POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH
INSTYTUT AUTOMATYKI I INFORMATYKI STOSOWANEJ

mgr inż. Ewa Snitkowska

ANALIZA TEKSTUR W OBRAZACH CYFROWYCH I JEJ ZASTOSOWANIE DO OBRAZÓW ANGIOGRAFICZNYCH

Promotor:
dr hab. Włodzimierz Kasprzak

Warszawa 2004

Spis treści

1. WPROWADZENIE ... 3

1.1. CEL PRACY ... 3
1.2. PODSTAWY ANALIZY TEKSTUR ... 6
1.2.1 Pojęcie tekstury ... 6
1.2.2 Parametry statystyczne pierwszego rzędu ... 7
1.2.3 Histogramy sum i różnic ... 9
1.2.4 Macierz spójności wartości jasności ... 11

2. HEURYSTYCZNE PRZEKSZTAŁCENIA OBRAZU ... 15

2.1. HEURYSTYCZNE LINIOWE PRZEKSZTAŁCENIA WZORCA ... 15
2.1.1 Baza ortogonalna ... 15
2.1.2 Transformata Fouriera (DFT) ... 16
2.1.3 Przekształcenie Walsha-Hadamarda ... 17
2.1.4 Transformata falkowa ... 19
2.1.5 Funkcje Gabora ... 21
2.2. OPIS TEKSTURY W MPEG-7 (DESKRYPTORY) ... 22
2.2.1 Wyznaczanie energii ... 23
2.2.2 Deskryptor tekstury jednorodnej ... 24
2.2.3 Deskryptor przeglądania tekstury ... 27
2.2.4 Deskryptory tekstury w analizie obrazów ... 30

3. ANALITYCZNE PRZEKSZTAŁCENIA OBRAZU ... 32

3.1. PRZEKSZTAŁCENIE PCA ... 32
3.2. ICA ... 34
3.3. ANALIZA CZYNNIKOWA (*FACTOR ANALYSIS*) ... 37

4. DETEKCJA CECH DZIĘKI DEKOMPOZYCJI ICA ... 39

4.1. OPIS METODY ... 39
4.2. PROCES UCZENIA WEKTORÓW BAZOWYCH ... 41
4.2.1 Przetwarzanie wstępne ... 41
4.2.2 Algorytm ICA ... 43
4.3. DETEKCJA CECH A ANALIZA OBRAZU I KLASYFIKACJA. ... 44

5. OBRAZY ANGIOGRAFICZNE ... 48

6. KLASYFIKATORY CECH ... 51

6.1. PROBLEM KLASYFIKACJI ... 51
6.1.1 Podstawowe typy klasyfikatorów numerycznych ... 52
6.2. KLASYFIKATOR STATYSTYCZNY BAYESA ... 53
6.2.1 Uczenie klasyfikatora ... 54
6.2.2 Reguła decyzyjna ... 55
6.2.3 Klasyfikator według minimalnej odległości ... 56
6.3. KLASYFIKATOR „*K* NAJBLIŻSZYCH SĄSIADÓW” ... 57
6.4. KLASYFIKATOR SVM ... 58
6.4.1 Funkcje kosztu empirycznego ... 59
6.4.2 Próbki separowalne liniowo ... 60
6.4.3 Próbki nie separowalne liniowo ... 63
6.5. KLASYFIKATOR NEURONOWY ... 64
6.5.1 Klasyfikator zastosowany w testach ... 67

7. EKSPERYMENTY DOTYCZĄCE DETEKCJI CECH TEKSTUR ... 69

7.1. OBRAZY TESTOWE ... 69
7.1.1 Tekstury z albumu Brodatz’a ... 69
7.1.2 Obrazy angiograficzne ... 70
7.2. WYNIKI DETEKCJI CECH ... 71

7.2.1 Wektory cech oparte o filtry Gabora ... *71*
7.2.2 Wektory cech oparte o PCA ... *74*
7.2.3 Wektory cech oparte o ICA ... *76*
7.3. PORÓWNANIE WEKTORÓW CECH ... 80
7.3.1 Kryterium porównywania obszaru cech ... *80*
7.3.2 Wyniki porównania obszarów cech ... *84*
8. EKSPERYMENTY DOTYCZĄCE KLASYFIKACJI TEKSTUR ... 88
8.1. ZAŁOŻENIA ... 88
8.2. KLASYFIKATOR BAYESA ... 89
8.3. KLASYFIKATOR NN ... 91
8.4. KLASYFIKATOR NEURONOWY ... 92
8.5. KLASYFIKATOR SVM ... 94
9. ZAKOŃCZENIE ... 96
PODZIĘKOWANIA ... 99
SPIS RYSUNKÓW ... 100
SPIS TABEL ... 102
BIBLIOGRAFIA ... 103

1. Wprowadzenie

1.1. Cel pracy

Jedną z metod wykorzystywanych w analizie obrazów cyfrowych ([54], [66], [104]) jest analiza tekstury ([7], [105], [112]). Tekstura reprezentuje takie właściwości obrazu jak kierunkowość (kierunek układania się wzorca) czy porowatość. Tekstura jest właściwością niejednorodnego obszaru w obrazie. Na tej podstawie można rozróżnić dwa obrazy od siebie, jak również wyznaczyć w danym obrazie obszary spełniające określone warunki. Teksturę obrazu można symbolicznie opisać poprzez podanie wartości dla skończonego wektora cech. Wymiar tego wektora jest zwykle dużo niższy niż liczba pikseli w opisywanym fragmencie obrazu.

Wyszukiwanie w obrazach cyfrowych obszarów o określonych właściwościach może zostać wykorzystane w różnych zastosowaniach, np. w celu określenia zmian sceny (monitoring), wyszukiwania określonych obrazów z bazy, a także w analizach medycznych. Do określenia charakteru tych obszarów stosuje się różne opisy – od metod czysto statystycznych, poprzez detekcję dyskretnych segmentów, aż po te oparte na ludzkiej percepcji.

Angiografia jest to metoda diagnostyczna wykorzystywana w medycynie, która polega na badaniu kontrastowym naczyń krwionośnych. W przeprowadzonych testach chodzi o określenie przyrostu (lub zanikania) naczyń krwionośnych w tkance pod wpływem badanego preparatu. Tego typu diagnoza wykorzystywana jest m. in. w terapii nowotworów. W

niektórych przypadkach nie jest konieczne dokładne określenie liczby naczyń, a jedynie stwierdzenie zmian. Spodziewamy się, że analiza właściwości tekstury poszczególnych obszarów pozwoli wykryć i sklasyfikować stan naczyń krwionośnych, co ułatwi dalszą analizę, dokonywaną przez specjalistę. Poszukujemy takiej reprezentacji, która pozwoli na skuteczną klasyfikację obszarów niezależnie od obrotu obrazu, zakresu intensywności pikseli czy skali obrazu.

Kolejnym celem tej pracy jest znalezienie możliwie najdokładniejszego klasyfikatora dla opracowanej reprezentacji (cech) tekstury, przy czym weryfikacja jego skuteczności będzie przeprowadzona dla obrazów medycznych. Znane z literatury są główne trzy typy algorytmów:

- cechy oparte o filtry Gabora (deskryptory tekstury MPEG-7) ([42], [91], [107]),
- cechy oparte o PCA (analiza składowych głównych – popularny sposób redukcji informacji) ([16], [85], [99], [106]),
- cechy oparte o tzw. macierz spójności jasności (standardowa metoda opisu tekstur w obrazach) [29].

Głównym celem tej pracy jest wykazanie przydatności podejścia ICA (ang. *Independent Component Analysis* – analiza składowych niezależnych ([13], [37], [49]) do analizy tekstur, w szczególności w przypadku obrazów angiograficznych. Typowe podejścia do detekcji cech – wykorzystywane w problemach reprezentacji lub kompresji obrazu cyfrowego - w oparciu o liniowe przekształcenie [75], takie jak Fouriera, Hadamarda, transformata kosinusowa [85] czy filtry Gabora ([42], [107]) nie są dopasowane do konkretnych obrazów danego rodzaju, gdyż mają uniwersalny charakter. W przeciwieństwie do nich przekształcenia uzyskiwane w wyniku analizy PCA (analiza składowych głównych) lub ICA (przewidziana przez nas analiza składowych niezależnych) są estymowane w oparciu o same dane uczące – przykłady obrazów danego rodzaju.

Z tych dwóch metod PCA przeznaczona jest raczej do kompresji – orientując się na podobieństwa obrazów różnych klas – a ICA powinna nadawać się do klasyfikacji, gdyż orientuje się na różnice pomiędzy obrazami różnych klas [46]. ICA jest odmianą statystycznego „projection pursuit", czyli poszukiwania „ciekawych" (oddających strukturę rozkładów statystycznych) rzutów wielowymiarowych danych. Rozkład Gaussa jest najmniej ciekawy, natomiast kierunki najmniej gaussowskie są „najciekawsze". Obszary w przestrzeni

nie pokrywane przez niezależne składowe są wypełniane przez szum Gaussa - znajdując kierunki nie-gaussowskie znajdujemy efektywnie niezależne składowe, czyli docelowe osie przestrzeni cech. Niniejsza praca ma wykazać, że odpowiednio dobierając (do próbek danych) już same przestrzenie cech tekstury (a nie dopiero na późniejszym etapie klasyfikacji uniwersalnych przestrzeni cech), można stworzyć narzędzie dużo skuteczniejsze w analizie tekstur, niż dotychczas znane metody.

W skład narzędzia programowego, wykorzystanego do testów, wchodzi zarówno detekcja cech, jak i klasyfikator cech. Badane będą podstawowe typy klasyfikatorów: klasyfikator Bayesa, klasyfikator SVM (liniowa maszyna wektorów nośnych) i klasyfikator neuronowy ([22], [73], [82]). Wybrany klasyfikator powinien spełniać następujące warunki:

- pozwolić na wyłonienie największej liczby klas przy zachowaniu wysokiego stopnia poprawności klasyfikacji
- nie popełniać krytycznych błędów klasyfikacji.

W przypadku obrazów angiograficznych analiza tekstury może stanowić etap wstępny do bardziej precyzyjnych obliczeń, np. związanych z krętością naczynia. Aplikacja tego typu może być narzędziem pomocniczym w diagnostyce. Należy przede wszystkim wziąć pod uwagę ryzyko związane z automatyzacją procesu. Wprawdzie podejmowanie decyzji lekarskiej wyłącznie na podstawie wyniki klasyfikatora jest raczej niewskazane, niemniej jednak wspomniany program może być istotnym wsparciem w pracy eksperta, np. jako narzędzie do pre-selekcji.

Tezy rozprawy są następujące:

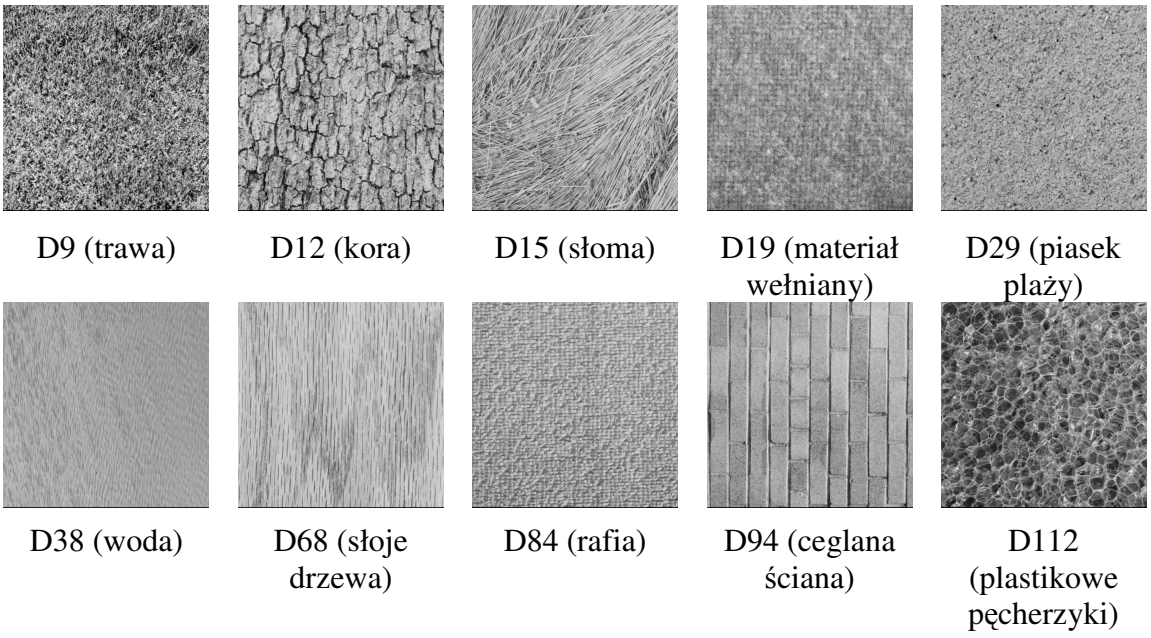
1. Odpowiednio dobierając reprezentację cech tekstury do rzeczywistych danych i klasyfikator, dla którego określono koszty ryzyka błędnej klasyfikacji, można stworzyć narzędzie skuteczne i pomocne w analizie obrazów.
2. Dla wybranego rodzaju obrazów (w tym przypadku obrazy angiograficzne) metoda ICA pozwala na lepszą spójność cech tej samej klasy i rozdział cech różnych klas (niż inne proponowane podejścia), a także ułatwia poprawną klasyfikację.

