K, R, Y

Powyżej przedstawiono wydruki tablic Y dla różnych alfabetów używanych w badaniach. W przypadku alfabetu dużych liter łacińskich przykłady tablic pokazano przy długości ciągu uczącego każdego symbolu równej 1 (tablica Y na str. 38) oraz kilkanaście lub więcej (tablica Y na str. 39).

Przed każdą tablicą Y jest umieszczona wartość zmiennej $z_{n\alpha}$ oraz n .

Po uzyskaniu taśm z tablicami Y dla różnych alfabetów i przy różnych wartościach y_{10} w następnych badaniach taśmy te wykorzystywano do wprowadzenia dowolnej organizacji wstępnej modelu. Umożliwiało to uniknięcie kłopotliwego uczenia układu przed każdym cyklem badań.

Na rysunkach 21 i 22 przedstawiono - uzyskane na podstawie tablic Y - wykresy ustalania się wartości wag cech w_{1j} , $j = 1, \dots, 30$, dla niektórych symboli (tj. przy różnych wartościach i) wraz ze wzrostem długości ciągów uczących y_{10} .

5.4.3. Ustalenie wartości g_g oraz g_d

W celu ustalenia optymalnych wartości g_g oraz g_d nauczonemu modelowi (tablica Y na str. 39) przedstawiono kolejno wybrane symbole do rozpoznawania i dla każdego symbolu badano odpowiedzi przy różnych wartościach g_g oraz g_d . W pierwszej fazie wyróżniono dwa rodzaje badań:

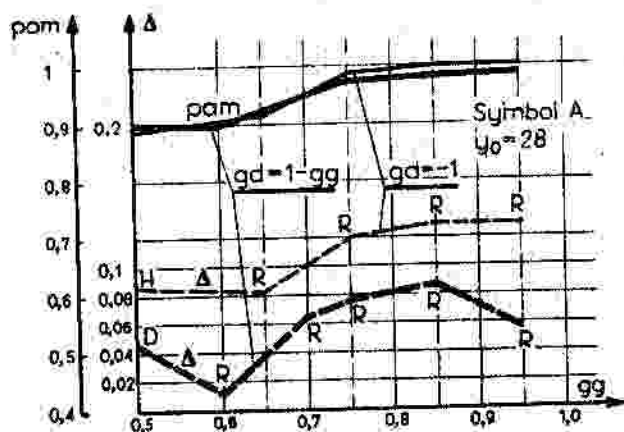
1) g_g oraz g_d przyjmują wartości symetryczne względem 0,5, tj. przy zmiennej wartości g_g jest $g_d = 1 - g_g$; ten rodzaj badań nazywa się dalej badaniami przy progach symetrycznych;

2) zmienia się tylko wartość g_g , natomiast $g_d = -1$ ¹⁾; w tym przypadku przy obliczaniu p_1 uwzględnia się tylko elementy ciągów y_i o wartościach względnych $y_{1j}/y_{10} = w_{1j} > g_g$.

Tego rodzaju rozpoznawanie ma charakter rozpoznawania wyłącznie przez stwierdzenie obecności w obiekcie cech, określonych warunkiem $w_{1j} \geq g_g$, jako istotnych dla danej klasy. Badania te nazywa się dalej badaniami przy progu jednostronnym.

W drugiej fazie stosowano wyłącznie badania przy progach symetrycznych.

1) Parametr g_d może tu przyjmować dowolną wartość, byleby spełniała nierówność $g_d < 0$.



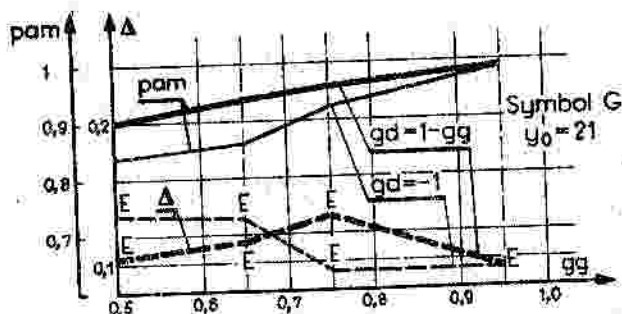
Rys. 23

W badaniach przyjęto $\Delta = 1$, aby każde rozpoznanie miało charakter alternatywny, dzięki czemu za każdym razem był określony także symbol uznany przez model za najbardziej podobny do właściwego, tj. odpowiadający $pam\ 1$.

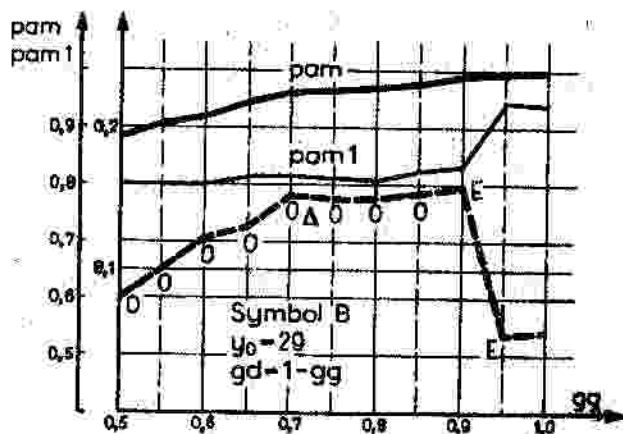
Na rys. 23 do 28 przedstawiono przykładowo niektóre wyniki powyższych badań przy rozpoznawaniu wybranych dużych liter alfabetu łacińskiego.

Na rys. 23 i 24 przedstawiono wykresy zależności $pam = f(gg)$ oraz $\Delta = f(gg)$ dla obu przypadków zmian progów ($gd = 1 - gg$ oraz $gd = -1$) przy rozpoznawaniu liter A oraz G, przy czym

$$\Delta = pam - pam\ 1$$



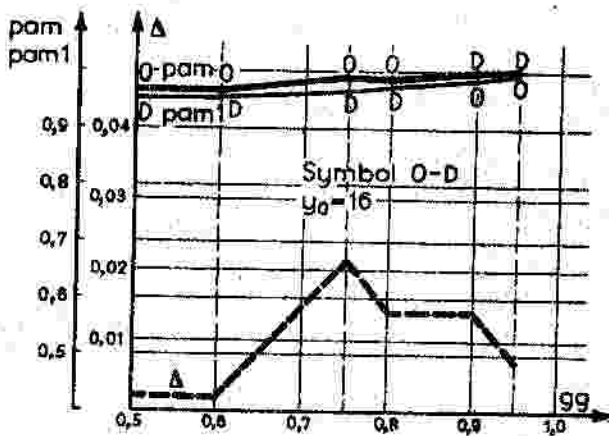
Rys. 24



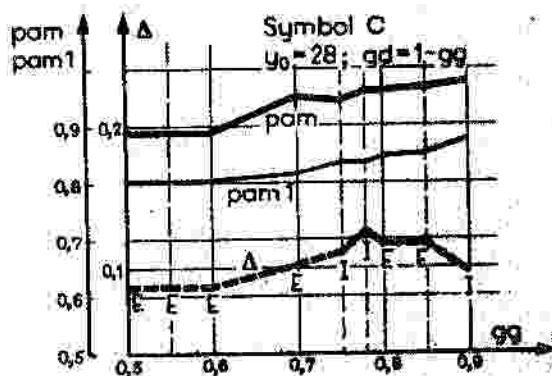
Rys. 25

Obok każdej wartości Δ podano nazwę symbolu odpowiadającego pam 1. Wartością y_0 oznaczono długość ciągu uczącego danego symbolu przed rozpoczęciem powyższych badań.

Na rys. 25 do 28 przedstawiono wykresy zależności $pam = f(gg)$, $pam 1 = f(gg)$ oraz $\Delta = f(gg)$ uzyskane w badaniach przy prógach symetrycznych.

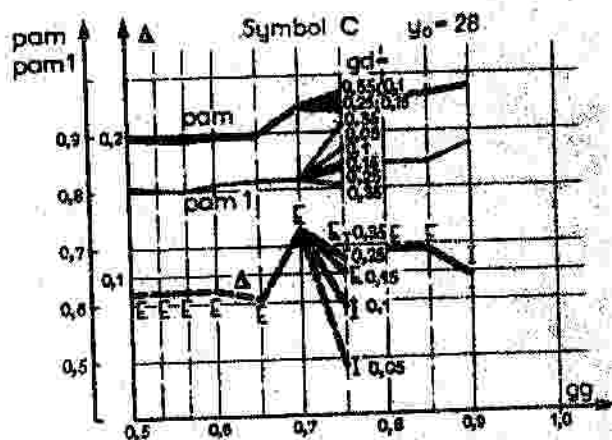


Rys. 26



Rys. 27

Na rys. 28 przedstawiono wpływ zmiany progu gd na wartości pam oraz $pam1$ przy ustalonej wartości $gg = 0,75$.



Rys. 28

Badania wykazały (ogółem uzyskano około 35 wykresów tego typu), że rozpoznawanie jest poprawne w dużym zakresie wartości gg oraz gd zarówno w przypadku progów symetrycznych, jak i progu jednostronnego. W tym ostatnim przypadku obserwowano jednak większą

częstość błędów i dlatego dalsze badania prowadzono już wyłącznie przy progach symetrycznych.

Wynik ten można wytłumaczyć tym, że w badaniach przy progu jednostronnym zostaje pominięta informacja o tym, które cechy nie występują w określonych klasach. Wykresy $pam = f(gg)$ zarówno w przypadku $gd = 1 - gg$, jak i w przypadku $gd = -1$ mają charakter rosnący wraz ze wzrostem gg , co jest zrozumiałe ze względu na przyjętą definicję współczynnika bliskości $p_{maks} = pam$.

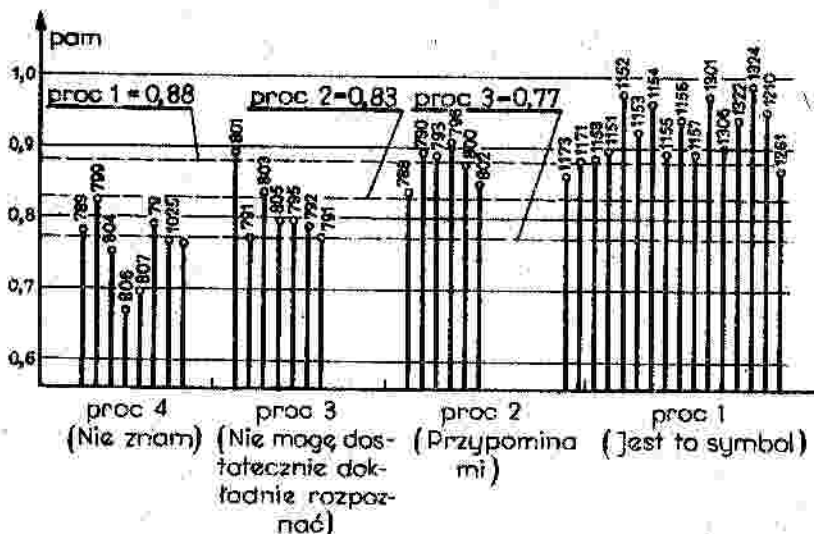
Traktując wartość Δ jako pewne kryterium pewności rozpoznawania stwierdzono, że optymalna wartość gg zawiera się w przedziale

$$0,7 \leq gg \leq 0,8$$

Do dalszych badań przyjęto wartości: $gg = 0,75$ i $gd = 0,25$.

