

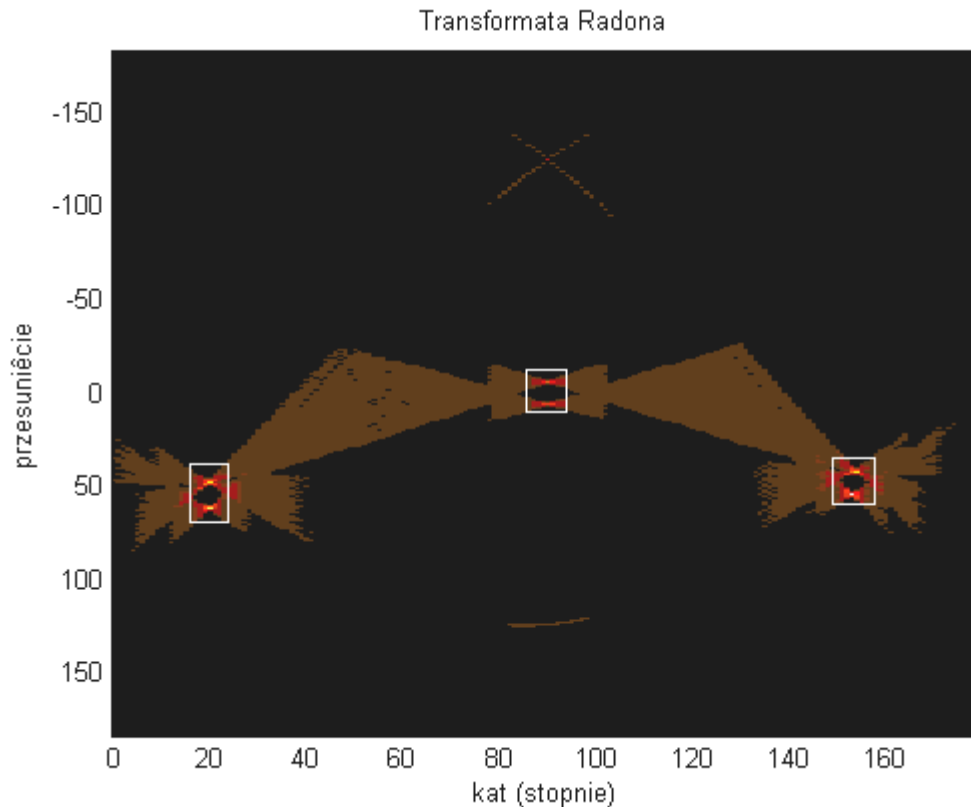
UNIwersYTET ZIELONOGÓRSKI	WYKONALI:	GRUPA		OCENA:
<i>Rozpoznawanie obrazów</i>				
ĆWICZENIE NUMER: 6	TEMAT: Klasyfikator	DATA WYKONANIA:	DATA ODDANIA:	PODPIS:

1. Cel ćwiczenia

Nasze zadanie polegało na zbudowaniu systemu klasyfikatora, który pozwoliłby automatycznie rozpoznawać znaki alfanumeryczne. Zadanie składało się z dwóch części:

- a) badanie skuteczności klasyfikatora
- b) wpływ szumu na skuteczność rozpoznawania

2. Budowa klasyfikatora.



Rys.1 Przykładowy wykres po transformacji Radona

Na rys.1 wyjaśniamy metodę którą wykorzystaliśmy do budowy klasyfikatora. Jest to przykładowy wykres tablicy cech pewnej litery. Na osi poziomej mamy wartości kątów a na osi pionowej wartości przesunięć. Białe prostokąty oznaczają tolerancję w

jakiej może się znaleźć maksimum aby zostało zaklasyfikowane prawidłowo. Dla każdej litery alfabetu zamieszczonego na rysunku poniżej stworzyliśmy osobny zestaw reguł. Prostokąty mają odpowiednio większy rozmiar niż epicentrum, ponieważ w ten sposób nasza metoda staje się odporna na drobne przesunięcia oraz pochylenia wzorcowych znaków.

Metoda rozpoznawania znaków, polega na badaniu czy punkt maksymalny, znajduje się poniżej lewego górnego a zarazem powyżej prawego dolnego punktu białego prostokąta. Jeśli wszystkie maksima wypełniają takie prostokąty, wtedy możemy przypuszczać że badany znak jest zaklasyfikowaną wcześniej literą.



Rys.2 Zestaw wzorcowych znaków użytych do budowy klasyfikatora

3. Testowanie klasyfikatora

Nasz klasyfikator przetestowaliśmy dla następujących zbiorów znaków:



Rys.3 Litery_0 (wzorcowy zestaw)



Rys.4 Litery_1 (testowy zestaw)



Rys.5 Litery_2 (testowy zestaw)



Rys.6 Litery_3 (testowy zestaw)



Rys.7 Litery_4 (testowy zestaw)



Rys.8 Litery_5 (testowy zestaw)

Wyniki naszego eksperymentu zamieściliśmy w tabeli poniżej. Zawiera ona informacje na temat ilości poprawnie rozpoznanych znaków.