Praca powstała dzięki badaniom prowadzonym w ramach grantu KBN nr 7T08A05016, dotyczącego analizy obrazów angiograficznych oraz grantu Ministerstwa Nauki i Informatyzacji 3T11A 015 26.

1.2. Podstawy analizy tekstur

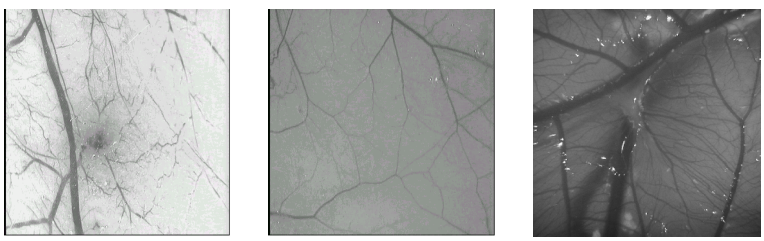
1.2.1 Pojęcie tekstury

Tekstura reprezentuje regularne cechy powierzchni obiektu – patrząc na obraz możemy powiedzieć, czy przedstawia on obiekt gładki (np. powierzchnia szyby) lub chropowaty (np. kora drzewa), a także czy prezentowany wzorzec jest mniej lub bardziej regularny. Człowiek decyduje o rodzaju tekstury na podstawie obserwacji pewnych niewielkich wzorów, których regularne rozmieszczenie pozwala na klasyfikację.



Rysunek 1.1 Przykłady tekstur z albumu Brodatz'a (źródło: [7]).

Popularnymi przykładami tekstury, często wykorzystywanymi w badaniach porównawczych metod analizy tekstury są tekstury z albumu Brodatz'a [7]. Jednak łatwo zauważyć, że obrazy angiograficzne, czyli przedstawiające układ naczyń krwionośnych, różnią się teksturą od typowych obrazów Brodatz'a.



Rysunek 1.2 Przykładowe obrazy angiograficzne: (a) i (b) - tkanka skórna myszy, (c) - zarodek kurzy

Aby scharakteryzować teksturę matematycznie, wprowadzono szereg parametrów obliczanych na podstawie właściwości obrazu cyfrowego. W literaturze można spotkać następujące typy parametrów służących do opisu tekstury:

- statystyczne [27],
- strukturalne [11],
- wykorzystujące techniki przetwarzania sygnałów [57],
- morfologiczne ([12], [64], [68]).

W analizie tekstur stosuje się technikę zwaną granulometrią morfologiczną [28]. Obraz traktowany jest jako zbiór ziaren, które mogą być następnie przesiewane przez zwiększające się oka w swego rodzaju „sicie". Te, które zostaną, mierzone są jako wielkości rozkładu granulometrycznego. Dla potrzeb definicji operacji morfologicznych obrazów w skali szarości obraz I(i,j) traktowany jest jako funkcja dwóch zmiennych, przyjmująca wartości ze zbioru liczb rzeczywistych. Z kolei przesiewanie obrazu przez sito o zmieniających się wielkościach oka realizowane jest w praktyce przez operacje morfologiczne obrazu, czyli zamykanie i otwieranie odpowiednim elementem strukturującym. Sekwencja obrazów, powstających w wyniku takiego przekształcenia, tworzy ciąg malejący. Sekwencję tego typu nazywamy granulometrią [9]. Odwrotnością granulometrii otwierającej jest granulometria zamykająca.

Niniejsza praca nie obejmuje metod morfologicznych. Metody morfologiczne należą do dziedziny segmentacji obrazu, natomiast celem pracy jest analiza metod opartych o transformację przestrzeni reprezentacji obrazu.

1.2.2 Parametry statystyczne pierwszego rzędu

Aby ułatwić opisywanie cech tekstury, wprowadzono szereg parametrów wyliczanych na podstawie obrazu cyfrowego [63]. Parametry te odzwierciedlają liczbowo właściwości tekstury, które można wykorzystać do analizy i klasyfikowania.

Podstawą obliczania parametrów statystycznych pierwszego rzędu jest histogram jasności obrazu, opisującego liczbę pikseli mających zadaną jasność. Obraz o N poziomach jasności

ma histogram o N prążkach. Na tej podstawie wyznaczane jest prawdopodobieństwo wystąpienia danego poziomu jasności w obrazie.

$$P(i) = \frac{H(i)}{n}, \tag{1.1}$$

gdzie:

- $P(i)$ – prawdopodobieństwo wystąpienia danego poziomu jasności i w obrazie,
- $H(i)$ – liczba punktów o jasności i w obrazie,
- n – liczba wszystkich punktów (pikseli).

Na podstawie histogramu można wyznaczyć parametry, które opisują globalne cechy obrazu w danym obszarze i są stosunkowo proste do wyznaczenia ([23], [31], [50], [53], [62]). Nie można ich jednak wykorzystać do określenia wyglądu obrazu (np. stopnia porowatości).

Jednym z podstawowych parametrów obrazu jest wartość średnia (jasności). Określa ona średnią jasność obrazu lub danego obszaru i wyznaczana jest według wzoru:

$$m = \sum_{i=0}^{N} iP(i). \tag{1.2}$$

Kolejnym parametrem wyznaczanym na podstawie histogramu jasności jest odchylenie standardowe, które określa globalną zmienność pikseli danego obrazu. Parametr ten wyznacza się według wzoru:

$$\sigma = \sqrt{\sum_{i=0}^{N} (i-m)^2 P(i)}. \tag{1.3}$$

Skośność histogramu charakteryzuje stopień asymetrii rozkładu jasności wokół jego średniej i określana jest mianem współczynnika asymetrii. Skośność dodatnia oznacza rozkład z asymetrycznym „ogonem" rozciągającym się w kierunku coraz wyższych wartości dodatnich, natomiast skośność ujemna – rozkład z asymetrycznym „ogonem" rozciągającym się w kierunku wartości bliższych zeru. Parametr ten wyznacza się na podstawie zależności:

$$S = \left[\sum_{i=1}^{N-1} (i-m)^3 \cdot P(i) \right] \cdot \frac{1}{\sigma^3}. \tag{1.4}$$

Dwa kolejne parametry nie mają swoich bezpośrednich odpowiedników w statystyce matematycznej, ale zaliczane są do opisywanej grupy ze względu na fakt, że wyznacza się je na podstawie histogramu jasności i charakteryzują one jej rozkład w obrazie.

Energia wyznaczana jest na podstawie wzoru:

$$E = \sum_{i=0}^{N} P^2(i), \tag{1.5}$$

natomiast entropia – z następującego wzoru.

$$Ent = -\sum_{i=0}^{N} P(i) \cdot \log_2 \left(P(i) \right) \tag{1.6}$$

Parametry te charakteryzują jedynie globalne właściwości tekstury w obrazie i są wrażliwe na wszelkie szumy i zakłócenia oraz sposób standaryzacji obrazu.

1.2.3 Histogramy sum i różnic

Innym sposobem jest utworzenie histogramów sum i różnic par pikseli odległych od siebie wzdłuż różnych kierunków. Niech $f_{j;k}$ i $f_{j+v1;k+v2}$ będą dwoma pikselami przesuniętymi o wektor **v**=(v1; v2). Niech suma i różnica dla tych pikseli wynoszą odpowiednio:

$$s_{j;k} = f_{j;k} + f_{j+v1;k+v2} \text{ oraz } d_{j;k} = f_{j;k} - f_{j+v1;k+v2}. \tag{1.7}$$

Obliczmy histogramy dla sum i różnic:

$$H_s(l) = \left| f_{j;k} \mid s_{j;k} = l \right|, \tag{1.8}$$

$$H_d(m) = \left| f_{j;k} \mid d_{j;k} = m \right|, \tag{1.9}$$

gdzie |…| oznacza liczbę elementów (moc) zbioru.

Następnie histogramy te normowane są względem liczby punktów.

$$h_s(l) = \frac{H_s(l)}{\sum_l H_s(l)} \tag{1.10}$$

$$h_d(m) = \frac{H_d(m)}{\sum_m H_d(m)} \tag{1.11}$$

Dla tak przygotowanych histogramów można wyznaczyć wartości cech statystyk opisanych w rozdziale 1.2.2. Przykładowy zestaw ośmiu cech – po cztery cechy wyliczone na podstawie każdego histogramu:

1. wartości średnie

$$c_1 = \sum_{l=0}^{2L-2} l h_s(l),\ c_2 = \sum_{m=-L+1}^{L-1} m h_d(m) \tag{1.12}$$

2. momenty 2-go rzędu

$$c_3 = \sum_{l=0}^{2L-2} (h_s(l))^2\ ,\ c_4 = \sum_{m=-L+1}^{L-1} (h_d(m))^2 \tag{1.13}$$

3. kontrasty

$$c_5 = \sum_{l=0}^{2L-2} (l - c_1)^2 h_s(l),\ c_6 = \sum_{m=-L+1}^{L-1} (m - c_2)^2 h_d(m) \tag{1.14}$$

4. entropie

$$c_7 = \sum_{l=0}^{2L-2} - h_s(l) \mathbf{log}(h_s(l)),\ c_8 = \sum_{m=-L+1}^{L-1} - h_d(m) \mathbf{log}(h_d(m)) \tag{1.15}$$

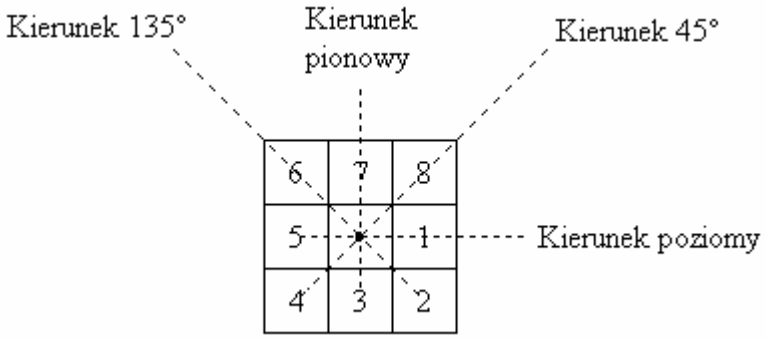
W praktyce wyznacza się pary histogramów dla czterech kierunków wektora **v**, co daje w sumie 32 cechy tekstury.

Liczba cech zależy od wymagań użytkownika – mogą to być np. wartość średnia, odchylenie standardowe, kontrast i energia – zob. przykłady powyżej. Daje to dwa razy większą liczbę parametrów do analizy, przy czym ilość ta zwiększa się z każdym nowym przesunięciem v, co pozwala na poprawienie wyników. W praktyce często stosuje się redukcję liczby poziomów jasności *L*, np. do *L*=32.

1.2.4 Macierz spójności wartości jasności

Parametry statystyczne drugiego rzędu wyznaczane są na podstawie specjalnie skonstruowanych macierzy opisujących zależności w przestrzennym rozkładzie jasności par pikseli w analizowanym obrazie. Podejście to zostało zaprezentowane w pracy [31] i znalazło szerokie zastosowanie w analizie obrazów (np. [96]).

Macierz spójności jasności (MSJ) jest postaci $\mathbf{G}(d; \alpha) = [\mathbf{g}_{\mu\nu}(d; \alpha)]$, gdzie element $\mathbf{g}_{\mu\nu}$ oznacza liczbę par pikseli ($f_{ij}=u$, $f_{kl}=v$), które są odległe o d i wyznaczają kierunek α. Najczęściej przyjmuje się, że każdy punkt (piksel) obrazu ma ośmiu sąsiadów: w kierunku poziomym (0˚), pionowym (90˚), diagonalnym (135˚) i przeciwdiagonalnym (45˚):



Rysunek 1.3 Sąsiedztwo punktów (pikseli) oraz kierunki analizy dla macierzy spójności jasności

Przykład 1.1 [47]

Dla obrazu f o wymiarze 5x5 pikseli i L=4 poziomach jasności mamy macierz MSJ o L^2=4x4 elementach.

$$f \rightarrow \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 2 & 2 & 3 & 3 & 0 \\ 2 & 2 & 2 & 2 & 2 \end{pmatrix} \qquad G(1;0) = \begin{pmatrix} 4 & 4 & 2 & 1 \\ 4 & 4 & 0 & 0 \\ 2 & 0 & 14 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix}$$

Dla elementu $g_{00}(1;0)$ mamy:

$$g_{00}(1,0) \rightarrow (f_{00}, f_{01}), (f_{01}, f_{00}), (f_{10}, f_{11}), (f_{11}, f_{10})$$

Otrzymane w ten sposób macierze są kwadratowe i symetryczne, czyli $\mathbf{g}_{\mu v}(d;\ \alpha) = \mathbf{g}_{vu}(d;\ \alpha)$. Rozmiar takiej macierzy równy jest liczbie poziomów jasności w obrazie. Dla każdego kierunku analizy buduje się odpowiednią macierz.