5.4.4. Ustalenie wartości proc 1, proc 2 oraz proc 3

W celu orientacyjnego określenia wartości proc 1, proc 2 oraz proc 3 badano odpowiedzi modelu przy rozpoznawaniu symboli, które z góry zakwalifikowano do założonych kategorii roz 1, roz 2,



Rys. 29

roz 3 oraz roz 4 . Wyniki przedstawiono na rys. 29. Przyjęto następujące wartości:

$$\text{proc 1} = 0,88, \quad \text{proc 2} = 0,83, \quad \text{proc 3} = 0,77$$

5.4.5. Ustalenie wartości delta

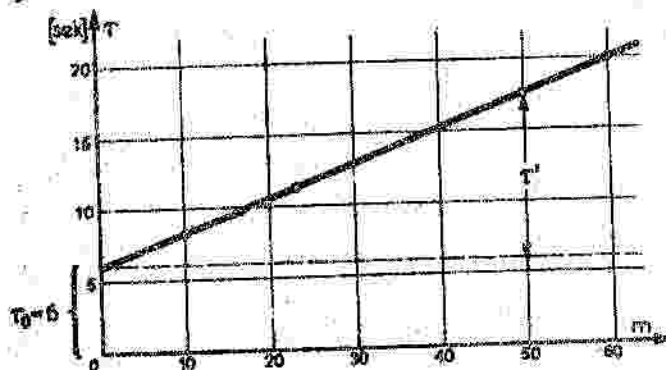
Ustalenie wartości delta opiera się na warunku, aby symbole uznawane przez model za bardzo podobne (mała wartość Δ) lub różniące się szczegółami, które nie mogą być wykryte wskutek małej rozróżnialności siatkówki (np. D i O, 5 i 8) były rozpoznawane alternatywnie. W tym celu badano wartość Δ dla szeregu symboli spełniających powyższy warunek (przede wszystkim O i D). Na podstawie wyników przedstawionych na rys. 36 do 41 przyjęto

$$\text{delta} = 0,022$$

5.4.6. Badania zdolności rozpoznawania

5.4.6.1. Czas rozpoznawania

Czas rozpoznawania τ zależał przede wszystkim od długości alfabetu (w programie zmienna znam , w opisie teoretycznym - m), którego model został nauczony. Wykres $\tau = f(m)$ przedstawiono na rys. 30.



Rys. 30

Czas określenia macierzy C (około 6 sekund) pokrywa się z wartością τ_0 uzyskaną przez ekstrapolację liniową wykresu $\tau =$

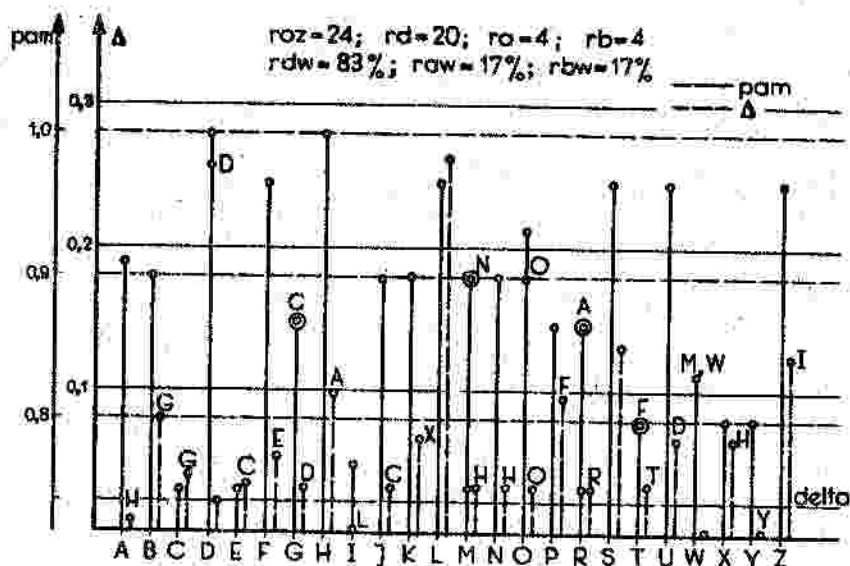
$= f(m)$. Czas $\tau = \tau - \tau_0$ można więc interpretować jako czas obliczania współczynników bliskości $p_i(C, Y_i)$, $i = 1, \dots, m$ w celu określenia p_{maks} . Ze względu na szeregowe obliczanie p_i liniowy przebieg $\tau' = f(m)$ jest w pełni zrozumiały.

5.4.6.2. Zależność poprawności rozpoznawania od długości ciągów uczących oraz inne badania

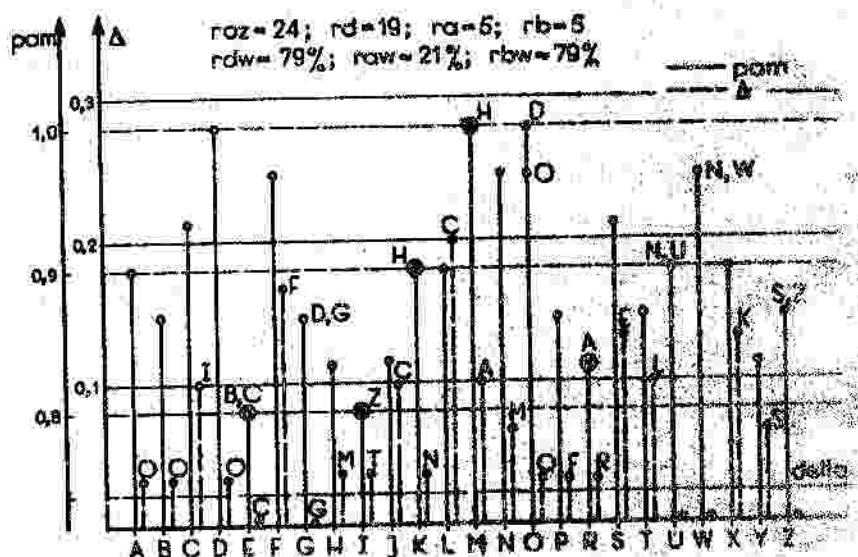
Badania przeprowadzono dla dużych liter alfabetu łacińskiego ($m = 24$). Badano odpowiedzi modelu uczonego ciągami uczącymi o różnych długościach y_0 przy rozpoznawaniu nowo dostarczonych alfabetów. Wartość y_0 oznacza średnią długość ciągu uczącego dla całego nauczonego alfabetu

$$y_0 = \frac{\sum_{i=1}^m y_{i0}}{m} = \frac{na}{m}$$

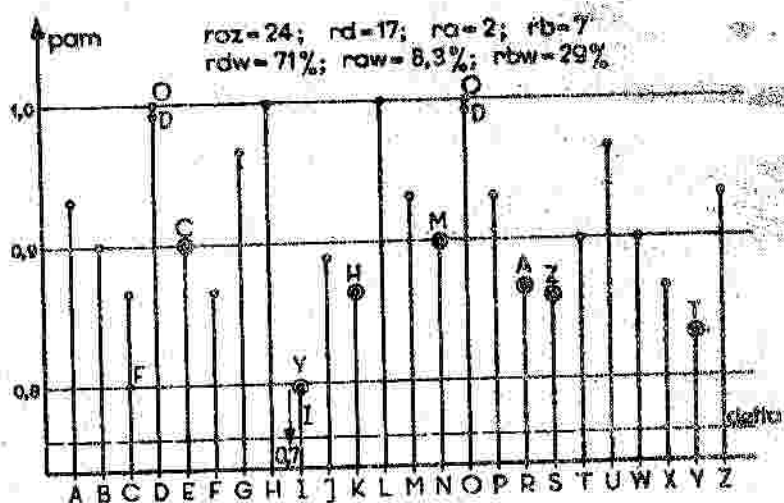
w danym cyklu badań.



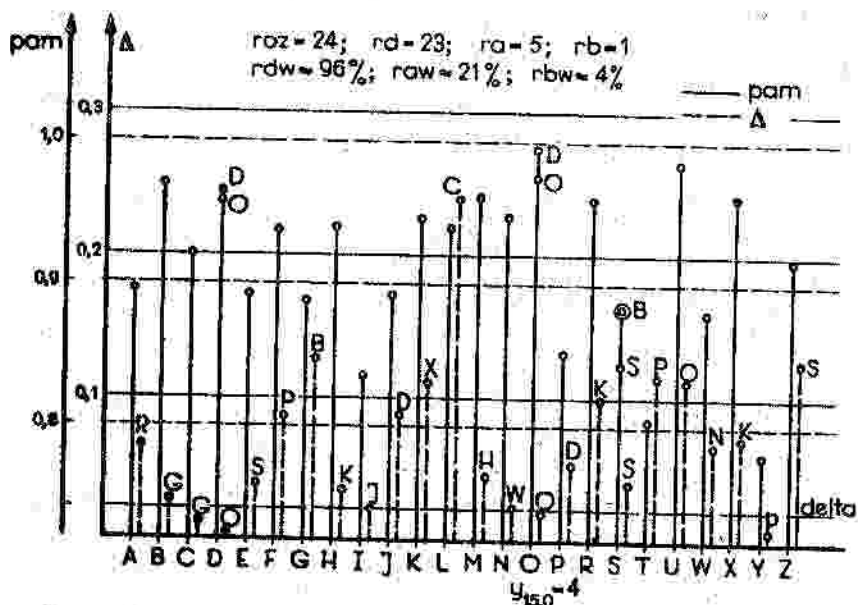
Rys. 31. Rozpoznawanie alfabetu dużych liter łacińskich
 $na = 24$, $y_{i0} = 1$, $i = 1, \dots, 24$, zestaw danych: 14 (1001-1024)



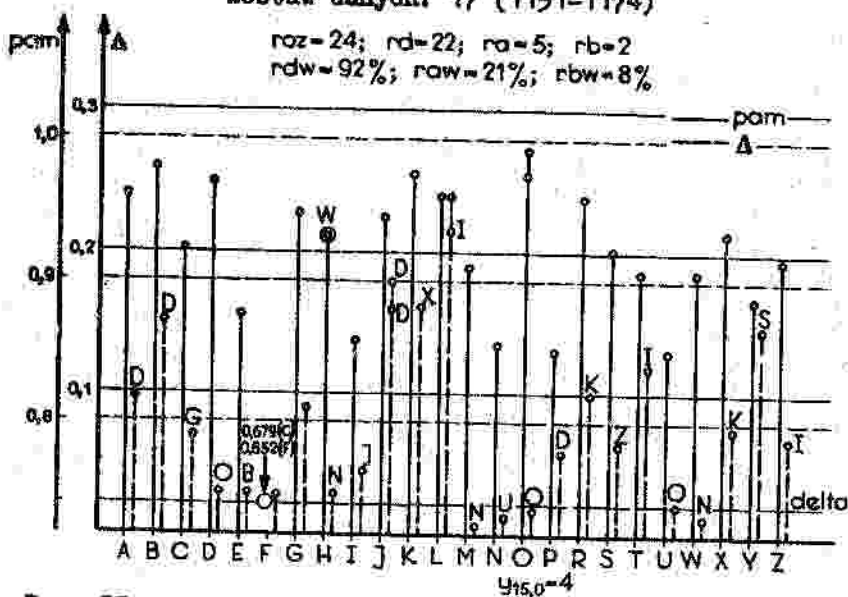
Rys. 32. Rozpoznawanie alfabetu dużych liter łacińskich
na = 24, $y_{10} = 1$, $i = 1$, ..., 24, zestaw danych: 15 (1051-1074)



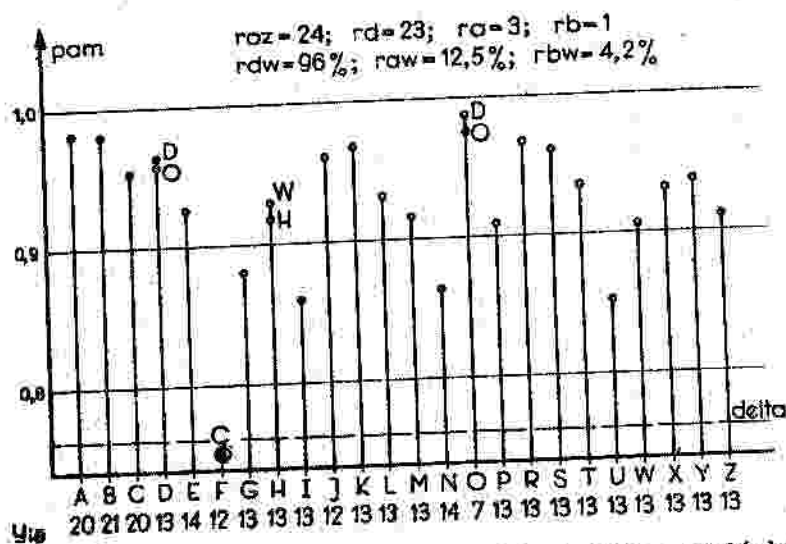
Rys. 33. Rozpoznawanie alfabetu dużych liter łacińskich
na = 24, $y_{10} = 1$, $i = 1$, ..., 24, zestaw danych: 21 (1351-1374)