Alfabet	Wynik ilościowy	Wynik procentowy
<i>litery_0 (wzorzec)</i>	14	93,3%
<i>litery_1</i>	3	20%
<i>litery_2</i>	2	13,3%
<i>litery_3</i>	0	0%
<i>litery_4</i>	0	0%
<i>litery_5</i>	0	0%

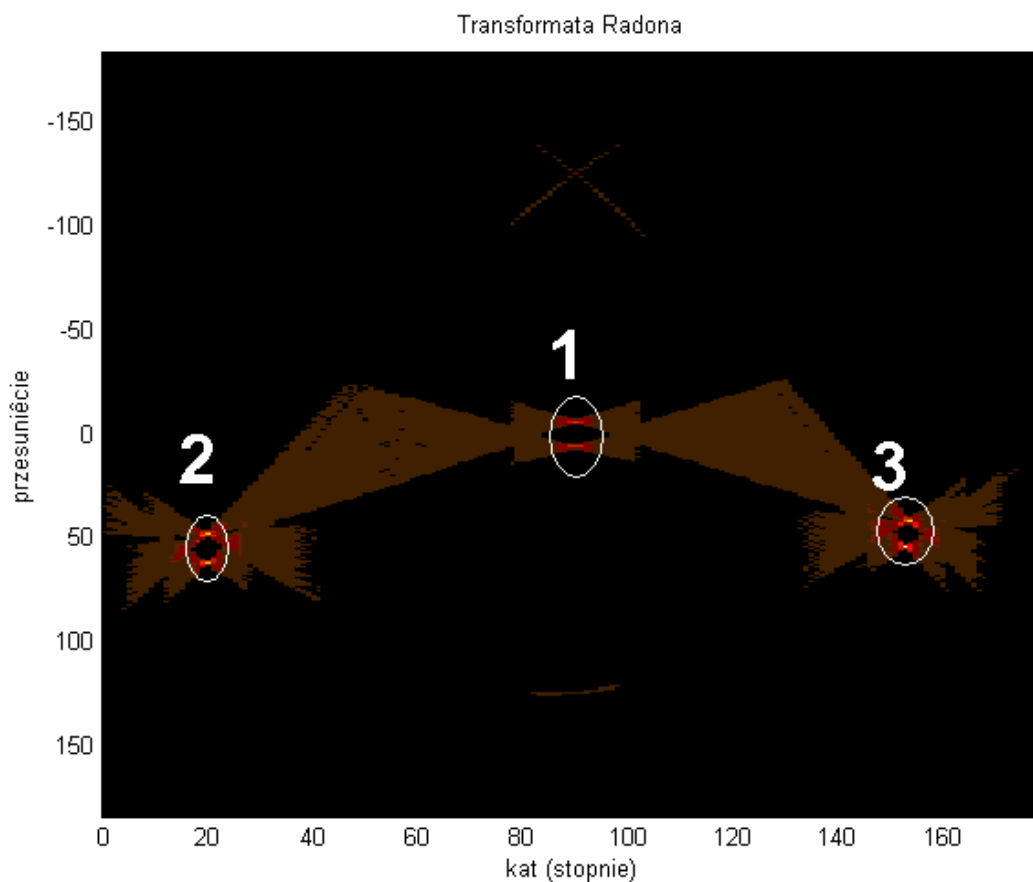
Tabela 1 Wyniki ilościowe i procentowe klasyfikatora

4. Teoretyczne opracowanie uniwersalnego klasyfikatora.

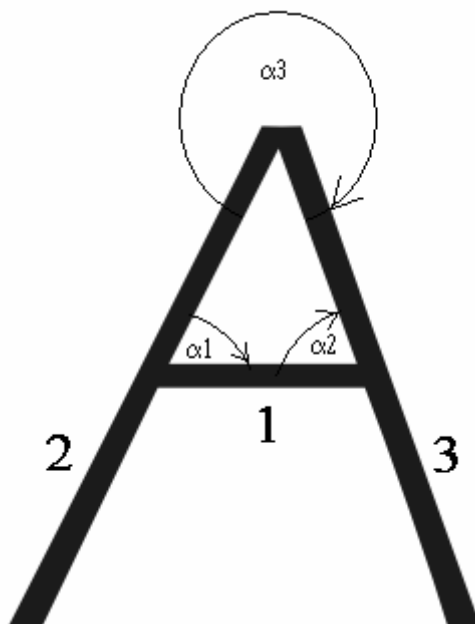
Jasne jest iż przedstawiony sposób budowy i zasady działania naszego klasyfikatora zawiera tak wiele braków że nie nadaje się do normalnej pracy i nie można go nigdzie wykorzystać ze względu na słabą skuteczność.

Pracę takiego klasyfikatora można jednak znacznie poprawić, przez zmianę zasady jego działania.