Aby wyznaczyć wektor cech, należy przeprowadzić normalizację wartości MSJ względem sumy wszystkich jej elementów. Po znormalizowaniu otrzymujemy macierz, której elementami są prawdopodobieństwa P(u,v) występowania w odległości d pikseli o jasnościach u i v. Określmy 4 parametry, które będą pomocne przy wyznaczaniu cech – są to:

- suma wierszy znormalizowanej macierzy

$$P_x(u) = \sum_{v=0}^{N-1} P(u,v)\,, \tag{1.16}$$

- suma kolumn znormalizowanej macierzy

$$P_y(v) = \sum_{u=0}^{N-1} P(u,v)\,, \tag{1.17}$$

- rozkład sumacyjny znormalizowanej macierzy

$$P_{x+y}(l) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} P(u,v)\text{, gdy } u+v=l\,,\ l = 0,\ 1,\ \ldots,\ 2(N\text{-}1), \tag{1.18}$$

- rozkład różnicowy znormalizowanej macierzy

$$P_{x-y}(l) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} P(u,v)\text{, gdy } |u-v|=l\,,\ l = 0,\ 1,\ \ldots,\ 2(N\text{-}1). \tag{1.19}$$

Korzystając z powyższych parametrów dla znormalizowanej macierzy można wyznaczyć 14 cech:

1. Drugi moment zwykły

$$f_1 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\left[P(u,v)\right]^2 \tag{1.20}$$

2. Kontrast

$$f_2 = \sum_{k=0}^{N-1} k^2 P_{x-y}(k) \tag{1.21}$$

3. Korelacja

$$f_3 = \frac{\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}(u \cdot v) \cdot P(u,v) - \mu_x \mu_y}{\sigma_x \sigma_y}, \tag{1.22}$$

gdzie μ i σ oznaczają wartość średnią i odchylenie standardowe odpowiednich rozkładów brzegowych.

4. Suma kwadratów – wariancja

$$f_4 = \sum_{u=0}^{N-1}\sum_{v-0}^{N-1}(u-v)^2 \cdot P(u,v) \tag{1.23}$$

5. Odwrotny moment różnicowy

$$f_5 = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\frac{1}{1+(u+v)^2} P(u,v) \tag{1.24}$$

6. Suma średnich

$$f_6 = \sum_{i=0}^{2(N-1)} i \cdot P_{x+y}(i) \tag{1.25}$$

7. Entropia sumacyjna

$$f_7 = -\sum_{i=0}^{2(N-1)} P_{x+y}(i) \cdot \log\left[P_{x+y}(i)\right] \tag{1.26}$$

8. Wariancja sumacyjna

$$f_8 = \sum_{i=0}^{2(N-1)} (i - f_7) \cdot P_{x+y}(i) \tag{1.27}$$

9. Entropia

$$f_9 = -\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} P(u,v) \cdot \log\left[P(u,v)\right] \tag{1.28}$$

10. Entropia różnicowa

$$f_{10} = -\sum_{i=0}^{2(N-1)} P_{x-y}(i) \cdot \log\left[P_{x-y}(i)\right] \tag{1.29}$$

11. Wariancja różnicowa

$$f_{11} = \sum_{i=0}^{2(N-1)} (i - f_{10}) \cdot P_{x-y}(i) \tag{1.30}$$

12. Korelacyjne mierniki informacji

$$f_{12} = \frac{h_{xy} - h1_{xy}}{\max(h_x, h_y)}, \tag{1.31}$$

$$f_{13} = \sqrt{1 - \exp\left[-2 \cdot (h2_{xy} - h_{xy})\right]}, \tag{1.32}$$

gdzie:

$h_{xy} = f_9$,

$h1_{xy} = -\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} P(u,v) \cdot \log\left[P_x(v) \cdot P_y(u)\right]$,

$h2_{xy} = -\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} P_x(v) \cdot P_y(u) \cdot \log\left[P_x(v) \cdot P_y(u)\right]$,

$h_x = -\sum_{v=0}^{N-1} P_x(v) \cdot \log\left[P_x(v)\right]$,

$h_y = -\sum_{u=0}^{N-1} P_y(u) \cdot \log\left[P_y(u)\right]$.

13. Maksymalny współczynnik korelacji

$$f_{14} = \sqrt{\lambda_{2\max}}\,. \tag{1.33}$$

Symbol λ_{2max} oznacza drugą największą wartość własną macierzy:

$$C(u,v) = \sum_{k=0}^{N-1} \frac{P(u,k) \cdot P(v,k)}{P_x(v) \cdot P_y(u)}\,. \tag{1.34}$$

Statystyczną klasyfikację można również przeprowadzić bez wyznaczania powyższych cech, na samej macierzy **G**, uśrednionej po dyskretnych kierunkach. Nie wszystkie cechy mają swoje odzwierciedlenie w wyglądzie obrazu. Jedną z cech, która może decydować o wrażeniach wzrokowych, jest kierunkowy moment drugiego rzędu (f1), który określa jednorodność obrazów (im większa liczba poziomów jasności, tym mniejsza wartość cechy, a więc dla obrazów gładkich o małych wahaniach jasności cecha ta przyjmuje większe wartości). Od zakresu poziomów jasności zależy również kontrast (f2) – przyjmuje większe wartości dla szerszych zakresów. Entropia zaś określa stopień skomplikowania obrazu – im więcej szczegółów jest widoczne w teksturze, tym większą wartość przyjmuje ta cecha.

2. Heurystyczne przekształcenia obrazu

W tym rozdziale zaprezentowane zostaną podejścia służące do wyznaczania cech obszaru obrazu, należące do grupy przekształceń heurystycznych. Heurystyczne przekształcenia przestrzeni wykorzystują, dobrze zdefiniowane na potrzeby reprezentacji, funkcje matematyczne i bazują na zrozumieniu problemu, eksperymentach i doświadczeniu. Metody te jednak nie optymalizują żadnego kryterium zależnego od danych reprezentowanych w tych przestrzeniach.

2.1. Heurystyczne liniowe przekształcenia wzorca

2.1.1 Baza ortogonalna

Przekształcenie ortogonalne odpowiada rozwinięciu wzorca $f(x)$ w ortogonalnym systemie funkcji $\varphi_v(x)$ lub rozwinięcie wektora pomiarowego $\boldsymbol{f}$ w ortogonalnej przestrzeni wyznaczonej przez wektory $\boldsymbol{\varphi}_v$ ([52], [67]).

$$c_v = \boldsymbol{\varphi}_v^T \mathbf{f} = \sum_j \varphi_{vj} f_j \text{ lub } \mathbf{c} = \boldsymbol{\Phi} \cdot \mathbf{f} \tag{2.1}$$

Można powiedzieć, że wymiar wektora **c** jest mniejszy lub równy wymiarowi wektora **f**, ale w procesie detekcji cech dążymy do sytuacji, gdy wymiar wektora **c** będzie dużo niższy niż wymiar wektora pomiarowego.

Załóżmy, że istnieje przekształcenie „pseudo-odwrotne" wyrażające się wzorem:

$$\hat{\mathbf{f}} = \Phi^T \mathbf{c} = \sum_{v=1}^{n} c_v \varphi_v \text{ , gdzie } \Phi\Phi^T = \mathbf{I} \tag{2.2}$$

Dążymy to takiej postaci **Φ**, dla której następuje minimalizacja błędu średniokwadratowego aproksymacji **f** przez $\hat{\mathbf{f}}$:

$$\varepsilon = \left(\mathbf{f} - \hat{\mathbf{f}}\right)^T \left(\mathbf{f} - \hat{\mathbf{f}}\right) \tag{2.3}$$

Jednym ze sposobów na wyprowadzenie powyższego jest zastosowanie zasady ortogonalności ([52], [67]). W tym przypadku otrzymujemy:

$$\begin{aligned} &\Phi\left(f - \Phi^T c\right) = 0, \\ &\Phi\Phi^T \cdot c = \Phi f, \\ &c = \Phi f \end{aligned} \tag{2.4}$$

Ostatnie przekształcenie wynika z ortonormalności wektorów bazowych:

$$\varphi_v^T \cdot \varphi_\mu = \begin{cases} 1 & \mu = v \\ 0 & przeciwnie \end{cases}$$

Wektory bazowe są wprawdzie uporządkowane, ale ich wyznaczenie nie zależy od konkretnych danych, nie jest więc zupełnie prawdziwe stwierdzenie, że *n* początkowych współczynników rozwinięcia stanowi najlepsze (lub najgorsze cechy). Znanych jest wiele przekształceń ortogonalnych, np. Fouriera, Hadamarda-Walsha, Haara [67].

2.1.2 Transformata Fouriera (DFT)

Dwu-wymiarowa dyskretna transformata Fouriera przedstawia się następująco ([85], [100]):

$$F_{\mu v} = DFT\left\{\left[\tilde{f}_{jk}\right]\right\} = \sum_{j=0}^{M_x-1}\sum_{k=0}^{M_y-1} \tilde{f}_{jk} \exp\left[-i2\pi\left(\frac{\mu j}{M_x} - \frac{vk}{M_y}\right)\right], \tag{2.5}$$

gdzie μ, ν=0, ±1, ±2, …

Dla przypadku wektorowych próbek rozpatrzona zostanie 1-wymiarowa DFT, dla której funkcja wykładnicza we wzorze (2.5) zostanie oznaczona jako $W_M = \mathbf{exp}\left[\frac{-i2\pi}{M}\right]$. Wówczas 1-wymiarowa DFT wynosi:

$$F_v = \sum_{j=0}^{M-1} f_j W_M^{jv} \tag{2.6}$$

Przyjmując następnie wektor $\varphi_v = \left(W_M^0, W_M^v, W_M^{2v}, ..., W_M^{(M-1)v}\right)^T$ można zauważyć, że zachodzi $F_v = \varphi_v^T f$, co jest zgodne ze wzorem (2.1).

Aby określić cechy w przestrzeni jednowymiarowej DFT, przyjmujemy moduły współczynników przekształcenia Fouriera jako wartości rzeczywiste:

$$c_v = \left|F_v\right|^2 = F_v F_v^* \tag{2.7}$$

2.1.3 Przekształcenie Walsha-Hadamarda

Ciągłe funkcje Walsha definiowane są rekursywnie jako:

$$wal(x;2j+p) = (-1)^{\lfloor j/2 \rfloor + p}\left\{wal\left(2\left(x+\frac{1}{4}\right);j\right) + (-1)^{j+p}\, wal\left(2\left(x-\frac{1}{4}\right);j\right)\right\}$$

$$wal(x;0) = \begin{cases} 1 & -\frac{1}{2} \le x \le \frac{1}{2} \\ 0 & przeciwnie \end{cases} \tag{2.8}$$

dla $j = 0, 1, 2, \ldots$ oraz $p = 0,1$.

Funkcje zdefiniowane powyżej ograniczone są do dziedziny (-0.5, 0.5). Funkcje Walsha są ortogonalne, tzn.:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} wal(x;j)\, wal(x;k) = \begin{cases} 1 & j = k \\ 0 & przeciwnie \end{cases} \tag{2.9}$$

Przykładem dyskretnej wersji funkcji Walsha jest przekształcenie Walsha-Hadamarda ([74], [84]). Wektory $\boldsymbol{\varphi}_v$ otrzymujemy poprzez próbkowanie funkcji Walsha w przedziale (-0.5, 0.5), w wyniku czego otrzymywane jest $M=2^q$ wektorów próbek.

Macierz przekształcenia o rozmiarze M^2 można obliczyć rekursywnie z macierzy Hadamarda

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{H}_M = \mathbf{H}_2 \otimes \mathbf{H}_{M/2} = \mathbf{H}_2 \otimes \mathbf{H}_2 \otimes \ldots \otimes \mathbf{H}_2 \ (q \text{ czynników}).$$

Symbol $\otimes$ oznacza produkt Kroneckera dwóch macierzy.

Przykład 2.1

Dla macierzy **A** o wymiarach M^2 i macierzy **B** o wymiarach m^2 otrzymujemy macierz $(Mm)^2$, np. dla $\mathbf{H}_8$:

$$\mathbf{H}_8 = \mathbf{H}_2 \otimes (\mathbf{H}_2 \otimes \mathbf{H}_2) = (\mathbf{H}_2 \otimes \mathbf{H}_2) \otimes \mathbf{H}_2 =$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \otimes \mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Uporządkowane macierzą Hadamarda przekształcenie Walsha-Hadamarda HWH zastosowane dla wzorca **f** o M elementach dostarcza wektor cech

$$\mathbf{c} = \mathbf{H}_M \mathbf{f} = HWH(\mathbf{f}) \tag{2.10}$$

Własności przekształcenia HWH:

- wymagane są jedynie operacje dodawania i odejmowania na argumentach rzeczywistych,
- wynikiem przekształcenia są tylko wartości rzeczywiste,

- istnieje szybki algorytm obliczenia tego przekształcenia (szybka transformata Walsha [74]) – podobnie jak w przypadku FFT uzyskana przez faktoryzację macierzy $\mathbf{H}_M$.