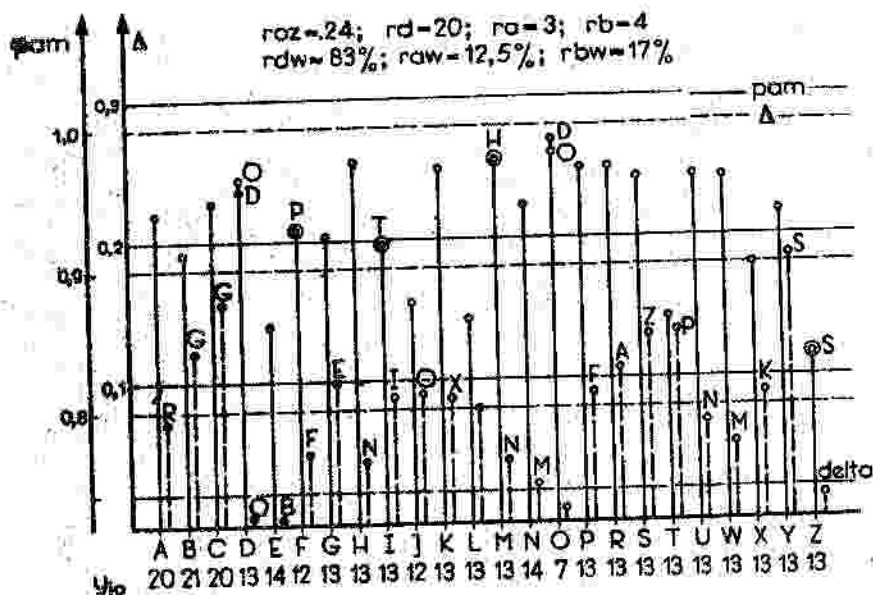
Rys. 34. Rozpoznawanie alfabetu dużych liter łacińskich
 $n_a = 189$, $y_{10} = 8$, $i = 1, \dots, 24$, $i \neq 15$, $y_0 \approx 8$,
 zestaw danych: 17 (1151-1174)



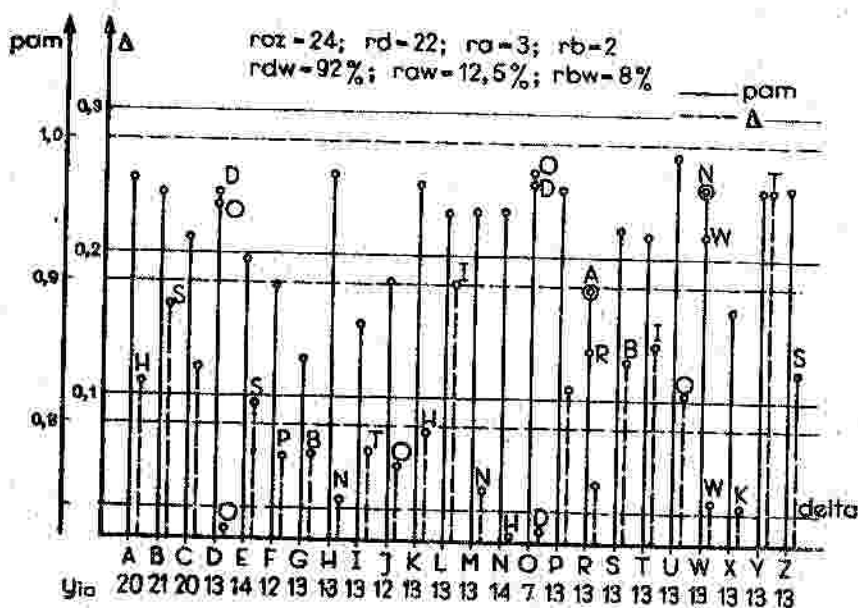
Rys. 35. Rozpoznawanie alfabetu dużych liter łacińskich
 $n_a = 189$, $y_{10} = 8$, $i = 1, \dots, 24$, $i \neq 15$, $y_0 \approx 8$,
 zestaw danych: 18 (1201-1224)



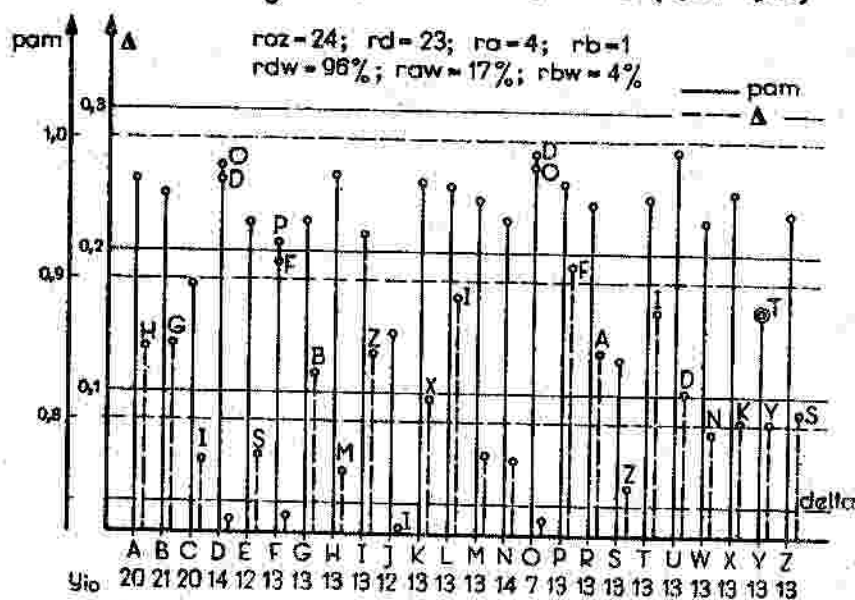
Rys. 36. Rozpoznawanie alfabetu dużych liter łacińskich
na = 325, $y_0 = 13$, zestaw danych: 18 (1201-1224)



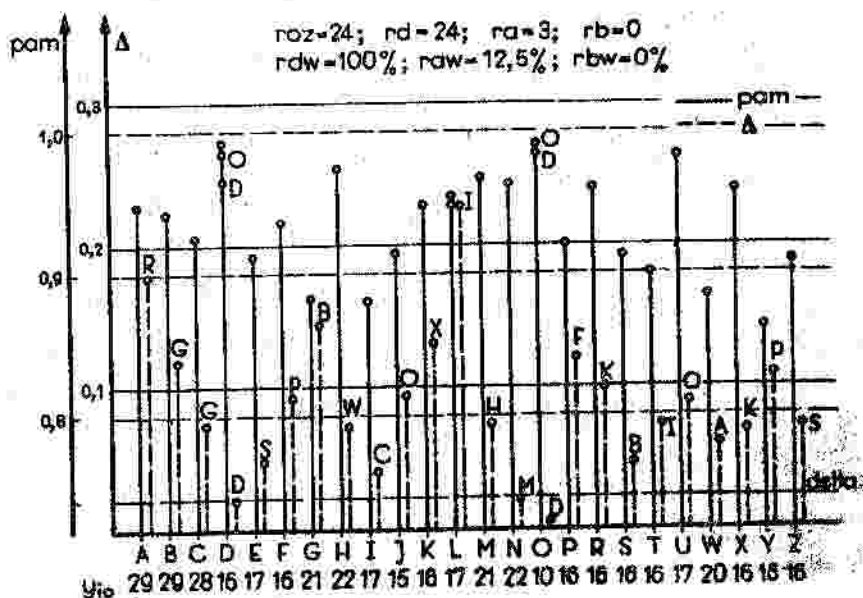
Rys. 37. Rozpoznawanie alfabetu dużych liter łacińskich
na = 325, $y_0 = 13$, zestaw danych: 19 (1251-1274)



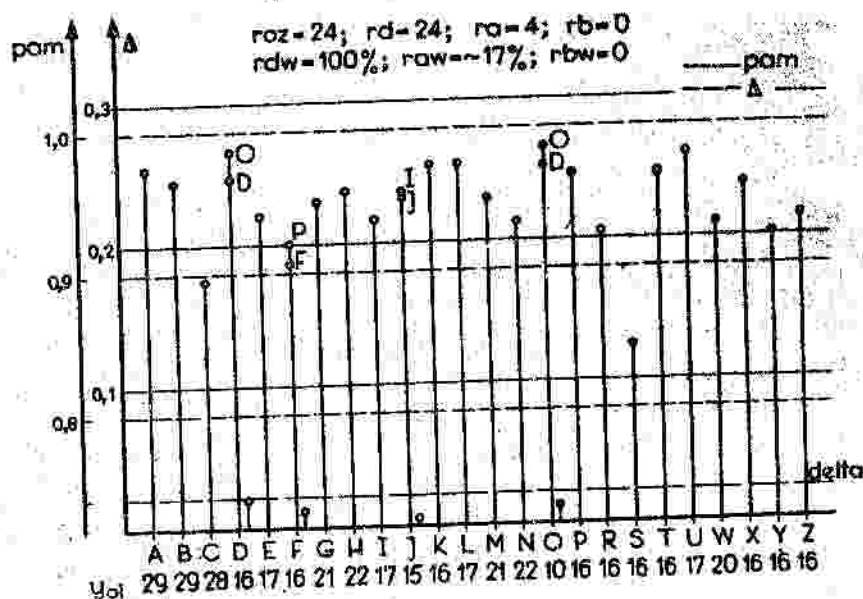
Rys. 38. Rozpoznawanie alfabetu dużych liter łacińskich
na = 325, $y_0 = 13$, zestaw danych: 20 (1301-1324)



Rys. 39. Rozpoznawanie alfabetu dużych liter łacińskich
na = 325, $y_0 = 13$, zestaw danych: 21 (1351-1374)



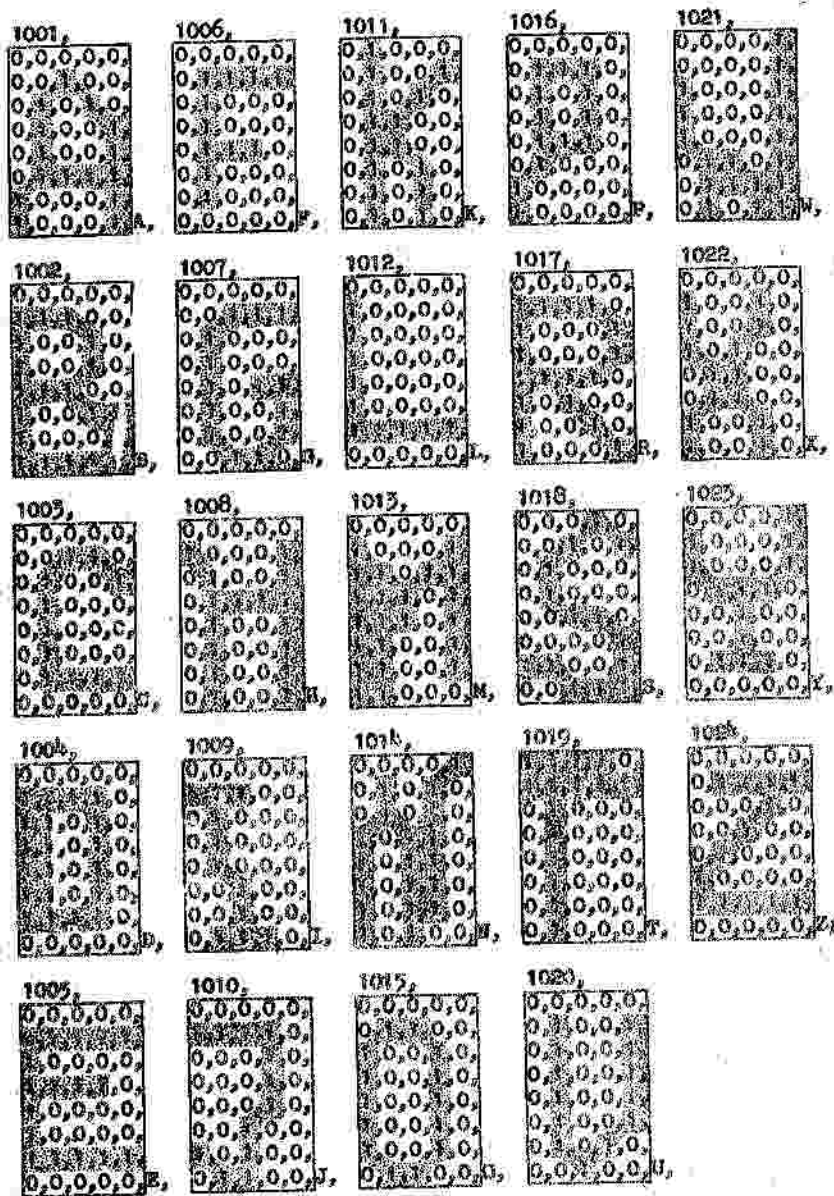
Rys. 40. Rozpoznawanie alfabetu dużych liter łacińskich
na = 414, $y_0 = 17$, zestaw danych: 17 (1151-1174)