Spójrzmy na rysunek poniżej. Jest to wynik transformaty Radona dla literki 'A'. Trzy punkty oznaczone cyframi (1,2,3) oznaczają linie z których składa się literka.



Wykorzystując znajomość kątów pod jakimi są nachylone poszczególne proste (1-20°, 2-90°, 3-155°), możemy obliczyć kąty pomiędzy tymi liniami. Przykład zamiesciliśmy na rysunku poniżej.



$$\alpha_1 = \text{abs}(20^\circ - 90^\circ), \quad \alpha_2 = \text{abs}(90^\circ - 155^\circ), \quad \alpha_3 = \text{abs}(20^\circ - 155^\circ)$$

Z wykresu transformaty Radona odczytujemy wartości kątów nachylenia poszczególnych prostych i obliczamy kąty pomiędzy liniami jak w przypadku liter 'A' powyżej. Pamiętać musimy że kąt rysujemy zawsze zgodnie z ruchem wskazówek zegara, od jednej prostej do drugiej. Rysunek poniżej przedstawia prawidłowo zaznaczony kąt (po lewej) i nie prawidłowo (po prawej).



W ten sposób, mając informację o ilości prostych i zawartych między nimi kątami, pozbedziemy się wady poprzedniej metody, ponieważ nie wykorzystujemy tu wcale informacji o przesunięciu a ponadto takie rozwiązanie będzie odporne na skalowanie i obrót. Odporność na skalowanie i obrót wynika stąd iż podczas tych transformacji nie zmieniają się kąty pomiędzy prostymi.

5. Budowa klasyfikatora piktogramów niewiele różniła się od klasyfikatora alfabetu. Znaki piktogramu były bardziej złożone (posiadały więcej linii) co równoznaczne było z zastosowaniem większej ilości instrukcji warunkowych w naszym skrypcie, ponieważ sprawdzaliśmy kąt nachylenia i przesunięcie każdej linii znaku.. Niektóre z linii były zaokrąglone i okazały się niemożliwe do wykrycia przez system. Mimo tego układ bezbłędnie wykrywał wszystkie znaki przy wartości progu równej 30.

Nieznaczące zaszumienie (10 %) miało niewielki wpływ na działanie pracy układu. Nastąpił spadek skuteczności do 93,3 % (układ nie rozpoznał obrazu nr 14).

Zaszumienie na poziomie 50 % pogorszyło warunki pracy układu do poziomu skuteczności 33%. Układ nie rozpoznał obrazów 5, 11, 12, 14, 15.

Obraz	Oryginał	Szum = 10%	Szum = 50%
1	日	日	日
2	中	中	中
3	古	古	古
4	上	上	上
5	否	否	否
6	里	里	里
7	口	口	口
8	言	言	言
9	只	只	只
10	不	不	不
11	同	同	同
12	于	于	于

13	合	合	合
14	用	用	用
15	后	后	后

Tabela 2 Oryginalne oraz zaszumiane piktogramy

6. WNIOSKI

W pierwszej części ćwiczenia zaprojektowaliśmy system do rozpoznawania znaków. System ten wykorzystywał transformatę Radona, a więc posiadał wadę wykrywania tylko linii prostych. W przypadku liter, takich jak B, C czy O nasz system nie nadawał by się do tego zadania. Nie jest to jedyna wada takiego klasyfikatora. Klasyfikator reaguje prawidłowo tylko na dane wzorcowe, jeśli lekko zmodyfikujemy (przeskalujemy, przesuniemy, zmienimy czcionkę) wzorec, klasyfikator nie jest w stanie udzielić właściwej odpowiedzi. Ponadto jeśli litery umieszczamy dokładnie w centrum obrazu to przy podaniu E na wejście, klasyfikator udzieli odpowiedzi E oraz L, ponieważ znak L zawiera się w E. Podobna sytuacja będzie miała miejsce w przypadku liter T oraz I.

Można uniknąć kilku wad przez zbudowanie klasyfikatora opartego na zasadzie jaką przedstawiliśmy punkcie 4, jednak nie rozwiązuje to wszystkich problemów.

Naszym zdaniem klasyfikator oparty na transformacie Radona, na pewno nie nadaje się do budowy systemu rozpoznawania znaków, posiada kilka wad które wydają się być trudne do ominięcia (np. nakładanie się podobnych znaków).

W drugiej części ćwiczenia, zajęliśmy się badaniem wpływu zaszumienia na pracę klasyfikatora. Mimo bardzo zbliżonych kształtów znaków wzorcowych, udało nam się uzyskać 100%-ową poprawność rozpoznawania. W zależności od wielkości zaszumienia klasyfikator radził sobie całkiem dobrze, przy 50% zaszumieniu rozpoznał aż 33% znaków. Natomiast przy zaszumieniu wielkości 10% system poprawnie rozpoznał 93,3% znaków.

Można zatem wnioskować iż system rozpoznawania znaków oparty na transformacie Radona jest odporny na zaszumienie obrazu podawanego na wejście, jednak wady części pierwszej ćwiczenia, przykrywają tą zaletę.