2.1.4 Transformata falkowa

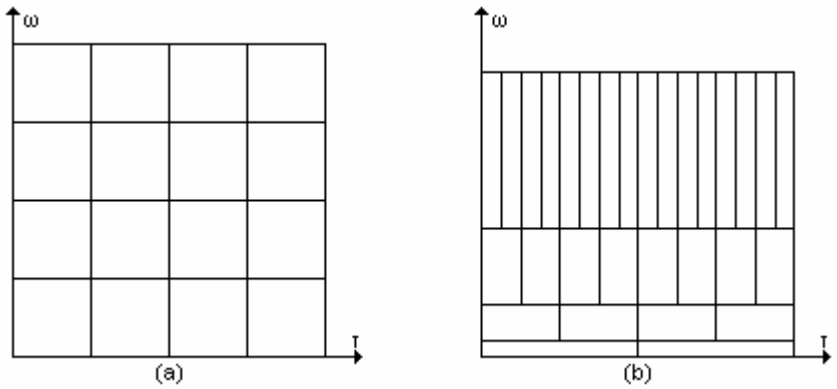
Transformata falkowa [86] (ang. *wavelet*) funkcji $f(t)$ w oparciu o bazę $\psi^*(.)$ w czasie ciągłym dana jest wzorem:

$$CWT(\tau,\alpha)=\frac{1}{\sqrt{|\alpha|}}\int f(t)\psi^*\left(\frac{t-\tau}{\alpha}\right)dt, \tag{2.11}$$

gdzie:
- τ oznacza współczynnik przesunięcia falki (wpływa na położenie na osi czasu),
- α oznacza współczynnik skali falki (wpływa na czas trwania)
- ψ oznacza funkcję falki bazowej.

Współczynniki τ i α interpretuje się jako miarę podobieństwa do danego fragmentu analizowanego sygnału. Czynnik $\frac{1}{\sqrt{|\alpha|}}$ normalizuje wyniki tak, aby na każdym poziomie każdy transformowany sygnał miał tą samą energię.

W odróżnieniu od wcześniej wspomnianych transformat Fouriera i Walsha-Hadamarda, które charakteryzują się równomiernym podziałem płaszczyzny spektrogramu, transformata falkowa dzieli przestrzeń wyznaczoną przez czas i częstotliwość w sposób logarytmiczny.



Rysunek 2.1 Podział płaszczyzny spektogramu w przekształceniach: (a) - DFT, (b) - WT.

Transformata odwrotna dla wzoru (2.11) wygląda następująco:

$$f(t)=\frac{1}{c_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} CWT(\tau,\alpha)\frac{1}{\sqrt{\alpha}}\psi\left(\frac{t-\tau}{\alpha}\right)\frac{d\tau d\alpha}{\alpha^2} \tag{2.12}$$

Można powiedzieć, że $f(t)$ jest splatana z funkcją ψ. Funkcja $\psi(t)$ musi spełniać następujący warunek związany z jej transformatą Fouriera $\Psi(\omega)$:

$$c_\psi=\int_{-\infty}^{\infty}\frac{|\Psi(\omega)|^2}{|\omega|}d\omega<\infty \tag{2.13}$$

Warunek ten jest spełniony, jeśli $\Psi(0)=0$ lub równoważnie $\int_{-\infty}^{\infty}\varphi(t)dt=0$.

W celu uzyskania dyskretnej transformaty falkowej, współczynniki τ i α są próbkowane tak, aby falka bazowa mogła być skalowana i przesuwana tylko w dyskretnych krokach. W rezultacie otrzymujemy dla jednowymiarowej dyskretnej transformaty falkowej:

$$DWT(j,k)=\frac{1}{\sqrt{\alpha_0^j}}\sum_n x(n)\Psi\left(\frac{n-k\tau_0\alpha_0^j}{\alpha_0^j}\right), \tag{2.14}$$

gdzie *i* oraz *j* są liczbami całkowitymi dodatnimi.

Schemat tworzenia 2-wymiarowej dyskretnej transformaty falkowej DWT wygląda następująco:

1. Przy pomocy 1-wymiarowej DWT przekształcane są wszystkie wiersze obrazu. W ten sposób powstaną dwa bloki współczynników o tej samej liczbie wierszy i połowie kolumn, co oryginalny obraz.
2. Wszystkie kolumny obu bloków współczynników przekształcane są przez 1-wymiarową DWT. W ten sposób powstają 4 bloki współczynników, każdy o połowie liczby wierszy i kolumn oryginalnego obrazu.
3. Proces przekształcania opisany w punktach 1. i 2. należy powtarzać do uzyskania 4 bloków o tylko jednym współczynniku.

Powyższy proces można zacząć poprzez przekształcenie kolumn oryginalnego obrazu i dalej postępując analogicznie.

Przykłady zastosowań falek w analizie tekstur można znaleźć w [81] oraz [90]. Dla obrazów w skali szarości właściwości tekstury określane są na podstawie wartości energii dyskretnej transformaty falkowej. Technika ta może być również rozszerzona na obrazy barwne.

2.1.5 Funkcje Gabora

Funkcja Gabora to liniowe przekształcenie obrazu x o postaci:

$$\begin{aligned} &\hat{\mathbf{s}} = \mathbf{H} \cdot \mathbf{x}, \\ &\mathbf{H} = \left[h_1(x, y), ..., h_n(x, y)\right] \end{aligned} \tag{2.15}$$

Funkcje bazowe dla $i = 1, \ldots, n$ dane są jako:

$$h_i(x, y) = g\left[q(x, y), w(x, y)\right]\cos\left(2\pi f_0 q(x, y)\right), \tag{2.16}$$

gdzie:

$$\begin{aligned} g(x, y) &= \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\}, \\ q(x, y) &= \cos(\Phi_0)x + \sin(\Phi_0)y, \\ w(x, y) &= -\sin(\Phi_0)x + \cos(\Phi_0)y. \end{aligned} \tag{2.17}$$

W powyższych wzorach Φ_0 oznacza kąt obrotu względem osi X, natomiast f_0 – częstotliwość podstawową, a środek oscylacji znajduje się w środku przekształcanego obszaru obrazu.

Można interpretować funkcje Gabora jako detektory krawędzi o różnych (dyskretnych) orientacjach i różnych częstotliwościach. Funkcje Gabora przyjmowane są uniwersalnie – niezależnie od zastosowania, w tym również w analizie tekstur [114].

2.2. Opis tekstury w MPEG-7 (deskryptory)

W tym rozdziale zostanie zaprezentowane podejście do opisu tekstury bazujące na funkcjach Gabora, a przewidziane w standardzie MPEG-7, zgodnie z dokumentami [18] oraz [116].

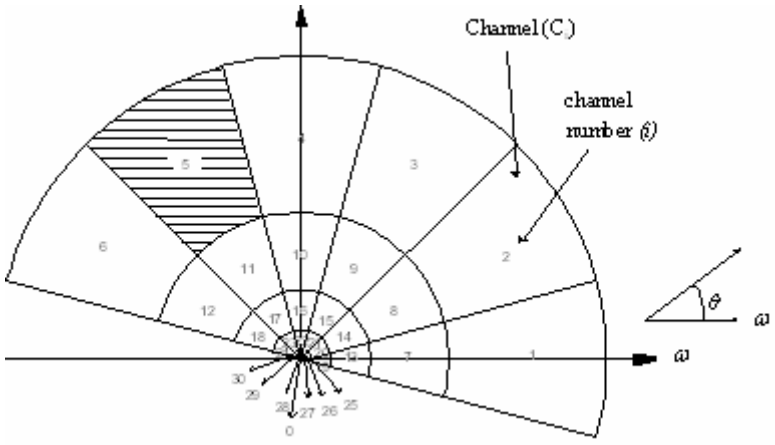
Celem MPEG-7, znanego również jako „Multimedia Content Description Interface", jest standaryzacja opisu obiektów multimedialnych. MPEG-7 proponuje trzy podejścia do opisu tekstury (tworzenia tzw. „deskryptorów"). Wynikają one z różnych sposobów interpretacji danych obrazu, stąd też znajdują zastosowanie w odmiennych przypadkach. Podstawą wyznaczania ich parametrów są funkcje Gabora.

1. Deskryptor tekstury jednorodnej, dalej oznaczany jako HTD (Homogeneous Texture Descriptor), opiera się na właściwościach statystycznych obrazu, reprezentowanych jako 62-elementowy wektor. Wektor ten wyznaczany jest na podstawie odpowiedzi filtru Gabora dla 30 kanałów częstotliwości obrazu.
2. Kolejny deskryptor standardu – deskryptor przeglądania tekstury TBD (Texture Browning Descriptor) również wyznacza się na podstawie odpowiedzi filtrów Gabora dla kanałów częstotliwości. Jego parametry zaś odpowiadają percepcyjnemu, zbliżonemu do ludzkiego, postrzeganiu tekstury. W testach wykorzystywany jest tzw. PBD – Perceptual Browning Descriptor, który zawiera informację o regularności, kierunkowości i porowatości wzorca.
3. Trzeci z proponowanych przez MPEG-7 deskryptorów, deskryptor histogramu krawędzi EHD (Edge Histogram Descriptor) to 80-punktowy histogram reprezentujący rozkład krawędzi w obrazie. Wykorzystywany jest przeważnie do analizy zmian sceny w sekwencji obrazów.

Podczas wcześniejszych prac autorka przebadała użyteczność tych trzech deskryptorów. W wyniku testów stwierdzono, że deskryptory HTD, PBD lub ich kombinacje są użytecznym narzędziem w analizie tekstur. Deskryptor EHD pominięto jako dający najmniej istotne wyniki. Badania nad wykorzystaniem właściwości deskryptorów MPEG-7 były prowadzone w ramach grantu KBN nr 7TO8A05016 ([91], [92]).

2.2.1 Wyznaczanie energii

Poszczególne elementy deskryptorów tekstury wyznaczane są na podstawie rozkładu częstotliwości w analizowanym obrazie. Przestrzeń częstotliwości dzielona jest na równe obszary o szerokości 30 stopni w kierunku kątowym i oktawy w kierunku biegunowym. Obszary te nazywane są kanałami własności. Podyktowane to jest sposobem postrzegania przez mózg, który dokonuje dekompozycji całego spektrum na pasma percepcji.



Rysunek 2.2 Schemat rozkładu częstotliwości dla 30 kanałów.

W przestrzeni znormalizowanej częstotliwości środkowe każdego kanału wyznaczane są według wzoru $\theta_r = 30^o \times r$, gdzie $0 \le r \le 5$ jest indeksem kątowym, natomiast szerokość kątowa wszystkich kanałów jest równa 30 stopniom. W kierunku biegunowym częstotliwości kątowe wyrażane są zależnością $\omega_s = \omega_0 \cdot 2^{-s}$, $0 \le s \le 4$, ω_0=3/4.

Do każdego z kanałów stosowana jest dwuwymiarowa funkcja Gabora:

$$G_{P_{s,r}}(\omega,\theta) = \mathbf{exp}\left[\frac{-(\omega-\omega_s)^2}{2\sigma_{\rho_s}^2}\right] \cdot \mathbf{exp}\left[\frac{-(\theta-\theta_r)^2}{2\sigma_{\theta_r}^2}\right] \qquad (2.18)$$

gdzie s- indeks kanału w kierunku biegunowym, r – indeks kanału w kierunku kątowym. Zmienne $\sigma_{\rho s}$ i $\sigma_{\theta r}$ oznaczają standardowe odchylenie w kierunku kątowym lub biegunowym, w miejscu zetknięcia się sąsiednich filtrów w połowie maksimum. W kierunku kątowym

odchylenie standardowe ma stałą wartość równą $\sigma_{\theta_r} = \frac{15°}{\sqrt{2\ln 2}}$, natomiast w kierunku biegunowym dana jest zależność:

$$\sigma_{\rho_s} = \frac{B_s}{2\sqrt{2\ln 2}} \tag{2.19}$$

gdzie $B_s = B_0 \cdot 2^{-s}$, $B_0 = 1/2$ jest szerokością pasma kanału w kierunku biegunowym.

Na podstawie rozkładu częstotliwości i funkcji Gabora wyznaczana jest energia e_i i-tego kanału własności:

$$\begin{aligned} e_i &= \log[1 + p_i], \\ p_i &= \sum_{\omega=0+}^{1} \sum_{\theta=0°+}^{360°} \left[G_{Ps,r}(\omega,\theta) \cdot P(\omega,\theta)\right]^2 \end{aligned} \tag{2.20}$$

oraz odchylenie standardowe energii

$$\begin{aligned} d_i &= \log_{10}[1 + q_i], \\ q_i &= \sqrt{\sum_{\varpi=0+}^{1} \sum_{\theta=0°+}^{360°} \left\{\left[G_{Ps,r}(\varpi,\theta) \cdot P(\varpi,\theta)\right]^2 - p_i\right\}^2} \end{aligned} \tag{2.21}$$

gdzie $P(\omega,\theta)$ oznacza transformatę Fouriera obrazu w biegunowej dziedzinie częstotliwości [75].