Rys. 41. Rozpoznawanie alfabetu dużych liter łacińskich
na = 414, $y_0 = 17$, zestaw danych: 21 (1351-1374)

951, 0,0,1,0,0, 0,1,0,1,0, 1,0,0,1,0, 1,1,0,1,0, 1,0,1,1,1, 1,0,0,0,1, 1,0,0,0,1, 0,0,0,0,0,A,	956, 1,1,1,1,1, 1,0,0,0,0, 0,1,0,0,0, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0,F,	961, 1,0,0,0,0, 0,1,0,1,1, 0,1,1,0,0, 0,1,1,0,0, 0,1,1,0,0, 0,1,1,0,0, 0,1,0,1,0, 0,1,0,0,1,K,	966, 0,1,1,1,1, 0,1,0,0,1, 0,1,0,0,1, 1,1,0,0,1, 1,1,1,1,1, 1,0,0,0,0, 1,0,0,0,0, 1,1,0,0,0,P,	971, 0,0,0,1,0, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 0,1,0,0,1, 0,1,0,1,1, 0,1,1,0,1, 0,1,0,0,1,W,
952, 0,0,1,1,0, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,1,1,1, 0,1,0,0,1, 1,0,0,0,1, 1,1,1,1,0,B,	957, 0,1,1,1,0, 0,1,0,0,1, 0,1,0,0,0, 1,0,0,0,0, 1,0,1,1,0, 1,0,0,0,1, 0,1,0,0,1, 0,0,1,1,0,G,	962, 1,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0,L,	967, 0,0,0,0,0, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,0,1, 0,1,0,1,0, 0,1,1,0,0, 0,1,0,1,0, 0,1,0,0,1,R,	972, 1,0,0,0,0, 0,1,0,1,1, 0,1,0,1,0, 0,0,1,0,0, 0,0,1,1,0, 0,0,1,1,0, 0,1,0,0,1, 1,1,0,0,1,X,
953, 0,0,0,0,0, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,1,C,	958, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0,H,	963, 0,0,0,0,0, 1,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0,M,	968, 0,0,1,0,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,1,1,1,S,	973, 0,0,0,1,0, 0,0,0,1,0, 0,1,0,0,0, 0,0,1,0,0, 0,0,1,0,0, 0,0,1,0,0, 0,0,1,0,0, 0,1,1,1,0,Y,
954, 0,0,0,0,0, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,1,1,0,D,	959, 0,0,0,0,0, 0,0,1,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 1,1,1,0,0,I,	964, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0,N,	969, 0,1,1,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,T,	974, 1,1,1,1,1, 0,0,0,1,0, 0,0,0,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 1,0,0,0,1,Z,
955, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,1,0, 0,1,1,1,0, 0,1,0,1,0, 0,1,0,0,0, 1,0,0,0,1, 1,1,1,1,1,E,	960, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,1,0,0,J,	965, 0,0,0,0,0, 0,0,0,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,0,0, 0,0,1,1,0,O,	970, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,1,1,1,U,	

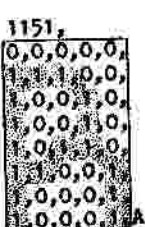
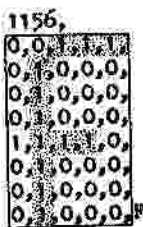
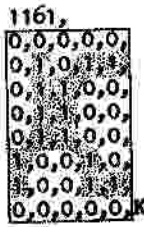
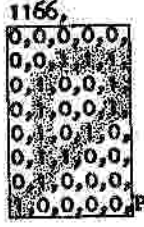
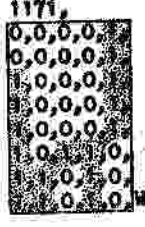
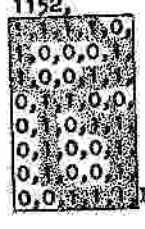
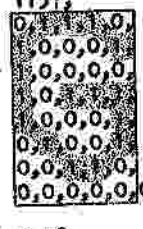
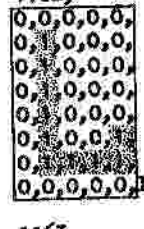
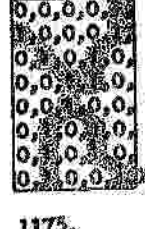
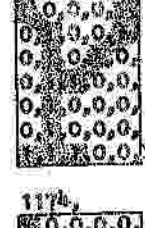
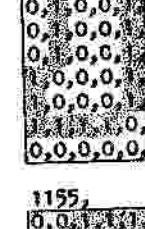
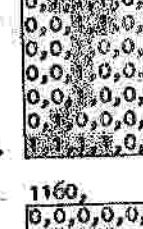
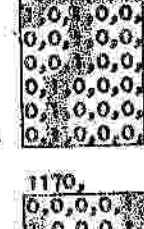
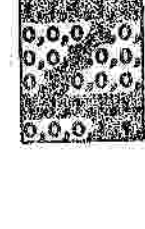
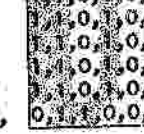
Zestaw danych nr 13



Zestaw danych nr 14

1051, 0,0,0,0,0, 0,1,1,1,1, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,1,1,1,1, 1,0,0,0,1, 1,0,0,0,1 A,	1056, 0,0,0,0,1, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 1,0,0,0,0 F,	1061, 1,0,0,0,1, 1,0,0,1,0, 1,0,1,0,0, 1,1,1,0,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 0,0,0,0,0 K,	1066, 0,0,0,0,0, 0,1,1,1,0, 0,1,0,0,1, 0,1,0,0,1, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0 P,	1071, 0,0,0,0,0, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1 W,
1052, 0,0,0,0,0, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1, 0,1,0,0,1 B,	1057, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 G,	1062, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 L,	1067, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 R,	1072, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 X,
1053, 0,0,1,0,0, 0,1,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 C,	1058, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 H,	1063, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 M,	1068, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 S,	1073, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 Y,
1054, 0,0,0,0,0, 0,1,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 D,	1059, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 I,	1064, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 N,	1069, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 T,	1074, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 Z,
1055, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 E,	1060, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 J,	1065, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 O,	1070, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 U,	

Zestaw danych nr 15

1151,  A,	1156,  F,	1161,  K,	1166,  P,	1171,  W,
1152,  B,	1157,  G,	1162,  L,	1167,  R,	1172,  X,
1153,  C,	1158,  H,	1163,  M,	1168,  S,	1173,  Y,
1154,  D,	1159,  I,	1164,  N,	1169,  T,	1174,  Z,
1155,  E,	1160,  J,	1165,  O,	1170,  U,	

1201, 0,0,1,0,0, 0,1,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,1,1,1,0, 1,0,0,0,1, 0,0,0,0,1 A,	1206, 0,0,0,0,0, 0,0,0,0,1, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 1,1,1,1,0, 1,0,0,0,0, 1,0,0,0,0 F,	1211, 1,0,0,0,0, 1,0,0,0,1, 1,0,0,1,0, 0,1,1,0,0, 0,1,1,0,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,0,1,0, 0,0,0,0,0 K,	1216, 0,0,1,0,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,1,0,0, 0,1,1,0,0, 0,1,0,0,0, 0,1,0,0,0 P,	1221, 0,0,0,0,0, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1 W,
1202, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 B,	1207, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0 G,	1212, 0,0,0,0,0, 0,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 L,	1217, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 R,	1222, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 X,
1203, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 C,	1208, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 H,	1213, 0,0,0,1,0, 0,0,0,1,0, 0,0,0,1,0, 0,0,0,1,0, 0,0,0,1,0, 0,0,0,1,0, 0,0,0,1,0 M,	1218, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 S,	1223, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 Y,
1204, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0 D,	1209, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 I,	1214, 0,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 N,	1219, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 T,	1224, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 Z,
1205, 0,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 E,	1210, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 J,	1215, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 C,	1220, 0,0,0,0,0, 0,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0 U,	

Zestaw danych nr 18

1301, 0,0,1,0,0, 0,0,1,1,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,1,1,0, 1,0,0,0,1, 1,0,0,0,1, 0,0,0,0,0,A,	1306, 0,0,0,0,0, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,1,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,F,	1311, 0,0,0,0,0, 1,0,0,0,1, 1,0,0,1,0, 1,0,1,0,0, 1,1,1,0,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0,K,	1316, 0,0,0,0,0, 1,1,1,1,1, 1,0,0,0,1, 1,0,0,0,1, 0,1,1,1,1, 0,1,0,0,0, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,0,0,P,	1321, 0,0,0,0,0, 1,0,0,0,1, 1,0,1,0,1, 1,0,1,0,1, 1,0,1,0,1, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0,W,
1302, 0,0,0,0,0, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 1,1,1,1,0, 0,0,1,0,0, 0,0,0,0,0,B,	1307, 0,0,0,0,0, 0,0,1,1,0, 0,1,0,0,0, 0,0,0,0,0, 0,1,1,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,G,	1312, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,I,	1317, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0,R,	1322, 0,0,0,0,0, 1,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,1,0,0, 0,0,1,0,0, 0,0,1,0,0, 0,0,0,0,0,X,
1303, 0,0,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,C,	1308, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,0,0, 0,1,1,1,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,H,	1313, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,M,	1318, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,S,	1323, 0,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,Y,
1304, 0,0,0,0,0, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,1,0,0,D,	1309, 0,0,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,1,0,0, 0,0,0,0,0, 0,0,0,0,0,I,	1314, 0,0,0,0,0, 0,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,N,	1319, 0,0,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,0,0,T,	1324, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,Z,
1305, 0,0,0,0,0, 0,1,1,1,1, 0,1,0,0,0, 0,1,0,0,0, 0,1,1,1,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,1,1,1,E,	1310, 0,0,0,0,0, 0,1,1,1,1, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,1,0, 0,1,1,1,0,J,	1315, 0,0,0,1,0, 0,1,0,0,1, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,0,1,0, 0,0,0,0,0,O,	1320, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,1,0, 0,1,1,0,0,U,	

Zestaw danych nr 20

1351, 0,0,1,0,0, 0,0,1,0,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,1,1,0, 0,1,1,1,0, 0,1,0,0,1, 1,0,0,0,1, 1,0,0,0,1,A,	1356, 1,1,1,1,1, 1,1,0,0,1, 1,1,0,0,0, 1,1,0,0,0, 1,1,1,0,0, 1,1,0,0,0, 1,1,0,0,0, 1,1,0,0,0, 1,1,0,0,0,F,	1361, 1,0,0,0,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,0,0, 1,0,1,0,0, 1,0,1,0,0, 1,0,0,0,0, 1,0,0,0,0,K,	1366, 1,1,1,1,0, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,0,0,0,1, 1,1,1,0,0, 1,1,1,0,0, 1,0,0,0,0, 1,0,0,0,0,P,	1371, 0,0,0,0,0, 1,0,1,0,1, 1,0,1,0,1, 1,0,1,0,1, 1,1,0,1,1, 1,1,0,1,1, 1,1,0,1,1, 0,1,0,1,0, 0,0,0,0,0,W,
1352, 1,1,1,0,0, 1,0,0,0,1, 1,0,0,0,0, 1,1,1,1,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,B,	1357, 1,1,1,0,0, 1,0,0,1,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,G,	1362, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0,L,	1367, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,0,0,0,0,R,	1372, 0,0,0,0,0, 1,0,1,0,0, 0,1,1,1,0, 0,0,1,0,0, 0,1,1,0,0, 0,1,1,0,0, 0,1,0,0,0, 1,0,0,1,0, 1,0,0,1,0,X,
1353, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,C,	1358, 1,0,0,0,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0, 1,0,0,1,0,H,	1363, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,M,	1368, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,S,	1373, 1,0,0,0,0, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,Y,
1354, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,D,	1359, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,I,	1364, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0,N,	1369, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,T,	1374, 0,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,Z,
1355, 1,0,0,0,0, 1,0,0,0,0, 1,1,1,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,1,1,1,0,E,	1360, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,1,1,1,0,J,	1365, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 1,0,0,0,0, 0,1,1,1,0,U,	1370, 0,0,0,0,0, 1,0,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,1,0,0,0, 0,0,0,0,0,V,	

Zestaw danych nr 21

Zmienną y_0 wyprowadzono, aby umożliwić orientacyjne scharakteryzowanie wzrostu poprawnych odpowiedzi wraz ze wzrostem długości ciągów uczących.

Wyniki zebrano na rysunkach 31 do 41. Dla każdego rozpoznawanego symbolu zaznaczonego na osi poziomej rysunku podano wartości pam oraz Δ . W przypadku błędnego rozpoznania wartość pam podano w kółku, a nazwę symbolu podaną przez model umieszczono obok. W celu identyfikacji rozpoznawanych symboli na każdym rysunku podano numer zestawu danych i zakres numerów porządkowych zastosowanych symboli tego zestawu (zestawy danych używanych w tych badaniach przedstawiono na str. 56-63).

Przy opisie każdego rysunku podano wartości zmiennych:

roz - całkowita ilość operacji rozpoznawania

rd - ilość odpowiedzi poprawnych, przy czym za poprawną uznano odpowiedź, która: a) przy rozpoznaniu jednoznacznym ($\Delta > \delta$) jest prawidłowa bez względu na wartość pam , b) przy rozpoznawaniu alternatywnym ($\Delta \leq \delta$) jest jedną z dwóch odpowiedzi podanych alternatywnie. Uwaga: odnośnie do liter O i D odpowiedź przyjmowano zawsze za poprawną, jeśli wskazywała na dowolną z tych dwóch liter

rdw - wartość względna rd w procentach $rdw = rd/roz \cdot 100\%$

ra - ilość rozpoznań alternatywnych takich, że jedna z dwóch odpowiedzi była prawidłowa

raw - wartość względna ra w procentach $raw = ra/roz \cdot 100\%$

rb - ilość rozpoznań błędnych $rb = roz - rd$

rbw - wartość względna rb w procentach $rbw = rb/roz \cdot 100\%$

Przy wartościach Δ podano nazwy symboli odpowiadających pam

Na rysunkach 31 do 33 przedstawiono wyniki badań rozpoznawania trzech różnych alfabetów (zestawy nr 14, 15 i 21) przez model jednorazowo nauczony ($y_0 = 1$) alfabetu nr 13. Średnia wartość rdw z tych trzech badań

$$rdw_{sr} = \frac{83 + 79 + 71}{3} = 78\%$$

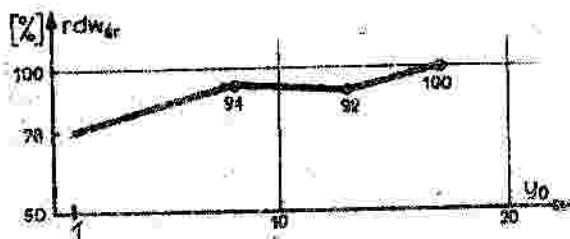
Na rysunkach 34 do 41 przedstawiono wyniki badań rozpoznawania przy $y_0 = 8$ (rys. 34 i 35), $y_0 = 13$ (rys. 36-39) oraz $y_0 = 17$ (rys. 40 i 41).

Na stronach 56 do 63 zestawiono alfabety liter, które używano przy powyższych badaniach.

Zależność wartości rdw_{gr} od y_0 ujemuje wykres na rys. 42. Wykres ten należy traktować jedynie jako bardzo orientacyjny z powodu stosunkowo niedużej ilości badań oraz operowania y_0 - wartością średnią dość nierównomiernego rozkładu wartości y_{10} (w przypadku $y_0 = 13$ i $y_0 = 17$).

Należy także zwrócić tu uwagę na przyjęte kryterium oceny poprawności odpowiedzi (określenie zmiennej rd).

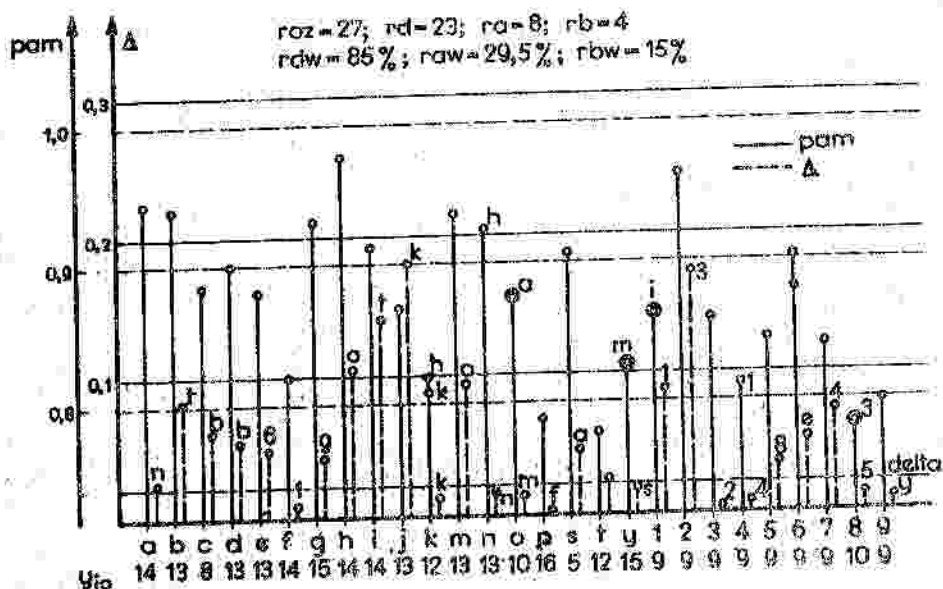
Wykres $rdw_{gr} = f(y_0)$ wskazuje na stosunkowo małą zależność rdw_{gr} od y_0 przy $y_0 \geq 8$. Tego rodzaju wynik jest w pełni zrozumiały w świetle wykresów ustalenia się wag cech w_{ij} (rys. 21 i 22), na których widać, że dla $y_{10} \geq 8$ wartości w_{ij} zmieniają się nieznacznie. Powyższe świadczy o stosunkowo szybkim ustalaniu się ostatecznych rozkładów w_{ij} w obrazach wzorcowych, a więc o dość szybkiej zbieżności algorytmu uczenia.



Rys. 42

Na rys. 43 przedstawiono próbę badań rozpoznawania modelu nauczzonego 51-literowego alfabetu (małe i duże litery łacińskie oraz cyfry). Mały procent poprawnie rozpoznanych małych liter i cyfr w stosunku do liter dużych można częściowo wytłumaczyć mniejszymi wartościami y_{10} dla $i = 25, \dots, 51$. W przypadku rozpoznawania alfabetu złożonego z małych liter oraz cyfr przy większych wartościach y_{10} (rys. 44) wartość rdw wyraźnie wzrosła (do 85%).

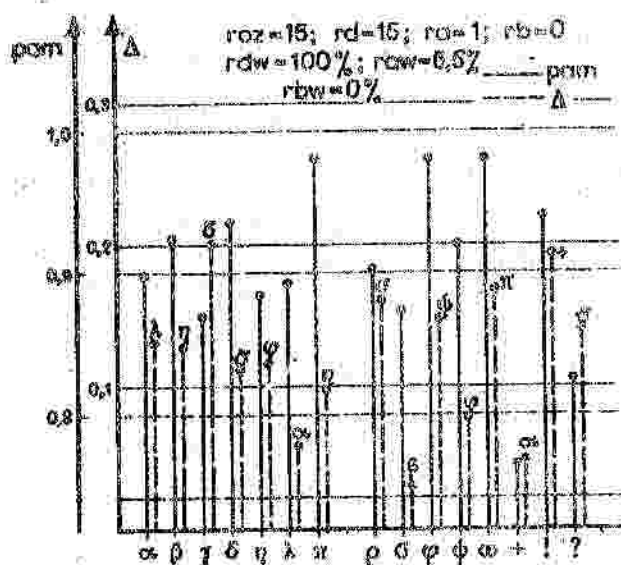
Pozostaje ona jednak nadal znacznie mniejsza niż w przypadku dużych liter. Można to przypisać przede wszystkim temu, że układ cech (rozwiązanie warstwy C) był dobrany głównie z punktu widzenia rozpoznawania dużych liter.

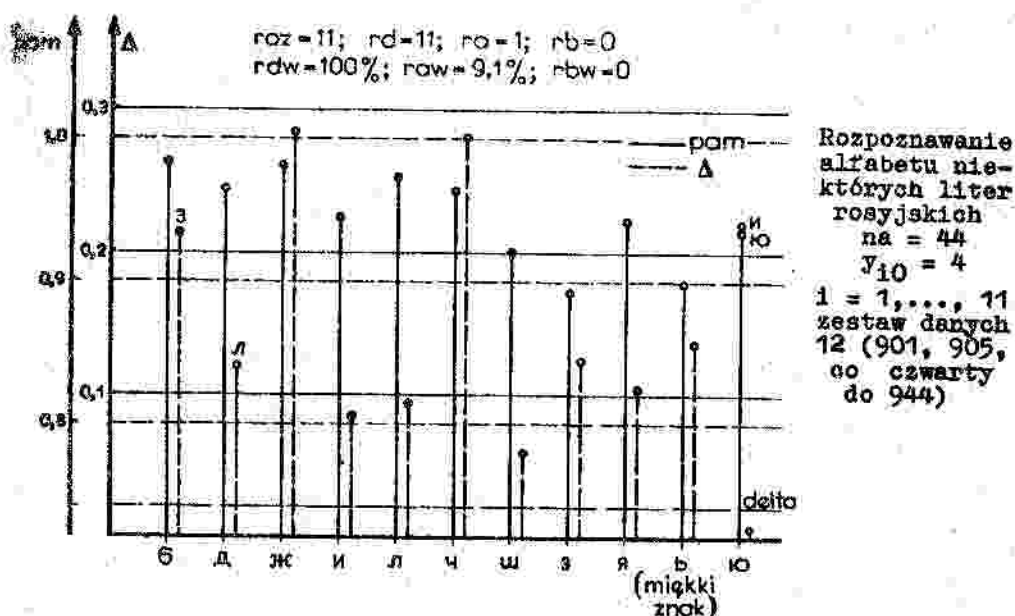


Rys. 44. Rozpoznawanie małych liter i cyfr
na = 292, $\gamma_0 \approx 11$, zestaw danych: 21 (1375-1401)

Rys. 45

Rozpoznawa-
nie alfabetu
małych liter
greckich i
niektórych
znaków pi-
sarskich -
zestaw da-
nych nr 11
(729, 733,
co czwarty
do 787) (al-
fabet był u-
żyty przy
nauczaniu





Rys. 46

Na rys. 45 i 46 zebrano wyniki badań rozpoznawania niektórych liter rosyjskich, greckich i kilku znaków pisańskich.

5.4.7. Wnioski

Badania modelu urządzenia rozpoznającego na maszynie cyfrowej wykazały, że istnieje pewien określony zakres wartości progów wag cech gg oraz gd, przy których proces rozpoznawania jest najkorzystniejszy. Potwierdziło się także przypuszczenie, że dla symboli mocno zniekształconych lub nieznanych odpowiednia wartość P_{maks} (pam) jest wyraźnie mniejsza niż w przypadku symboli znanych. Dzięki temu było możliwe wydzielenie różnych kategorii rozpoznawania przez ustalenie granicznych wartości: proc 1, proc 2 i proc 3 jednego tylko parametru - P_{maks} .

Proces ustalenia się wag cech w_{ij} (rys. 21 i 22) następował w układzie bardzo szybko. Dzięki temu już po jednorazowym nauczaniu uzyskano duży procent odpowiedzi poprawnych (78%). Świadczy to o słuszności przyjętej idei rozwiązania układu recepcyjnego, w szcze-

gólności zaś o właściwym doborze cech w warstwie C.

Analiza błędów rozpoznawania wykazuje, że błędy są spowodowane przede wszystkim pomyleniem przedstawicieli kilku charakterystycznych grup symboli:

O				D
R				A
H	M	N	W	
P				F
S				Z

W przypadku pierwszych trzech grup jako zasadniczą przyczynę błędów należy wskazać niedostateczną rozróżnialność siatkówki dla tych symboli. Mylenie zaś symboli P oraz F i Z oraz S ówia-
dczy o stosunkowo małej czułości modelu na własności różniące te litery (zamknięcie lub niezamknięcie linii ciągłej, kierunek nachylenia linii skośnej itp.).

Dla szeregu liter (np. B, C, L, T, U) rozpoznawanie było zawsze poprawne mimo nawet dużych przekształceń symboli. Ponieważ mylenie przedstawicieli różnych ukazanych pięciu grup było zjawiskiem bardzo rzadkim, przeto przez traktowanie tych grup jako klas symboli można by uzyskać stosunkowo bardzo dużą niezawodność rozpoznawania. Tak więc w pewnych szczególnych przypadkach urządzenie rozpoznające zbudowane na podstawie wpisanego modelu mogłoby znaleźć zastosowanie praktyczne nawet przy przyjętej 40-elementowej siatkówce.