2.2.2 Deskryptor tekstury jednorodnej

Deskryptor ten opisuje właściwości statystyczne tekstury i wyznaczany jest według schematu omówionego w poprzednim punkcie. Deskryptor ten może składać się z 32 lub 62 elementów, w zależności od tego, czy brana jest pod uwagę podstawowa płaszczyzna własności czy płaszczyzna rozszerzona.

HTD=[*a*, *sd*, *e1*, *e2*, ..., *e30*, *ed1*, *ed2*, ..., *ed30*] (2.22)

Pierwsze dwa reprezentują średnią intensywności pikseli obrazu *a* z zakresu 0 – 255 oraz odchylenie standardowe *sd*. Następne stanowią 30-elementową tablicę zawierającą wartości energii e_i wyznaczone dla każdego z kanałów własności. Ostatnie 30 elementów to tablica wartości odchylenia standardowego energii ed_i dla kanałów własności. Wartości te wyznaczane są, gdy brana jest pod uwagę rozszerzona płaszczyzna własności.

Deskryptor HTD wykorzystywany jest przy wyszukiwaniu precyzyjnym. Podobieństwo między obrazami mierzone jest jako odległość między wektorami własności:

$$d(TD_{query}, TD_{database}) = dist(TD_{query}, TD_{datatbase}) = \sum_k \left| \frac{TD_{query}(k) - TD_{database}(k)}{\alpha(k)} \right| \qquad (2.23)$$

gdzie $\alpha(k)$ jest odchyleniem standardowym $TD_{database}(k)$ dla danej bazy (może być również ustalane).

Proponowane są cztery sposoby porównywania [59]:

- niezależne od intensywności – z wektora własności HTD usuwany jest pierwszy element;
- niezależne od rotacji – wektor własności obrazu obróconego jest przesuniętą kątowo wersją dla obrazu referencyjnego; wówczas odległość wyznaczana jest jako najmniejsza z otrzymanych w wyniku przesunięcia;
- niezależna od skali – dla N możliwych wersji przeskalowanego czynnika odległość wyznaczana jest jako najmniejsza z uzyskanych;
- niezależne od obrotu i skalowania – odległość wyznaczana jest jako najmniejsza ze wszystkich możliwych kombinacji skalowania i obrotu.

Testy przeprowadzone członków grupy MPEG-7 wykazały dużą skuteczność tego deskryptora ([1], [97]). Jako miarę efektywności stosuje się ARR (Average Retrieval Rate – wskaźnik średniej liczby prawidłowo odnalezionych obrazów).

Prezentowane w dokumencie M5490 ([79]) wyniki utrzymują się na poziomie ARR powyżej 70 dla baz zawierających obrazy różnych tekstur zdefiniowanych w MPEG-7 CE (Core

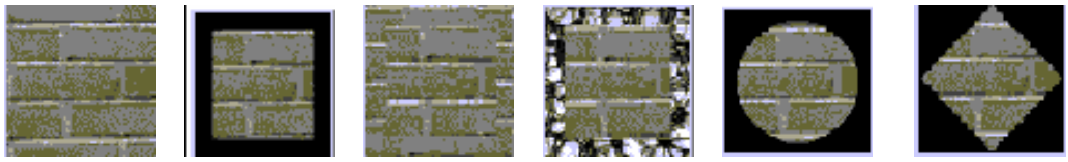
Experiments). W przypadku, gdy pod uwagę brana jest rozszerzona płaszczyzna własności (deskryptor HTD zawiera oprócz energii wektor wartości odchylenia standardowego) następuje poprawa wskaźnika ARR.

Dokumenty [76] oraz [115] omawiają wyniki eksperymentów dla deskryptora tekstury jednorodnej przebrowadzone na bazie 832 obrazów tekstury. Jako zapytanie pobrano arbitralnie 52 obrazy, które poddano obróbce, wycinając odpowiedni kształt. W efekcie uzyskano dodatkowe trzy zbiory – tekstury o kształcie prostokąta, rombu (diamentu) i koła. Poniższa tabela przedstawia wyniki trzech eksperymentów.

Tabela 2.1 Przykładowe wyniki testów dla wzorców HTD o różnym kształcie (wskaźnik ARR)

Zbiór zapytania	Wypełnianie	Czarne tło	Inna tekstura
Oryginały	91.03	86.62	86.62
Kształt prostokąta	83.97	67.05	45.38
Kształt koła	81.67	59.36	48.97
Kształt rombu	78.46	59.36	43.59

Kolumny odpowiadają kolejnym eksperymentom. W pierwszym tło obrazów zostało wypełnione danymi ze środka obszaru tekstury. W drugim pozostało one czarne. W trzecim eksperymencie tło zostało wypełnione inna teksturą. Przedstawiają to rysunki poniżej:



(ITD.) (b) (c) (d) (e) (f)

Rysunek 2.3 Przykłady wzorców o różnych kształtach: (a) - oryginał, (b) - wzorzec prostokątny na czarnym tle, (c) - wzorzec prostokątny na tle wypełnionym zawartością tektury wzorca, (d) - wzorzec prostokątny na tle wypełnionym inną strukturą, (e) - wzorzec okrągły, (f) - wzorzec romboidalny.

Najlepsze wyniki uzyskano dla obrazów oryginalnych – wartość wskaźnika ARR utrzymywała się powyżej 85. Dla wszystkich typów obrazów pogorszenie nastąpiło przy zmianie tła, przy czym najgorsze wyniki dało wypełnienie tła inną teksturą – dla tekstur o wyznaczonych kształtach wskaźnik ARR spadł poniżej 50. Pogorszenie wyników spowodowane jest faktem wykonywania operacji filtrowania na granicach obszarów.

2.2.3 Deskryptor przeglądania tekstury

Deskryptor ten odpowiada percepcyjnemu, zbliżonego do ludzkiego, charakteryzowaniu tekstury. Operuje takimi terminami jak regularność, porowatość, kierunkowość. Wyznaczanie, podobnie jak w przypadku HTD, odbywa się poprzez filtrację za pomocą filtrów Gabora, przy czym mamy 4 zamiast 5 indeksów *s*. Wykonywana jest dekompozycja obrazu – każdy z otrzymanych w jej wyniku obrazów zawiera informację w określonej skali i kierunku. Na tej podstawie wyznaczany jest wektor TBC (*Texture Browsing Component*), określający regularność lub jej brak we wzorcu.

Kolejne elementy wektora

$$\mathbf{TBC} = [v1\ v2\ v3\ v4\ v5] \tag{2.24}$$

oznaczają:

- regularność *v1* – reprezentuje stopień regularności obrazu (im większa wartość, tym bardziej regularny wzorzec);
- kierunkowość (*v2*, *v4*) – dwa kierunki dominujące tekstury;
- skala (*v3*, *v5*) – dwie dominujące wartości, zależne od stopnia regularności wzorca.

Kierunkowość oparta jest na histogramach wyznaczanych ze zbioru przefiltrowanych obrazów w różnych skalach, zdefiniowanych wzorem:

$$H(s,k) = \frac{N(s,k)}{\sum_0^5 N(s,k)} \tag{2.25}$$

dla $s = 0,...,3$ i $k = 0,...,5$. N(s, k) oznacza liczbę pikseli w filtrowanym obrazie w skali *s* i kierunku *k*, których natężenie jest większe niż dany próg t_s. Jako kierunek (kierunki) dominujący wybierany jest ten, którego wierzchołek jest w $H(s, k)$, również dla sąsiednich wartości skali. Jako elementy wektora TBC zapisywane są dwa kierunki *DO1* i *DO2* największego kontrastu (w niektórych przypadkach może być określony tylko jeden kierunek).

W celu wyznaczenia skali określane są dwie projekcje ([58], [60]) dla każdego przefiltrowanego obrazu $W_{mn}(x, y)$ wzdłuż kierunków dominujących *DO1* i *DO2*:

$$P_H^{mn}(l)\iint W_{mn}(x,y)\delta(x\cos\theta_{DO1} + y\sin\theta_{DO1} - l)dxdy \tag{2.26}$$

$$P_V^{mn}(l)\iint W_{mn}(x,y)\delta(x\cos\theta_{DO2} + y\sin\theta_{DO2} - l)dxdy \tag{2.27}$$

Następnie dla każdej ze zbioru projekcji (dla obu kierunków dominujących) wyznaczana jest znormalizowana funkcja autokorelacji *NAC* (Normalized Autocorrelation Function) [60]:

$$NAC(k) = \frac{\sum_{m=k}^{N-1} P(m-k)P(m)}{\sqrt{\sum_{m=k}^{N-1} P^2(m-k)\sum_{m=k}^{N-1} P^2(m)}} \tag{2.28}$$

gdzie $P(l)$ oznacza wybraną projekcję dla danego kierunku dominującego. Niech M i N – maksima i minima funkcji $NAC(k)$, natomiast *p_magn*(i) oraz *v_magn*(i) (i=1,2, …, M) - wartości lokalnych minimów i maksimów.Wówczas

$$K = \frac{1}{M}\sum_{i=1}^{M} p_magn(i) - \frac{1}{N}\sum_{j=1}^{N} v_magn(j) \tag{2.29}$$

wyznacza kontrast.

Na podstawie sekwencji wierzchołków *p_pos*(*i*) wyznaczane są odległości między wierzchołkami *dis* oraz odchylenie standardowe *std*. Niech γ = *std*/*dis*. Jeśli wartość ta jest mniejsza od założonego progu, daną projekcję uważa się za regularną. Po usunięciu pozostałych wyznacza się projekcje o największym kontraście. Przyjmując $m^*(H)$ jako indeks skali projekcji o największym kontraście i $m^*(V)$ – skala projekcji o najmniejszym kontraście, elementy wektora TBC wyznaczane są jako:

$$\begin{gathered} v3 = m^*(H) \\ \text{oraz} \\ v5 = m^*(V) \end{gathered} \tag{2.30}$$

Przy wyznaczaniu regularności stosowane są specjalne założenia dla tych projekcji, które przeszły test regularności omawiany powyżej. Dla tekstur jednorodnych kandydaci

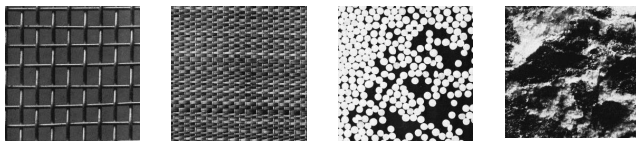
(projekcje) są sąsiadami w danej przestrzeni. Kandydatów dzieli się na trzy grupy i przydziela się im na tej podstawie *kredyty V*. Pierwsza grupa określa tych, dla których można znaleźć co najmniej jednego innego kandydata odpowiadającego w sąsiedniej skali lub kierunku. Kredyt wynosi wówczas V_1=1.0. Następną grupę stanowią projekcje, dla których istnieje odpowiedni kandydat dla takiej samej skali lub kierunku, ale nie ma ich dla skali (kierunków) sąsiednich. Grupie tej przyznaje się kredyt V_2=0.5. Ostatni zbiór to kandydaci, którzy są jedyni dla swojej skali i kierunku – przyznany kredyt wynosi V_3=0.2. Jako warunek określający regularność przyjmuje się sumę kredytów dla zbioru kandydatów w obu dominujących kierunkach:

$$M = \sum_{i=1}^{3} N_i * V_i \tag{2.31}$$

gdzie N_i określa liczbę kandydujących projekcji. Wówczas

$$\begin{aligned} &v1 = 0 \quad \text{jeśli} \quad M < 5, \\ &v1 = 1 \quad \text{jeśli} \quad 5 \le M < 10, \\ &v1 = 2 \quad \text{jeśli} \quad 10 \le M < 20, \\ &v1 = 3 \quad \text{jeśli} \quad M \ge 20 \end{aligned} \tag{2.32}$$

Im wyższa wartość parametru *v1*, tym bardziej regularna struktura. Poniższy rysunek przedstawia przykładowe tekstury dla określenia stopnia regularności.



Rysunek 2.4 Przykłady tekstur o różnym stopniu regularności wzorca (od lewej - zmniejszający się stopień regularności)

TBD wykorzystywany jest do zgrubnego wyszukiwania obrazów o podobnych właściwościach, może więc służyć do tworzenia zbioru kandydatów przy wyszukiwaniu precyzyjnym. Z tego względu używa się go w kombinacji z deskryptorem tekstury jednorodnej HTD.

Wyznaczanie podobieństwa dla obrazów z baz MPEG-7 XM omawia dokument M5092 [59]. Podane wyniki dotyczą deskryptora, który oprócz wektora TBC zawiera wektor SRC, o elementach określających pierwszy i drugi moment przefiltrowanych obrazów. Osiągnięto

wyniki powyżej 70% prawidłowo odnalezionych obrazów, przy czym porównania obrazu zapytania samego z sobą były pomijane. Dla tekstur z albumu Brodatz'a osiągnięto wynik 99.96%.