6. ZAKOŃCZENIE

Przedstawiona praca stanowi próbę opracowania urządzenia rozpoznającego dla symboli alfanumerycznych, które przy możliwie najprostszej realizacji technicznej zapewniałoby dużą niezawodność rozpoznawania.

Ponieważ spełnienie powyższych warunków zależy w dużym stopniu od rozwiązania układu recepcyjnego, jemu też poświęcono stosunkowo dużo uwagi.

Przyjęty algorytm określania macierzy C można bardzo łatwo zrealizować w oparciu o półprzewodnikową technikę progową. Jednocześnie, jak wykazały badania modelu na maszynie cyfrowej, zapewnia on stosunkowo dużą niezależność macierzy C od przekształceń izomorficznych symboli. Chociaż przyjęte w niniejszej pracy roz-

wiązanie układu recepcyjnego dotyczy siatkówki o małej ilości elementów (8×5), to jednak ogólna idea rozwiązania może być łatwo rozszerzona na siatkówkę o większej ilości elementów. Są tu jednak pożądane dalsze badania, zwłaszcza z punktu widzenia optymalnego doboru obszarów siatkówki oraz wykrywania bardziej złożonych cech.

Opisane algorytmy uczenia i rozpoznawania nie zależą od ilości cech (n), toteż mają charakter uniwersalny. Są także bardzo łatwe do realizacji technicznej. Algorytmowi uczenia odpowiada w interpretacji geometrycznej określanie współrzędnych środka rozkładu punktów odpowiadających jednej klasie obiektów. Dzięki wygodnej postaci macierzy Y każda operacja ucząca sprowadza się do dodawania jednostki do określonych elementów tej macierzy, co jest bardzo proste do realizacji.

W procesie rozpoznawania odbywa się obliczanie współczynników bliskości p_i , które stanowią liniową funkcję odległości w przestrzeni E_n . Przyjęcie powyższego, nie stosowanego dotychczas, określenia współczynnika bliskości wynika przede wszystkim z powodu dużej prostoty jego praktycznego obliczenia (zwłaszcza w postaci wzoru (10)). Problem jego optymalności nie został tu podjęty należałoby tu szczegółowo przeanalizować rozkłady punktów w przestrzeni E_p dla poszczególnych klas symboli (rozkłady te zależą m. in. od budowy układu recepcyjnego). Wydaje się jednak, że w przypadku odpowiedniego rozwiązania układu recepcyjnego (dającego dobre "skupienie" punktów poszczególnych klas) problem ten nie jest pierwszorzędnej wagi. Analiza badań modelu urządzenia, jak i pewnych wstępnych prac przeprowadzonych przez autorów przed przyjęciem podanego rozwiązania układu recepcyjnego wykazała bowiem, że zasadniczy wpływ na poprawność rozpoznawania ma układ recepcyjny, a stosunkowo mniejszy wybór algorytmu rozpoznawania (w sensie przyjęcia tej lub innej miary odległości lub podziału przestrzeni E_p na obszary aproksymowane na przykład hiperpłaszczyznami).

Należy jednak zaznaczyć, że podane powyżej algorytmy uczenia i rozpoznawania opierają się na pewnych założeniach upraszczających w stosunku do znanych optymalnych teoretycznych algorytmów rozpoznawania, a mianowicie:

1. Założenie równomiernego rozkładu prawdopodobieństwa zjawiania się reprezentantów poszczególnych klas, tzn. nieuwzględnianie w algorytmie rozpoznawania możliwości wykorzystania informacji a

priori, np. o częstszym zjawianiu się reprezentantów pewnych klas przed reprezentantami innych klas. Przy tego typu prostym urządzeniu rozpoznającym założenie to wydaje się zupełnie naturalne.

2. Założenie równej dyspersji rozkładów położenia punktów odpowiadających poszczególnym klasom w przestrzeni obrazów badanych E_b traktowanej jako przestrzeń euklidesowa. Założeniu temu odpowiada operowanie w algorytmie rozpoznawania wyłącznie obrazami wzorcowymi, które w interpretacji geometrycznej stanowią środki rozkładu punktów jednej klasy.

Jak wiadomo z opracowań teoretycznych, w przypadku różnej dyspersji tych punktów optymalny algorytm rozpoznawania wymaga uwzględnienia tego faktu.

Założenie to wydaje się całkowicie do przyjęcia w przypadku rozpatrywania zbioru obiektów rozpoznawania takiego jak symbole alfanumeryczne oraz w świetle uwag podanych powyżej.

Korzystne wyniki badań modelu na maszynie cyfrowej wskazują na to, że proponowana koncepcja rozwiązania urządzenia rozpoznającego jest perspektywiczna i może być wykorzystana do realizacji praktycznej przy założonej 40-elementowej siatkówce a także przy siatkówce o znacznie większej ilości elementów (po odpowiedniej modyfikacji układu recepcyjnego).

LITERATURA

1. Akers S.B., Rutter B.H.: The use of threshold logic in pattern recognition. Wescon 63. 1963 pt. 4 p. 1-7.
2. Doyle W.: Recognition of sloppy hand-printed characters. Proc. WJCC 1960.
3. Jelisiejew B.K., Kowalewski B.A.: Issledowanie algoritma rozpoznawania maszynopisanych znakow. Z. vychisl. matematiki 1962 Nr. 5 s. 902-911.
4. Gluszkow B.M.: Pro adin princip pobudowl uniwersalnogo czytajuszczo awtomata. Awtomatika 1962 Nr. 1.
5. Iwachnienko O.g.: Sistemy szczo samoorganizujutsja z dodatnimi zworotnymi zwjazzkami. Awtomatika 1962 Nr. 3.
6. Kamensky L.A., Liu C.N.: Computer-automated design of multi-font print recognition logic. J. Research a. Development 1963 No. 1.

7. Kazmierczak H., Steinbuch K.: Adaptive systems in pattern recognition. Trans. IEEE 1963 No. EC-12, 6.
8. Parra I.K.: Dosliżenijs processow nawczanijs u rozpiżnajućyjs sistemi z odnim dodatnim zwrotnim zwjazkom. Awtomatika 1963 Nr. 5.
9. Socziwko W.P.: Elektronnyje opoznajusćyjs ustrojstwa. Eniergietika 1966 Moskwa.
10. Spooner M.G., Swonger C.W., Beach J.B.: An evaluation of certain property formation techniques for pattern recognition. Wescon 64 Technical papers 1964 pt. 4 p. 4-81.
11. Vossler C., Uhr L.: Computer simulation of a perceptual learning model for sensory pattern recognition. System Development Corp. Santa Monica (Cal.), 1962 March 8 (Report).
12. Madey J.: "Algol 60" "Gier Algol III". Warszawa 1965 Uniwersytet Warszawski.

Program modelujący urządzenie rozpoznające w języku Gier-Algol

```

begin comment model. ukl. rozp.
  n=ilosc cech
  m=ilosc obiektow do rozp.
  sterowanie z monitora:
  n - nauczanie
  x - organizacja wstepna (jednorazowe nauczanie
    wstepne calego ukladu)
  r - rozpozn.
  j - powtorne rozpoznawanie (bez zmiany C[i])
  e - wypisanie tablicy C[i]
  z - zerowanie roz,rd,roz1,roz2,roz3,roz4,
  c - zmiana m, delta
  d - zmiana gg i gd
  f - zmiana proc1, proc2, proc3
  k - zakonczenie
  kb on ^ a - nauczanie automatyczne
    ^ l - znam,na,
    ^ n - .....+tablica Y dla wywolanego symbolu,
    ^ v - .....+wyprowadzenie tablicy Y,
    ^ m - .....+..... Ysora,
    ^ r - roz,rd,roz1,roz2,roz3,roz4 -z wydrukowa
      napisu
    ^ s - .....-bez wydruku
  Na pytanie „Skad dane:,, : c - z czytnika
    p - z maszyny do pisania
  Jesli litera - nazwa symbolu - zakonczona jest sred-
  nikiem (x:=1) w danych przy naucz. automat., to maszy-
  na idzie do startu.
  Na pytanie „Czy dobrze:,, : t - tak i uczy sie tego
    symbolu rd:=rd+1
    n - nie
    u - pyta sie „Co to jest:,,
    i nastepnie uczy sie!

  integer i,j,k,l,n,n1,m,v,ster,ns,znam,adres,roz,rd,odp,
    roz1,roz2,roz3,roz4,po,sum,adr,adr1,adr2,adr3;
  real pam,pam1,pam2,pam3,pam4,proc1,proc2,proc3,proc4,gg,gd,delta;
  boolean b;

  input(n,m,proc 1,proc 2,proc 3,gg,gd,delta);

  begin integer array Q[1:m,1:3],R[1:2],X[0:n],Z[1:7],C[1:n];
  integer r,r1,x2,x3,r4;

  poczatek:
    Y[0]:= 0;
    for i:= 1 step 1 until n do

```

```

begin
  Y[1]:= 0; C[1]:= 0;
end;
for i:= 1 step 1 until m do
  for j:= 1,2,3 do
    Q[i,j]:= 0;
  for j:=1 step 1 until m do
    begin
      drumplace:=4095 - (j-1)*(n+1);
      to buf (Y);
    end;
  end;
  sum:=na:=0;
zeroword:
  ros:=rd:=roz1:=roz2:=roz3:=roz4:=0;
  sdr:=1;

start:
  writcr;
  writetext(⟨Go robic: ⟩);
  star:=typechar;
  if star = 33 then go to rozpoznanie;
  if star = 51 then
    begin
      writcr; writetext(⟨m,delta= ⟩);
      write(⟨ddd, ⟩,m);
      writetext(⟨< ⟩); write(⟨n.ddd0, ⟩,delta);
      writcr; writetext(⟨a= ⟩);
      a:= typein;
      writetext(⟨< delta= ⟩);
      delta:= typein;
      go to start;
    end;
  if star = 54 then
    begin
      writcr; writetext(⟨proc1,2,3= ⟩);
      write(⟨n.ddd0, ⟩,proc1,writetext(⟨< ⟩),proc2,writetext(⟨< ⟩),proc3);
      writcr; writetext(⟨proc1= ⟩);
      proc1:= typein;
      writetext(⟨< proc2= ⟩);
      proc2:= typein;
      writetext(⟨< proc3= ⟩);
      proc3:= typein;
      go to start;
    end;
  if star = 52 then

```

```

begin
  writecr; writetext(<<gg= >);
  gg:= typein;
  writetext(<<   gd= >);
  gd:= typein;
  go to start;
end;
if ster=23 then
begin
  writecr;
  writetext(<< Organizacja wystepna: >);
  writecr; writetext(<<znaz= >);
  input(znaz,na);
  write(<ddd>,znaz);
  writetext(<< naz= >); write(<ddd>,na); writecr;
  for j:=1 step 1 until znaz do
  begin
    input(1);
    for i:=1,2,3,4 do
    begin
      Z[i]:=inchar;
      if Z[i]<64 then w:=0 else
      begin
        w:=1; Z[i]:=Z[i]-128;
      end;
      if Z[i]=0 v i=4 then
      begin
        for i:=i+1 while i<7 do
          Z[i]:=0;
        go to okrQ;
      end;
      end okroslenia Z[i];
    okrQ:
      if j = znaz then
      begin
        write(<ddd>,1);
        writetext(<< >);
        writechar(Z[i]); writetext(<< >);
      end;
      if j = znaz then
      begin
        write(<ddd>,1);
        writetext(<< >); writechar(Z[i]); writetext(<< >);
      end;
  end;

```

```

pack(b,0,5,Z[7],6,11,Z[6],12,17,Z[5],18,23,Z[4],24,29,Z[3],
    30,35,Z[2],36,41,Z[1]);
split(b,0,23,R[2],24,41,R[1]);
Q[j,1]:=R[1];Q[j,2]:=R[2];Q[j,3]:=w;
if 1=10 V 1=20 V 1= 30 V 1=40 V 1=50 V 1=60 V 1=70 V 1=80
then
writecr;
drumplace:=4095 - (j-1)*(n+1);
input(Y);
to buf(Y);
end zalad.Y;
go to start;
end;
if ster=53 then
begin
writecr;
writetext('<C>',write('<ddd>',1),<[>]);
for i:=1 step 1 until n do
write('<n>',C[i]);
writetext('<[>');
go to start;
end;
end;
if ster=34 then go to koniec;
if ster=25 then go to zerowanie;
if kb on ^ ster=49 then
begin
if ster=41 V ster=18 then go to rozpoznanie
else go to nauczanie
end;
cechy;
begin comment cechy;
integer array A[1:8,1:5],B1[1:8,1:4],B2[1:7,1:5],P[1:52];
integer i,j,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13;
integer procedure l(z); value z; boolean z;
if z then 1 else 0;
if ster=49 then
begin
input(1,A);
goto a1;
end else
begin
writetext('< End dane: >'); if typechar=39 then