Dokument M5449 omawia wyniki przeszukiwania dla podobnie zbudowanego deskryptora (zawierającego wektory TBC i SRC). Bazę stanowiły wzorce z albumu Brodatz'a. Każdy ze 166 wzorców został podzielony na 4 podobrazy, obrócony. Obrazy podzielone zostały na dwie grupy – w określonymi dwoma kierunkami dominującymi i z jednym kierunkiem dominującym. Podział taki służył zbadaniu, jak detekcja kierunku wpływa na ludzka percepcję. W przypadku pierwszej grupy lepsze wyniki (98 % znalezionych) osiągnięto przy porównywaniu dla pierwszego kierunku dominującego (88% dla drugiego). Dla drugiej grupy, z pojedynczym kierunkiem dominującym osiągnięto wynik 100% (zobacz [76]).

2.2.4 Deskryptory tekstury w analizie obrazów

Na potrzeby pracy wykonano serię testów dotyczących analizy tekstury w obrazach angiograficznych. Zastosowano połączenie omawianych deskryptorów ([61], [78]).Szczegółowe wyniki zostaną omówione w dalszych rozdziałach.

Ze względu na precyzję wyszukiwania przyjęto pełne wersje deskryptorów, to jest 62 elementowy deskryptor tekstury jednorodnej (dodatkowe wyznaczanie odchylenia energii) oraz 5-elementowy deskryptor przeglądania tekstury (dwa kierunki dominujące zamiast jednego). W tej sytuacji funkcja podobieństwa rozumiana jako odległość między elementami poszczególnych zbiorów cech wygląda następująco (indeks „*q*" – zapytanie, indeks „*db*" – wzorzec):

- dla deskryptora tekstury jednorodnej

$$\begin{aligned} d_{HTD}\left(\mathbf{TD}_q, \mathbf{TD}_{db}\right) &= 0.28 \cdot abs(\mathbf{TD}_{db}[0] - \mathbf{TD}_q[0]) + \\ &* 0.22 \cdot abs(\mathbf{TD}_{db}[1] - \mathbf{TD}_q[1]) + \\ &+ \sum_{s=0}^{5} \sum_{r=0}^{4} \{\mathbf{m}[r] \cdot abs(\mathbf{TD}_{db}[i] - \mathbf{TD}_q[j]) + \mathbf{d}[r] \cdot abs(\mathbf{TD}_{db}[j] - \mathbf{TD}_q[j])\} \end{aligned} \tag{2.33}$$

gdzie $i=s+r+2$; $j=s+r+32$;

- dla deskryptora przeglądania tekstury

$$\begin{aligned} d_{TBD}\left(\mathbf{TD}_{db}, \mathbf{TD}_{q}\right) &= 0.6 \cdot abs(\mathbf{TD}_{db}[0] - \mathbf{TD}_{q}[0]) + \\ &+ 0.2 \cdot \{0.5 \cdot [abs(\mathbf{TD}_{db}[1] - \mathbf{TD}_{q}[1]) + abs(\mathbf{TD}_{db}[2] - \mathbf{TD}_{q}[2])\} + \\ &+ 0.2 \cdot \{0.5 \cdot [abs(\mathbf{TD}_{db}[3] - \mathbf{TD}_{q}[3]) + abs(\mathbf{TD}_{db}[4] - \mathbf{TD}_{q}[4])\} \end{aligned} \tag{2.34}$$

Wartości elementów tablic współczynników *m* i *d* we wzorze (2.32) określone są w kodzie MPEG-7 Experimentation Model, podobnie jak pozostałe wagi we wzorach (2.32) i (2.33). Symbol $\mathbf{TD}_{db}$ określa wektor cech (tzw. „deskryptor") referencyjny, $\mathbf{TD}_{q}$ – wektor cech obszaru analizowanego, natomiast *abs* oznacza wartość bezwzględna wyrażenia podanego jako parametr. Indeksy określają kolejność elementów w wektorze cech (62 dla deskryptora tekstury jednorodnej i 5 dla deskryptora przeglądania tekstury). Wartości $dist_{HTD}$ i $dist_{TBD}$ są następnie sumowane i porównywane do wartości progowej α. Wartość ta należy do przedziału (0.5, 0.8), wyznaczonego w trakcie eksperymentów.

3. Analityczne przekształcenia obrazu

Rozdział ten omawia trzy główne sposoby pozyskiwania przekształceń przestrzeni reprezentacji obrazu, które, kryteria odróżnieniu od przedstawionych w rozdziale 2 przekształceń heurystycznych, optymalizują kryteria oparte na próbkach (danych) uczących charakteryzujących dziedzinę zastosowania.

3.1. Przekształcenie PCA

Po fazie ekstrakcji otrzymujemy pewne skupisko punktów. Jednym ze sposobów jego reprezentacji jest rzutowanie ortogonalne na podprzestrzeń liniową. Analiza składowych głównych PCA (ang. P*rincipal Component Analysis*) polega na przetworzeniu dużej ilości informacji zawartej we wzajemnie skorelowanych danych wejściowych na zbiór nowych, ortogonalnych względem siebie cech [99]. W niektórych opracowaniach nazywana jest ona również (dyskretną) transformatą Karhunena-Loevego. Dla wektora cech $\mathbf{x} \in R^N$ znajdowana jest baza podprzestrzeni, dla której wartość oczekiwana normy różnicy między wektorem **x** a jego rzutem na tę podprzestrzeń osiąga minimum. Innymi słowy, przekształca dany wektor **x** na wektor $y \in R^K$ za pośrednictwem macierzy $W \in R^{K \times N}$, $K<N$. Nowa przestrzeń o zredukowanym wymiarze zachowuje informacje dotyczące procesu, zatem metoda PCA może być uznana za formę kompresji danych ([86], [103], [108]). Wektory macierzy **W** nazywane są inaczej składowymi głównymi, a ich liczba jest równa wymiarowi szukanej podprzestrzeni liniowej.

Warunkiem na to, by zbiór wektorów skupiska **X** tworzył podprzestrzeń liniową jest odjęcie od każdego wektora wartości średniej tego zbioru, czyli spełnienie warunku:

$$E[X]=0 \tag{3.1}$$

Dla danego skupiska X o wymiarze N szukana jest podprzestrzeń S o wymiarze $0 < K < N$ taka, że oczekiwana wartość długości różnicy wektora (skupiska) X i jego rzutu jest minimalna:

$$E\left\|X - P_S(X)\right\|^2 = \mathbf{min}\,, \tag{3.2}$$

gdzie $P_S(\mathbf{x})$ oznacza projekcję wektora **x** na podprzestrzeń S ([87], [88]). Można zauważyć, że poszukiwanie podprzestrzeni minimalizującej oczekiwany błąd projekcji (3.2) jest równoważne poszukiwaniom podprzestrzeni maksymalizującej wariancję projekcji [26]:

$$\begin{aligned} &E\left\|X - P_S(X)\right\|^2 = \\ &E\left\|X - E[X]\right\|^2 - E\left\|P_S(X) - E[X]\right\|^2 \end{aligned}, \tag{3.3}$$

Niech $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ oznacza wektor losowy o zerowej wartości średniej, a $\mathbf{R}_{xx}=E[\mathbf{x}\mathbf{x}^T]$ – wartość średnią macierzy autokorelacji po wszystkich wektorach **x**. Niech $\mathbf{W}_i=[\mathbf{W}_{i1}, \mathbf{W}_{i2}, \ldots, \mathbf{W}_{iN}]$, $i=1, 2, \ldots, N$ oznacza ortogonalne wektory własne odpowiadające wartościom własnym λ_i macierzy $\mathbf{R}_{xx}$. Można zdefiniować przekształcenie PCA

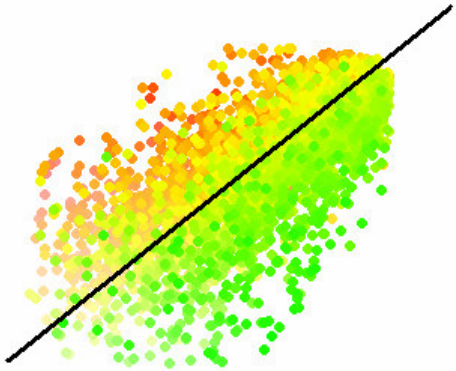
$$\mathbf{y}=\mathbf{W}\mathbf{x} \tag{3.4}$$

poprzez uszeregowanie wartości własnych macierzy kowariancji malejąco $\lambda_1<\lambda_2<\ldots<\lambda_N \leq 0$ i wybór z tej grupy jedynie K największych wartości macierzy **W**.

Minimalizację wartości oczekiwanej błędu rekonstrukcji otrzymuje się dla bazy przestrzeni, której wektory **w** spełniają warunek:

$$\begin{gathered} \mathbf{cov}(X)\cdot w_i = \lambda_i w_i\,, \\ i = 1, \ldots, N \\ \lambda_1<\lambda_2<\ldots<\lambda_N \end{gathered} \tag{3.5}$$

gdzie cov(*X*) jest macierzą kowariancji skupiska X, a λ_i – *i*-tą wartością własną tej macierzy. Można wyznaczyć również wektory własne macierzy kowariancji bez wyznaczania samej macierzy kowariancji. Metoda ta korzysta bezpośrednio z wektorów wejściowych (skupiska) ([16], zobacz także [83], [89]).



Rysunek 3.1 PCA: Składowa główna próbek w dwuwymiarowej przestrzeni cech. Linia wskazuje kierunek pierwszej składowej głównej, druga składowa główna jest prostopadła (w ogólności - ortogonalna).

3.2. ICA

Podstawowym zastosowaniem metody ICA (ang. *independent component analysis*) jest estymacja wielu sygnałów źródłowych jedynie na podstawie obserwacji ich mieszanin. Nie jest wymagana żadna informacja ani o kierunku źródła, ani o warunkach pomiaru. Zakłada się, że obserwowany *n*-wymiarowy sygnał wektorowy $\mathbf{x}(t)$ jest wynikiem "punktowego" (nieznanego) wymieszania *m* niezależnych statystycznie (nieznanych) sygnałów źródeł $\mathbf{s}(t)$ ([34], [35]):

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) = \sum_{i=1}^{m} s_i(t)\mathbf{a}_i^T + \mathbf{n}(t), \tag{3.6}$$

gdzie $\mathbf{a}_i^T$ oznacza i-ty wiersz macierzy $\mathbf{A}$.

Celem ICA jest jednoczesna estymacja nieznanych źródeł i $m \times n$ – wymiarowej macierzy separującej $\mathbf{W}(t)$ takiej, e *m*-wymiarowy wektor:

$$\mathbf{y}(t) = \mathbf{W}(t)\,\mathbf{x}(t) \tag{3.7}$$

staje się (z dokładnością do skali i permutacji sygnałów) aproksymacją nieznanych źródeł. Z punktu widzenia zapisu matematycznego wykorzystuje ona również liniowe przekształcenie przestrzeni wektorowej, podobnie jak transformata Fouriera, Hadamarda, transformata kosinusowa itp. Jednak w ICA następuje każdorazowa estymacja przekształcenia w oparciu o same dane pomiarowe - uzyskane w ten sposób przekształcenie jest dopasowane do aktualnego problemu (zastosowania).

Znanych jest szereg algorytmów do ICA. Zasadnicze kryteria ich zróżnicowania to:

- algorytm pracujący w trybie "wsadowym" lub adaptacyjny algorytm typu "on-line",
- algorytm wymagający obróbki wstępnej lub algorytm jedno-przebiegowy.

Najbardziej znane algorytmy do ICA to:

- C. Jutten, 1990 (historycznie 1-szy algorytm oparty jedynie o dekorelację wyższego rzędu) [44],
- P. Common, 1994 (definicja pojęcia “ICA”) [19],
- J. Cardoso, 1989-95 (algorytm “Jade ICA” - "wsadowy") ([9], [10]),
- Cichocki et al., 1994 (adaptacyjny algorytm typu „on-line”) ([14], [15], [17]),
- T. Bell & T. Sejnowski, 1995 (algorytm “Infomax ICA”) ([5], [6]),
- S. Amari et al., 1995-96 (adaptacyjny algorytm “Natural gradient ICA”) ([2], [3]),
- Hyvarinen, E.Oja, J. Karhunen, 1998-2000 (algorytm „Fast ICA” - "wsadowy") ([71], [72]).

W dalszej części pracy omawiany będą jedynie przypadki liniowe ICA, choć w literaturze można spotkać przykłady nieliniowych zastosowań ICA ([19], [44]). Pominięte zostaną również przypadki sygnałów zaszumionych. Podstawy (warunki) modelu ICA są następujące [36]:

1. Wszystkie komponenty niezależne s_i muszą mieć rozkład nie-Gaussowski z dokładnością do jednego elementu.
2. Liczba obserwowanych mieszanin liniowych m musi być co najmniej równa liczbie komponentów niezależnych n, czyli $m \geq n$.

3. Jeżeli macierz mieszająca A ma wymiary $m \times n$, to musi być rzędu n.

Dodatkowo zakłada się, że wektory **x** i **s** są wyśrodkowane (można to uzyskać poprzez odjęcie wartości średniej wektora).

Definicja ICA, w odróżnieniu od PCA, nie zakłada sortowania składowych niezależnych, ale można takie sortowanie zastosować (np. wykorzystując normy kolumn macierzy mieszającej).