```

```

begin
  i:=typein;
  for i:=1 step 1 until 8 do
    for j:=1 step 1 until 5 do
      A[i,j]:=typein;
    end
  else input(1,A);
end;
a1:
if 1-entier(1/10)*10=1 then writecr;
write('add1,1); writetext(4, 1);
s1:=s2:=s3:=s4:=s5:=s6:=s7:=s8:=s9:=s10:=s11:=s12:=s13:=0;
for i:=1 step 1 until 8 do
  for j:=1 step 1 until 4 do
    B1[i,j]:=I(A[i,j]+A[i,j+1]>0);
  for j:=1 step 1 until 5 do
    for i:=1 step 1 until 7 do
      B2[i,j]:=I(A[i,j]+A[i+1,j]>0);
    for i:=1 step 1 until 7 do
      begin
        P[1]:=I(B1[i,1]+B1[i+1,1]=1);
        P[1+7]:=I(B1[i,4]+B1[i+1,4]=1);
        P[1+14]:=I(A[i,1]+A[i+1,1]=1);
        P[1+21]:=I(A[i,3]+A[i+1,3]=1);
        P[1+28]:=I(A[i,5]+A[i+1,5]=1);
        P[1+35]:=I(B1[i,2]+B1[i+1,2]=1);
        P[1+42]:=I(B1[i,3]+B1[i+1,3]=1);
        s1:=s1+P[1];
        s2:=s2+P[1+7];
        s3:=s3+P[1+14];
        s4:=s4+P[1+21];
        s5:=s5+P[1+28];
        s9:=s9+P[1+35];
        s10:=s10+P[1+42];
      end;
    s1:=s1+B1[i,1]+B1[8,1];
    s2:=s2+B1[i,4]+B1[8,4];
    s3:=s3+A[i,1]+A[8,1];
    s4:=s4+A[i,3]+A[8,3];
    s5:=s5+A[i,5]+A[8,5];
    s9:=s9+B1[i,2]+B1[8,2];
    s10:=s10+B1[i,3]+B1[8,3];
    for i:=1 step 1 until 8 do
      begin
        s6:=s6+B1[i,1];
        s7:=s7+B1[i,4];

```

```

s8:=a8+A[1,3];
s11:=s11+B1[1,2];
s12:=s12+B1[1,3];
end;
C[1]:=I(s1=2 ∧ s6>5);
C[2]:=I(s1=2 ∧ s6<5);
C[3]:=I(s1=4 ∨ s3=4);
C[4]:=I(s2=2 ∧ s7>5);
C[5]:=I(s2=2 ∧ s7<5);
C[6]:=I(s2=4 ∨ s5=4);
C[7]:=I(s9=2 ∧ s11>5 ∨ s10=2 ∧ s12>5);
C[8]:=I(s4=4);
C[9]:=I(s4=6);
s1:=s2:=s3:=s4:=s5:=s6:=s7:=s8:=s9:=s10:=s11:=s12:=s13:=0;
for i:=1 step 1 until 4 do
begin
P[1]:=I(B2[1,1]+B2[1,1+1]=1);
P[1+4]:=I(B2[3,1]+B2[3,1+1]=1);
P[1+8]:=I(B2[4,1]+B2[4,1+1]=1);
P[1+12]:=I(B2[5,1]+B2[5,1+1]=1);
P[1+16]:=I(B2[7,1]+B2[7,1+1]=1);
P[1+20]:=I(A[1,1]+A[1,1+1]=1);
P[1+24]:=I(A[2,1]+A[2,1+1]=1);
P[1+28]:=I(A[3,1]+A[3,1+1]=1);
P[1+32]:=I(A[4,1]+A[4,1+1]=1);
P[1+36]:=I(A[5,1]+A[5,1+1]=1);
P[1+40]:=I(A[6,1]+A[6,1+1]=1);
P[1+44]:=I(A[7,1]+A[7,1+1]=1);
P[1+48]:=I(A[8,1]+A[8,1+1]=1);
s1:=s1+P[1];
s2:=s2+P[1+4];
s3:=s3+P[1+8];
s4:=s4+P[1+12];
s5:=s5+P[1+16];
s6:=s6+P[1+20];
s7:=s7+P[1+24];
s8:=s8+P[1+28];
s9:=s9+P[1+32];
s10:=s10+P[1+36];
s11:=s11+P[1+40];
s12:=s12+P[1+44];
s13:=s13+P[1+48];
end;
s1:=s1+B2[1,1]+B2[1,5];
s2:=s2+B2[3,1]+B2[3,5];
s3:=s3+B2[4,1]+B2[4,5];

```

$a4:=a4+B2[5,1]+B2[5,5];$
 $a5:=a5+B2[7,1]+B2[7,5];$
 $a6:=a6+A[1,1]+A[1,5];$
 $a7:=a7+A[2,1]+A[2,5];$
 $a8:=a8+A[3,1]+A[3,5];$
 $a9:=a9+A[4,1]+A[4,5];$
 $a10:=a10+A[5,1]+A[5,5];$
 $a11:=a11+A[6,1]+A[6,5];$
 $a12:=a12+A[7,1]+A[7,5];$
 $a13:=a13+A[8,1]+A[8,5];$
 $i:=B2[1,1]+B2[1,2]+B2[1,3]+B2[1,4]+B2[1,5];$

$C[10]:=I(a1=2 \wedge a6=4 \wedge i \geq 3);$
 $C[11]:=I(a1=2 \wedge a6=4 \wedge i \leq 3);$
 $C[12]:=I(a1=4 \vee a6=4);$
 $C[13]:=I(B2[2,2]+B1[3,2]+B2[3,3]+B1[3,3]+B2[2,4]=5);$
 $C[14]:=I(a8=2 \wedge A[3,1]+A[3,2]+A[3,3]+A[3,4]+A[3,5] \geq 2);$
 $C[15]:=I(a7=4 \vee a8=4 \vee a9=4);$

$C[16]:=I(B2[6,2]+B1[6,2]+B2[5,3]+B1[6,3]+B2[6,4]=5);$
 $C[17]:=I(a11=2 \wedge A[6,1]+A[6,2]+A[6,3]+A[6,4]+A[6,5] \geq 2);$
 $C[18]:=I(a10=4 \vee a11=4 \vee a12=4);$

$i:=B2[7,1]+B2[7,2]+B2[7,3]+B2[7,4]+B2[7,5];$
 $C[19]:=I(a5=2 \wedge a13=4 \wedge i \geq 3);$
 $C[20]:=I(a5=2 \wedge a13=4 \wedge i \leq 3);$
 $C[21]:=I(a5=4 \vee a13=4);$

$i:=B2[4,1]+B2[4,2]+B2[4,3]+B2[4,4]+B2[4,5];$
 $C[22]:=I((a2=2 \wedge B2[3,1]+B2[3,2]+B2[3,3]+B2[3,4]+B2[3,5] \geq 3 \vee a3=2 \wedge i \geq 3 \vee a4=2 \wedge B2[5,1]+B2[5,2]+B2[5,3]+B2[5,4]+B2[5,5] \geq 3) \wedge a3 \neq 6);$
 $C[23]:=I(a3=2 \wedge i \leq 3);$
 $C[24]:=I(a3=4 \vee a3=6);$

$P[1]:=I(B2[1,1]+B1[2,1]=1);$
 $P[2]:=I(B1[2,1]+B2[2,2]=1);$
 $P[3]:=I(B2[2,2]+B2[3,2]=1);$
 $P[4]:=I(B2[3,2]+B1[4,2]=1);$
 $P[5]:=I(B1[4,2]+B2[4,3]=1);$
 $P[6]:=I(B2[4,3]+B1[5,3]=1);$
 $P[7]:=I(B1[5,3]+B2[5,4]=1);$
 $P[8]:=I(B2[5,4]+B2[6,4]=1);$
 $P[9]:=I(B2[6,4]+B1[7,4]=1);$
 $P[10]:=I(B1[7,4]+B2[7,5]=1);$

```

P[11]:=I(B2[1,5]+B1[2,4]=1);
P[12]:=I(B1[2,4]+B2[2,4]=1);
P[13]:=I(B2[2,4]+B2[3,4]=1);
P[14]:=I(B2[3,4]+B1[4,3]=1);
P[15]:=I(B1[4,3]+B2[4,3]=1);
P[16]:=I(B2[4,3]+B1[5,2]=1);
P[17]:=I(B1[5,2]+B2[5,2]=1);
P[18]:=I(B2[5,2]+B2[6,2]=1);
P[19]:=I(B2[6,2]+B1[7,1]=1);
P[20]:=I(B1[7,1]+B2[7,1]=1);
s1:=s2:=s3:=s4:=0;
for i:=1 step 1 until 10 do
begin
  s1:=s1+P[i];
  s2:=s2+P[i+10];
end;

```

```

s3:=B2[1,1]+B1[2,1]+B2[2,2]+B2[3,2]+B1[4,2]+B2[4,3]+B1[5,3]+
  B2[5,4]+B2[6,4]+B1[7,4]+B2[7,5];
s4:=B2[1,5]+B1[2,4]+B2[2,4]+B1[4,3]+B2[4,3]+B1[5,2]+
  B2[5,2]+B2[6,2]+B1[7,1]+B2[7,1];
s1:=s1+B2[1,1]+B2[7,5];
s2:=s2+B2[1,5]+B2[7,1];

```

```

C[25]:=I(s1=2 ∧ s3>);
C[26]:=I(s1=2 ∧ s3<);
C[27]:=I(s1=4);

```

```

C[28]:=I(s2=2 ∧ s4>);
C[29]:=I(s2=2 ∧ s4<);
C[30]:=I(s2=4);

```

end programu ,,Cachy,,;

if ster = 41 ∨ ster=18 then go to rozpoznawanie;

naucanie:

if kb on ∧ ster=49 then

begin

writecr;

writetext(⟨znam ⟩);write(⟨add⟩,znam);writetext(⟨<
symbol⟩);

```

if znam = 1 then
go to 11;
i:=10x(znam/10 - entier(znam/10));
j:=100x(znam/100 - entier(znam/100));
if (i=2 v i=3 v i=4) ^ (j=12 v j=13 v j=14) then
writetext(<ce>) else writetext(<ci>);

11:
writcr;
writetext(<Uczono mnie >); write(<add>, na); writetext(<
razy>);

12:
if ster=22 v ster=36 then
begin
outcr;
outtext(<znam=>); output(<add>, znam); outtext(<, >);
output(<add>, na); outtext(<, >);
outcr;
for j:=1 step 1 until znam do
begin
if -, kbon then go to start;
drumplace:=4095-(j-1)x(n+1);
from buf(Y);
outcr;
outtext(if ster=22 then <Y> else <norm>);
output(<add>, j); outtext(<(>);
R[1]:=Q[j,1]; R[2]:=Q[j,2]; w:=Q[j,3];
for i:=1,2,3,4,5,6,7 do
begin
Z[i]:=entier(R[i] if i<3 then 1 else 2)/64 (if i<3 then
1-1 else 1-4));
Z[i]:=(Z[i]/64 - entier(Z[i]/64))x64;
end;
if w=1 then outchar(60);
for i:=1,2,3,4,5,6,7 do
begin
if i=4 then go to 13;
outchar(Z[i]);
end;

13:
outchar(58);
outtext(<(>);
if ster=36 then
for i:=1 step 1 until n do

```

```

Y[i]:=Y[i]/Y[0]*100;
for i:=0 step 1 until n do
begin
  output(⟨add⟩,Y[i]);
  if i≠n then outtext(⟨<⟩);
  if ster=36 then outtext(⟨<⟩);
end;
outtext(⟨<⟩);
end;
go to start;
end;
if ster=37 then
begin
  writecr;
  writetext(⟨<Y dla: ⟩);
  go to odczyt;
end else go to start;
end;

```

wskazanie symbolu:

```

if ster=49 then go to a2;
writecr;
writetext(⟨<Co to jest: ⟩);