Niech $p(y_1,y_2)$ oznacza gęstość rozkładu łącznego zmiennych y_1 i y_2, a $p_1(y_1)$ i $p_1(y_1)$ – gęstość rozkładu brzegowego dla y_1 i dla y_2 odpowiednio. Zmienne losowe y_1 i y_2 są niezależne wtedy i tylko wtedy, gdy:

$$p(y_1,y_2) = p_1(y_1)\, p_2(y_2). \tag{3.8}$$

Dla dwóch funkcji niezależnych zmiennych losowych h_1 i h_2 zachodzi:

$$E\{h_1(y_1)h_2(y_2)\} = E\{h_1(y_1)\}E\{h_2(y_2)\}. \tag{3.9}$$

Dwie zmienne y_1 i y_2 są zdekorelowane, jeśli ich kowariancja jest zerowa:

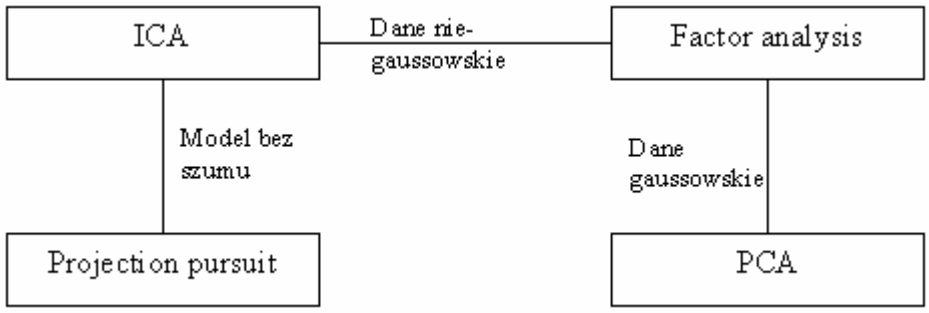
$$E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0. \tag{3.10}$$

Zmienne niezależne są jednocześnie zdekorelowane, ale nie na odwrót. Np. niech (y_1,y_2) są dyskretnymi zmiennymi o prawdopodobieństwie 1/4 dla wartości (0,1), (0,-1), (1,0), (-1,0). Wtedy y_1 i y_2 są zdekorelowane. Ale jednocześnie zachodzi $E\{y_1^2 y_2^2\} = 0 \neq \frac{1}{4} = E\{y_1^2\}E\{y_2^2\}$, czyli zmienne nie są niezależne.

Dwie zmienne Gaussowskie nie mogą być odseparowane w modelu ICA. Mieszaniny dwóch zmiennych Gaussa wymieszane ortogonalną macierzą zachowują rozkłady Gaussa. Gęstość rozkładu łącznego dwóch mieszanin jest wtedy w pełni symetryczna - nie zawiera ona żadnej informacji o kierunku wektorów w macierzy mieszającej. Taka macierz nie może zostać znaleziona w analizie ICA i jest to podstawą warunku numer 1.

Dla sygnałów niezaszumionych ICA może być uznana za przypadek „projection pursuit" – oznacza wtedy wyszukiwanie pewnych interesujących projekcji, które mogą estymować składowe niezależne ([25], [32], [98]). Równie duże podobieństwo można znaleźć w przypadku ślepego rozplatania ([15], [48]) oraz analizy czynnikowej (rozdział 3.3).

Można też zauważyć powiązania z analizą składowych głównych, niemniej jednak nie należy uznawać ICA za „rozszerzenie" PCA. PCA ma duże znaczenie w przypadku redukcji wymiarów, natomiast w przypadku ICA możemy mieć do czynienia zarówno z redukcją, zwiększeniem lub brakiem zmian, jeśli chodzi o wymiary wektorów. Powiązania z innymi metodami przedstawia poniższy rysunek [36]:



Rysunek 3.2 Linie przedstawiają powiązania między wybranymi metodami. Opisy obok linii mówią o wymaganiach, które muszą być spełnione, by relacja była prawdziwa.

Klasycznym przykładem zastosowań ICA jest ślepe rozplatanie wzorców (*blind source separation*) [46].

3.3. Analiza czynnikowa (*factor analysis*)

Analiza czynnikowa (ang. *factor analysis*) to metoda służąca do odnajdywania struktur w zbiorze zmiennych losowych ([30], [51]). Celem analizy czynnikowej jest zredukowanie dużej liczby zmiennych do mniejszego zbioru. Istnieją dwa podejścia:

- eksploracyjna analiza czynnikowa EFA (*Exploratory Factor Analysis*): czynniki są początkowo nieznane i zostają wyodrębnione dzięki analizie wartości zmiennych
- konfirmacyjna analiza czynnikowa CFA (Confirmatory Factor Analysis) – zakładamy istnienie pewnego określonego zbioru czynników i dzięki analizie wartości zmiennych

losowych badamy zasadność naszego przypuszczenia i estymujemy parametry modelu.

Zakładany jest następujący model danych:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n}, \tag{3.11}$$

gdzie **x** jest wektorem obserwowanych zmiennych, **s** – wektorem ukrytych zmiennych (czynników), których nie można zaobserwować. **A** jest macierzą o wymiarach $m \times n$, a wektor **n** reprezentuje szum. Wektory **x** i **n** mają ten sam wymiar n, natomiast zazwyczaj zakłada się, że **s** ma wymiar niższy od **x**. Porównując wzory (3.11) i (3.6) łatwo zauważyć powiązanie z ICA, natomiast biorąc pod uwagę redukcję wymiarów można porównywać analizę czynnikową z PCA.

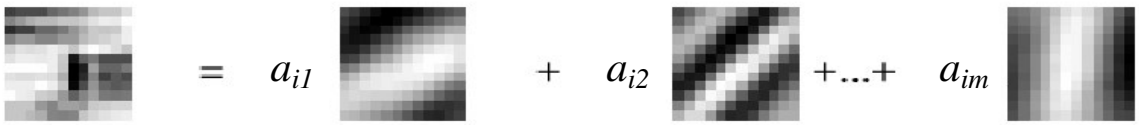
Istnieją dwie główne metody analizy czynnikowej. Pierwszą z nich jest analiza czynników głównych PFA (*Principal Factor Analysis*), która stanowi modyfikację PCA. Idea polega na zastosowaniu PCA do danych **x** w taki sposób, by uwzględnić wpływ zaszumienia. Innymi słowy zakłada się, że macierz kowariancji szumu jest znana. Czynniki znajdowane są poprzez wykonanie PCA z wykorzystaniem zmodyfikowanej macierzy kowariancji powstałej w wyniku odjęcia od macierzy kowariancji **x** macierzy kowariancji **n**. W ten sposób wektor **s** uzyskuje się jako wektor składowych głównych **x** po usunięciu szumu. Druga popularna metoda może być sprowadzona do wyznaczania składowych głównych na zmodyfikowanej macierzy kowariancji.

4. Detekcja cech dzięki dekompozycji ICA

W tym rozdziale przedstawiona zostanie autorska metoda detekcji cech tekstury wykorzystująca dekompozycję przestrzeni reprezentacji znalezioną dzięki przekształceniu ICA

4.1. Opis metody

Podstawą stosowania algorytmów ICA w analizie obrazów jest założenie, że dany obraz można przedstawić w postaci sumy ważonej obrazów składowych ([31], [43], [113]). Obrazy składowe są statystycznie niezależne.



Rysunek 4.1 Ilustracja składania tekstury obrazu z niezależnych składowych: x_i - obserwowany i-ty blok obrazu; a_i współczynniki wymieszania źródeł (tu: wartości cech obserwowanego bloku obrazu); s_1, s_2 ,..., s_m, - źródła w ICA (ich 2-wymiarowa ilustracja) niezależne składowe (wektory bazowe)

Prostokątne bloki obrazu (każdy o rozmiarze $k \times l = N$) traktowane są jako mieszaniny m niezależnych składowych (źródeł w modelu ICA). Bloki te są skanowane w ustalonym z góry porządku do postaci sygnałów $x_i(t)$ ($i = 1,..., n; t = 1,..., N$). Zakładamy, że te sygnały (zebrane w wektor $\mathbf{x}(t)$) spełniają model punktowego mieszania przyjęty w ICA:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) = \sum_{i=1}^{m} s_i(t)\mathbf{a}_i^T \ . \tag{4.1}$$

Czyli $\mathbf{x}(t)$ jest wektorową reprezentacją obserwowanych bloków obrazów (jest to wektor n sygnałów, każdy z nich o długości $N = k \times l$, jako wynik skanowania bloków obrazu). $\{\mathbf{a}_i\}$ to n wektorów cech (każdy o długości m) tworzących wiersze macierzy (mieszającej) $\mathbf{A}$ (każdy wiersz stanowi wartości cech, poszukiwane dla opisu tekstury w obserwowanym bloku obrazu). $\{s_i\}$ to m N-elementowych sygnałów źródeł jak na rysunku powyżej.

Dysponując obserwacjami danymi w postaci kolejnych bloków obrazu $\mathbf{x}_i(t)$ stosujemy algorytm ICA pozwalający na jednoczesną estymację nieznanych niezależnych składowych $\mathbf{s}$ i pewnej macierzy $\mathbf{W}$. Po "zamrożeniu" macierzy $\mathbf{W}$ określany jest ostateczny wektor wyjściowy $\mathbf{y}$ taki, że:

$$\begin{aligned} \hat{\mathbf{s}} = \mathbf{y} &= \mathbf{W} \cdot \mathbf{x} \\ \mathbf{W} &\approx \mathbf{A}^{-1} \end{aligned} \tag{4.2}$$

Macierz $\mathbf{W}$ znaleziona w procesie „nauczenia" systemu składowych niezależnych nie jest dalej brana pod uwagę. Dopiero w fazie aktywnej pracy systemu detekcji cech tekstury celem jest ustalenie dla każdego analizowanego bloku obrazu jego wektora cech $\mathbf{a}_i$. Ten problem sprowadza się do standardowego problemu identyfikacji kanału przesyłowego o wejściu $\mathbf{s}$ i wyjściu $\mathbf{x}$.

Podsumowując własności metody ICA w procesie opisu tekstury obrazu:

- ICA znajduje najbardziej „nie-Gaussowską" dekompozycję obrazu;
- funkcje bazowe (źródła $\mathbf{s}$) są dopasowane do konkretnej klasy obrazów;
- wektory $\mathbf{s}_i$ posiadają naturalną interpretację po przekształceniu ich w bloki obrazu - są detektorami krawędzi w obrazie - interesujący aspekt dla „biologicznych badań percepcji człowieka".

Aby uzyskać redukcję rozmiaru przestrzeni cech w stosunku do rozmiaru oryginalnego bloku obrazu, należy zastosować obróbkę wstępną typową dla kompresji obrazu, np. w postaci PCA. Dla przykładu blok o rozmiarze 12 x 12 prowadzi do wektora obserwacji o długości

144; w procesie PCA można wybrać np. 50 największych składowych głównych, dokonać kompresji wektorów **x** do rozmiaru 50 x 1 i znaleźć w procesie ICA macierz **W** o rozmiarze 50 x 50. Oczywiście w celu 2-wymiarowej ilustracji wektorów bazowych (wierszy macierzy **W**) należy najpierw przywrócić każdy 50-elementowy wektor do postaci 144-elementowego wektora i dopiero wtedy zamieniać je na bloki 12 x 12-elementowe

4.2. Proces uczenia wektorów bazowych

4.2.1 Przetwarzanie wstępne

Przypomnijmy, że w przypadku algorytmów ICA mówimy o zmiennych niezależnych (zob. rozdział 3.2). Dwie zmienne Gaussowskie nie mogą być odseparowane w modelu ICA. Mieszaniny dwóch zmiennych Gaussa wymieszane ortogonalną macierzą zachowują rozkłady Gaussa. Gęstość rozkładu łącznego dwóch mieszanin jest wtedy w pełni symetryczna - nie zawiera ona żadnej informacji o kierunku wektorów w macierzy mieszającej. Taka macierz nie może zostać znaleziona w analizie ICA.

W celu uproszczenia algorytmów ICA stosuje się usuwanie wartości średniej. Niech **m** będzie wektorem wartości średnich wektora obserwacji **x**(*t*). Po estymacji sygnałów wyjściowych w ICA ich wymagane wartości średnie mogą zostać odtworzone jako:

$$\mathbf{A}^{-1}\mathbf{m}, \tag{4.3}$$

gdzie **A**-1 jest odwrotności macierzy mieszającej (określonej w procesie ICA).