```

a2:

odczyt:

```

w:=0;
for i:=1,2,3,4,5,6,7 do
begin
  Z[i]:=if ster=49 then inchar else typechar;
  if Z[i]=27 ∨ Z[i]=64 ∨ Z[i]=133 then
  begin
    if Z[i]=133 then r:=1 else r:=0;
    for i:=i+1 while i<8 do
      Z[i-1]:=0;
    goto PA;
  end;
  if Z[i]<64 then w:=0 else
  begin
    w:=1; Z[i]:=Z[i]-128;
  end;
end;

```

PA:

```

pack(b,0,5,Z[7],6,11,Z[6],12,17,Z[5],18,23,Z[4],24,29,Z[3],
30,35,Z[2],36,41,Z[1]);

```

```

split(n,0,23,R[2],24,41,R[1]);
if kb on A ster=37 then go to ktore j;
if znam=0 then
begin
  Q[1,1]:=R[1];Q[1,2]:=R[2];Q[1,3]:=w;
  znam:=j:=1;
  go to ucx sie;
end;
ktore j:
for j:=1 step 1 until znam do
if Q[j,1]=R[1] ^ Q[j,2]=R[2] ^ Q[j,3]=w then
begin
  if kb on A ster=49 then
  begin
    drumplace:=4095-(j-1)x(n+1);
    from buf(Y);
    writetr;
    writetext(<<R>);write(<ad>,j);writetext(<<+>);
    for i:=0 step 1 until n do
      write(<ad>,Y[i],writetext(<<+>));
    writetext(<<+>);
    go to start;
  end;
  go to ucx sie;
end;
if kb on A ster=37 then
begin
  writetr;
  writetext(<<Nie znam wskazanego symbolu >);
  go to start;
end;
znam:=j:=znam+1;
Q[j,1]:=R[1];Q[j,2]:=R[2];Q[j,3]:=w;

ucx sie:
na:= na+1;
adres:=4095-(j-1)x(n+1);
drumplace:=adres;
from buf(Y);
Y[0]:=Y[0]+1;

for i:=1 step 1 until n do
  Y[i]:=Y[i]+C[i];
drumplace:=adres;

```

```

to buf(Y);
if r=1 then
begin
  r:=0;
  go to start;
end;
if ster=49 ∧ kb on then go to czechy else go to start;

```

rozpoznawanie:

```

pam:=0;
if kb on then
begin
  if roz = 0 then go to start;
  writecr;
  if ster=41 then
  begin
    writetext(⟨Kontrola rozpoznawania:
Y[0] roz rd roz1 roz2 roz3 roz4 rdw roz1w roz2w roz3w
roz4w ⟩);
    end;
    writecr;
    drumplace:=4095-(adr-1)×(n+1);
    from buf(Y);
    write(⟨add⟩,Y[0]); writetext(⟨⟨
write(⟨add⟩,roz),⟨⟨
write(⟨add⟩,rd),⟨⟨
write(⟨add⟩,roz1),⟨⟨
write(⟨add⟩,roz2),⟨⟨
write(⟨add⟩,roz3),
⟨⟨
write(⟨add⟩,roz4),⟨⟨
write(⟨add.0⟩,rd/roz×100),
⟨⟨
write(⟨add.d⟩,roz1/roz×100),⟨⟨
write(⟨add.d⟩,roz2/roz×100),⟨⟨
write(⟨add.d⟩,roz3/roz×100),⟨⟨
write(⟨add.d⟩,roz4/roz×100));
    go to start;
  end;
  roz:=roz+1;
  for i:=1 step 1 until n do
  pam:=pam+C[i];
  if pam=0 then
  begin
    writetext(⟨
Nie ma nic ⟩);
    go to start;
  end;

```

```

pam:=0;
for j:=1 step 1 until znem do
begin
  sum:=0;
  drumplace:=4095-(j-1)*(n+1);
  from buf(Y);
  po:=Y[0];  nl:=n;
  for i:=1 step 1 until n do
  if Y[i] > entier(Y[0]*gg + 0.5) v Y[i] < entier(Y[0]*gd+0.5)
  then
  begin
    if C[i]=1 then sum:=sum+Y[i] else sum:=sum+po-Y[i];
  end
  else
  nl:=nl - 1;
  pam:=sum/(po*nl);
  if pam-pum < 0 then
  begin
    pam1:=pam;
    pam:=pam;
    adr:=adr;
    adr:=j;
  end
  else
  if pam - pum < pam - pam1 then
  begin
    pam1:= pam;
    adr1:= j;
  end;
end;

if pam>proc 1 then
begin
  writecr;
  writetext(«Jest to symbol ,,»);
  j:=0;
AA:
  R[1]:=Q(adr,1);R[2]:=Q(adr,2);w:=Q(adr,3);
  for i:=1,2,3,4,5,6,7 do
  begin
    Z[i]:=entier(R[(if i<3 then 1 else 2)/64]*(if i<3 then 1-1
      else 1-4)));
    Z[i]:=(Z[i]/64- entier(Z[i]/64))*64;
  end;

```

```

if w=1 then writechar(60);
for i:=1,2,3,4,5,6,7 do
begin
  if Z[i]=0 then go to 14;
  writechar(Z[i]);
end;
14:
writechar(58);
writetext(⟨<, p=⟩);write(⟨n.dd0⟩,pam);
if j=1 then
begin
  adr:=adr2;
  go to CC;
end;
BB:
if pam-pam1 ≤ delta then
begin
  writetext(⟨< lub ,⟩);
  j:=1; adr2:=adr;
  adr:=adr1; pam:=pam1;
  go to AA;
end;
CC:
j:=0; roz1:= roz1+1;
DD:
writecr;
writetext(⟨<Czy dobrze: ⟩);
odp:=typechar;if odp=19 then
begin
  rd:=rd+1;
  j:=adr;
  go to ucz sie;
end;
if odp=20 then go to wskazanie symbolu;
go to start;
end;
if pam<proc 1 ∧ pam>proc 2 then
begin
  writecr;
  writetext(⟨<roz2 (⟩);
  if w=1 then writechar(60);
  for i:=1,2,3,4,5,6,7 do
  begin
    if Z[i]=0 then go to 15;
    writechar(Z[i]);
  end;

```

```

15 : writechar(58);
    writetext(⟨,p=⟨,write(⟨n.dd0⟩,pam));
    roz2:=roz2+1;
    go to DD;
end;
if pam<proc 2 ∧ pam>proc 3 then
begin
writetext(⟨roz3 (⟩);
if w=1 then writechar(60);
for i:=1,2,3,4,5,6,7 do
begin
if Z[i]=0 then go to 16;
writechar(Z[i]);
go to DD;
end;
end;
16: writechar(58);
    writetext(⟨,p=⟨,write(⟨n.dd0⟩,pam));
    roz3:=roz3+1;
    go to start;
end;
if pam<proc3 then
begin
writetext(⟨roz4,⟨,write(⟨n.dd0⟩,pam));
roz4:=roz4+1;
go to DD;
end;
go to start;
koniec:
writetext(⟨

```

K O N I E C

DZIEKUJE PANSTWU

Do widzenia ⟩);

end

end programu rozpozn.

Spis treści

	Str.
1. Uwagi wstępne	3
2. Ogólny opis koncepcji urządzenia rozpoznającego oraz przyjęte założenia początkowe	5
2.1. Schemat blokowy urządzenia oraz opis działania i współpracy bloków	5
2.2. Przyjęte założenia początkowe	7
3. Układ recepcyjny	7
3.1. Krótki opis podstawowej idei budowy układu	7
3.2. Oznaczenia operacji logicznych. Definicje wprowadzonych operatorów	8
3.3. Opis matematyczny układu recepcyjnego	10
3.3.1. Schemat blokowy	10
3.3.2. Warstwa A	11
3.3.3. Warstwa B	12
3.3.4. Warstwa D	12
3.3.5. Warstwa S	16
3.3.6. Warstwa C	18
3.4. Interpretacja elementów macierzy cech C	19
3.5. Charakterystyka ogólna układu	22
4. Układ asocjacyjny	23
4.1. Sieć ucząca się. Opis algorytmu uczenia	23
4.2. Układ decyzyjny. Opis algorytmu rozpoznawania	25
4.3. Interpretacja geometryczna algorytmu uczenia i rozpoznawania	27
5. Opis programu modelującego urządzenie rozpoznające na maszynie cyfrowej oraz wyniki badań modelu	31
5.1. Opis przyjętych oznaczeń w schemacie operacyjnym	32
5.2. Opis najważniejszych zmiennych i macierzy używanych w programie	32
5.3. Opis działań maszyny	34
5.4. Wyniki badań	37
5.4.1. Opis danych do badań	37
5.4.2. Uczenie modelu	37
5.4.3. Ustalenie wartości gg oraz gd	44

	Str.
5.4.4. Ustalenie wartości proc1, proc2, proc3	48
5.4.5. Ustalenie wartości delta	49
5.4.6. Badania zdolności rozpoznawania	49
5.4.6.1. Czas rozpoznawania	49
5.4.6.2. Zależność poprawności rozpoznawania od długości ciągów uczących oraz inne badania	50
5.4.7. Wnioski	68
6. Zakończenie	69
Literatura	71

Errata

Str.	Wiersz	Jest	Powinno być
2	2g	ОСНАКОВАНИЯ	ОПОВНАВАННЯ
2	9d	of characters the	of characters. The
6	11g	urzącej się	urzącej się
6	12d	urząca	urząca
7	4g	urzącej się	urzącej się
9	7d	$b_{ij}^0, i=1,2,\dots,n$	$[b_{ij}^0]_{i=1,2,\dots,n}$
10	2g	$k \in \{i, j\}$	$k \in \{i, j\}$
10	5d	bloki D^1	bloki D^1
13	wzór	$\begin{bmatrix} t_{1,1}^3, \dots, t_{1,4}^3 \\ \hline t_{2,1}^3, \dots, t_{2,4}^3 \\ \vdots \\ t_{9,1}^3, \dots, t_{5,4}^3 \\ \hline t_{10,1}^3, \dots, t_{6,4}^3 \end{bmatrix}$	$\begin{bmatrix} t_{1,1}^3, \dots, t_{1,4}^3 \\ \hline t_{2,1}^3, \dots, t_{2,4}^3 \\ \vdots \\ t_{9,1}^3, \dots, t_{5,4}^3 \\ \hline t_{10,1}^3, \dots, t_{6,4}^3 \end{bmatrix}$
14	wzór	$\begin{bmatrix} t_{1,1}^4 & t_{1,2}, \dots, t_{1,6} & t_{1,7} \\ \vdots & & \vdots \\ t_{5,1} & t_{5,2}, \dots, t_{5,6} & t_{5,7} \end{bmatrix}$ ${}^*B^2 \rightarrow$	$\begin{bmatrix} t_{1,1}^4 & t_{1,2}^4, \dots, t_{1,6}^4 & t_{1,7}^4 \\ \vdots & & \vdots \\ t_{5,1}^4 & t_{5,2}^4, \dots, t_{5,6}^4 & t_{5,7}^4 \end{bmatrix}$ ${}^*B^2 \text{ (zob. rys. 7)}$
19	12g	$s_{4,1}^4 = 2 \wedge s_{4,1}^7 \geq 3$	$v_{4,1}^4 = 2 \wedge s_{4,1}^7 \geq 3$
21	3g-27g	przy opisach linii we wszystkich wyrażeniach w na-	
22	1g-10g	wiasach, np. $(1 \geq 5)$, zamiast 1 powinno być 1	
23	6d	$c_1^k = c_{11}^k, c_{12}^k, \dots, c_{1n}^k$	$c_1^k = [c_{11}^k, c_{12}^k, \dots, c_{1n}^k]$
25	5d	$c_j \in \{0, 1\}$	$c_j \in \{0, 1\}$
28	7g	wyprowadzona	wprowadzona
28	wzór (13a)	$\sum_{i=1}^k 1 \quad x_i - y_i$	$\sum_{i=1}^{k \times 1} x_i - y_i $
30	wzór (15a)	$\sum_j c_j - w_{ij} $	$\sum_j c_j - w_{ij} $
31	3g	$(0 \leq p_1(0, w_1) \leq 1)$	$(0 \leq p_1(0, w_1) \leq 1)$
32	7g	(a)	[a]
37	7g	ok. 900 różnych symboli	ok. 900 różnych ręcznie pisanym symboli
44	1g	przedstawiono wydruki	przedstawiono niektóre wy- druki