Stosowane jest również „wybielanie” (ang. „whitening”, ortogonalizacja). Jest to liniowe przekształcenie wektora tak, aby poszczególne składowe obserwowanego wektora były nieskorelowane i posiadały jednostkowe wariancje:

$$E\left[\tilde{x}\tilde{x}^T\right] = \mathbf{I}. \tag{4.4}$$

Wybielenie próbek można przeprowadzić np. dzięki dekompozycji na wektory własne macierzy kowariancji próbek (EVD).

$$E\left[xx^T\right] = EDE^T$$
$$D = diag\left(d_1,...,d_n\right) \tag{4.5}$$

Nowy wektor pomiarowy otrzymujemy wyznaczając:

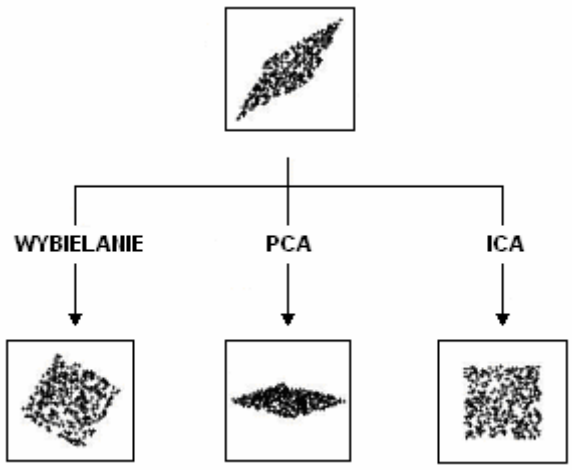
$$\tilde{x} = ED^{-1/2}E^T x\,. \tag{4.6}$$

Proces ten przeprowadza macierz mieszającą w macierz ortogonalną

$$\tilde{x} = ED^{-1/2}E^T As = \tilde{A}s$$
$$E\left[\tilde{x}\tilde{x}^T\right] = \tilde{A}E\left[ss^T\right]\tilde{A}^T = \tilde{A}\tilde{A}^T = \mathbf{I} \tag{4.7}$$

Po przeprowadzeniu "wybielenia" obserwacji w procesie ICA zamiast estymacji n^2 elementów (swobodnej) macierzy wystarczy teraz estymacja $n(n\text{-}1)/2$ elementów macierzy ortogonalnej. W graficznej interpretacji dla 2-wymiariowego rozkładu (po „wybieleniu" uzyskujemy rozkład próbek w obszarze kwadratu) wystarczy teraz znalezienie jednego kąta obrotu, aby boki kwadratu były równoległe do osi układu współrzędnych. Możliwa też staje się ewentualna redukcja rozmiaru macierzy - wtedy, gdy pewne wartości własne λ_j są już zbyt małe.

Przetwarzanie wstępne jest istotnym etapem przekształcenia ICA – pozwala na zmniejszenie liczby wykonywanych operacji oraz przygotowuje dane do dalszego przetwarzania.



Rysunek 4.2 Graficzna interpretacja procesu wybielania - po wybieleniu uzyskujemy kwadratowy rozkład próbek, wystarczy więc znalezienie jednego kąta obrotu.

4.2.2 Algorytm ICA

Do testów wykorzystano algorytm "FastICA", opracowany przez Hyvarinena, Karhunena i Oję ([33], [39]). Pracuje on w trybie wsadowym i dlatego te możliwe jest zastosowanie kroków wstępnej obróbki danych, które przyspieszają i ułatwiają proces zbiegania się procesu uczenia wag macierzy **W**. Ten algorytm oparty jest na iteracyjnej regule modyfikacji macierzy wag i takim kryterium kosztu, które wymusza takie wartości macierzy **W**, przy których następuje maksymalizacja niegaussowskiego charakteru wektora wyjściowego **y = Wx**. Ten niegaussowski charakter jest wyrażony np. przez dywergencję Kulbacka-Leiblera ("negentropię"):

$$KL(\mathbf{W}) = \int p(\mathbf{y}) \log \left[\frac{p(\mathbf{y})}{\prod_{k=1}^{K} p(y_k)} \right] d\mathbf{y}, \qquad (4.8)$$

$$KL(\mathbf{W}) = -H(\mathbf{Y};\mathbf{W}) + \sum_{k=1}^{K} H(Y_k;\mathbf{W})$$

Pierwszy składnik wyraża entropię rozkładu łącznego, a drugi - sumę entropii 1-wymiarowych rozkładów brzegowych. Wartość $KL(\boldsymbol{W})$ jest minimalna wtedy, gdy poszczególne zmienne są wzajemnie niezależne. Ponieważ bezpośrednie obliczenie $KL(\boldsymbol{W})$ jest zbyt złożone, w praktyce stosuje się aproksymacje rozkładów przy pomocy momentów wyższego rzędu i nieliniowych funkcji. Sam proces iteracji, którego celem jest minimalizacja kryterium kosztu $KL(\boldsymbol{W})$, jest zbliżony do metody Newtona.

Przed wykonaniem algorytmu wymagana jest obróbka wstępna sygnału wektorowego, a więc usunięcie wartości średniej wektora obserwacji oraz wybielenie obserwacji.

Pierwszym etapem algorytmu „FastICA" jest inicjalizacja macierzy wag **W** dowolnymi niezerowymi wartościami. Następnie dla wyjść p = 1, ..., *n* wykonywane są kroki (2-4):

1. Obliczany jest zmodyfikowany wektor:

$$\mathbf{w}^{+} = E\left\{\mathbf{x} \cdot g\left(\mathbf{w}^{T}\mathbf{x}\right)\right\} - E\left\{g'\left(\mathbf{w}^{T}\mathbf{x}\right)\right\} \cdot \mathbf{w}, \qquad (4.9)$$

gdzie g jest funkcją nieliniową , g' jej pierwszą pochodną względem czasu.

2. Uzyskany w punkcie 1. wektor jest normalizowany do wektora o długości jednostkowej :

$$\mathbf{w} = \frac{\mathbf{w}^+}{\left\|\mathbf{w}^+\right\|}. \tag{4.10}$$

3. Aby zapobiec zdążaniu poszczególnych wektorów do tego samego maksimum należy dokonywać dekorelacji kolejnego wektora względem poprzednich wektorów wag. Możliwy jest tu np. schemat deflacji oparty na dekorelacji Grama-Schmidta (od aktualnego wektora odejmij sumę rzutów poprzednich "ustalonych" wektorów, a następnie znormalizuj tak otrzymany nowy wektor):

$$\begin{aligned} \mathbf{w}_{p+1} &= \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}_{p+1}^T \mathbf{w}_j \,; \\ \mathbf{w}_{p+1} &= \frac{\mathbf{w}_{p+1}}{\sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}}. \end{aligned} \tag{4.11}$$

Jeśli nie chcemy nadawać określonej kolejności wektorom wag, to możliwa jest też symetryczna dekorelacja „macierzowa”:

$$\mathbf{W} = \frac{\mathbf{W}}{\sqrt{\mathbf{W}\mathbf{W}^T}}. \tag{4.12}$$

4. Jeśli zbieżność **W** nie wystąpiła to algorytm jest powtarzany od kroku 1.

Przekształcenia uzyskiwane w wyniku analizy ICA są estymowane w oparciu o same dane uczące – przykłady obrazów danego rodzaju ([38], [40], [41], [55]). ICA powinna dobrze spełniać rolę klasyfikatora, gdyż orientuje się na różnice pomiędzy obrazami różnych klas ([45], [56]).

4.3. Detekcja cech a analiza obrazu i klasyfikacja.

Opisywany algorytm „FastICA” wykorzystano do testów przydatności deskryptora opartego o współczynniki przekształcenia ICA w klasyfikacji tekstur dla obrazów cyfrowych. W testach

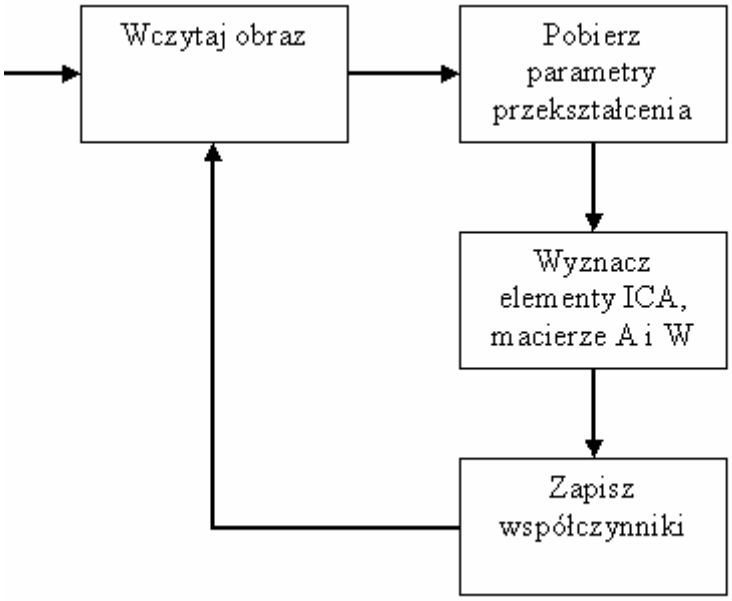
wykorzystano obrazy angiograficzne oraz, celem porównania, obrazy z albumu tekstur Brodatz'a.

Algorytm „FastICA" pozwala na określenie warunków wyznaczania współczynników. Testowano następujące parametry algorytmu:

1. sposób estymacji składowych niezależnych w procesie dekorelacji:
   - symetryczny (jeden po drugim),
   - równoległy;
2. funkcja nieliniowa do wyznaczania wektora we wzorze 4.9:
   - $g_1(u) = u^3$,
   - $g_2(u) = \mathbf{tanh}(u)$,
   - $g_3(u) = u \cdot \mathbf{exp}\left(\frac{-u^2}{2}\right)$,
   - $g_4(u) = u^2$.

Badane były możliwe kombinacje tych parametrów. W niektórych przypadkach dla testów funkcji nieliniowej nie udało się osiągnąć zbieżności algorytmu. Taka sytuacja miała miejsce dla zastosowanych funkcji $g_2(u)$, $g_3(u)$ oraz $g_4(u)$, dlatego też w testach klasyfikatorów wykorzystano tylko deskryptory wyznaczone z wykorzystaniem funkcji $g_1(u)$.

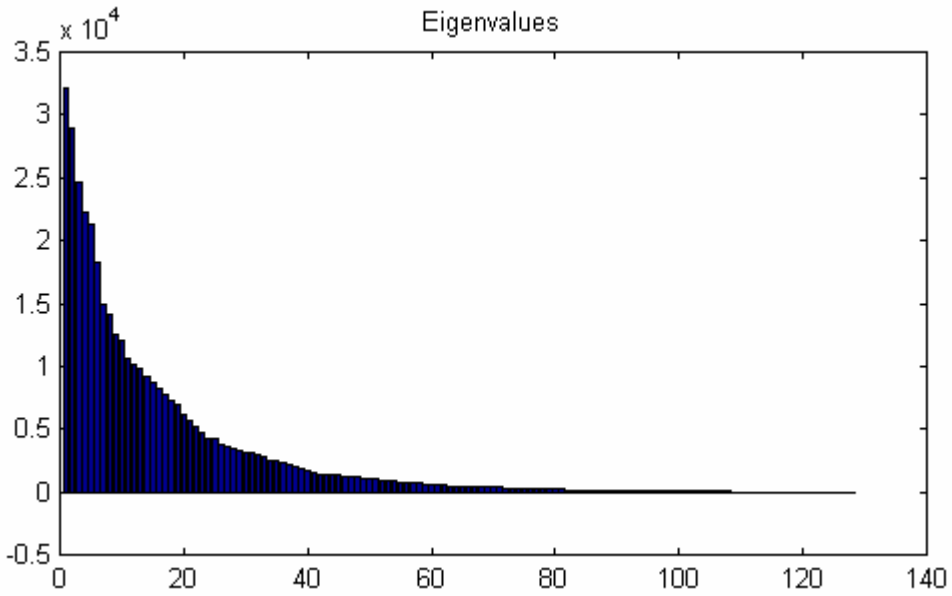
Schemat działania aplikacji testowej przedstawiony jest poniżej:



Rysunek 4.3 Schemat działania aplikacji testowej.

Do testów wybrano obrazy w skali szarości, o wymiarach 64 × 64 pikseli. Taki dobór parametrów obrazu umożliwia szybsze działanie algorytmu bez utraty istotnych informacji.

W wyniku działania przekształcenia ICA otrzymuje się zestaw informacji, składający się z macierzy składowych niezależnych, macierzy wag **W** oraz macierzy **A**, będącej odwrotnością macierzy **W** (zob. wzór 4.2), a zawierającej wartości cech obrazu, które będą następnie wykorzystywane w procesie klasyfikacji. Rozmiary tych macierzy zależą od liczby próbek podlegających przetwarzaniu – jak wspomniano wcześniej, na etapie przetwarzania wstępnego możliwe jest ograniczenie rozmiarów macierzy poprzez wybranie odpowiednio wysokich wartości własnych. W najprostszym przypadku, kiedy nie stosujemy żadnych ograniczeń, liczba próbek jest równa rozmiarowi macierzy reprezentującej obraz. Jeśli natomiast stosuje się ograniczenie liczby próbek, należy pamiętać o konieczności zachowania tej samej liczby próbek dla każdego obrazu testowego, aby proces klasyfikacji na podstawie wektora cech, zbudowanego z współczynników przekształcenia ICA, przebiegał prawidłowo. Wymaga to więc wcześniejszych testów, w trakcie których można określić dopuszczalną minimalną ilość próbek.

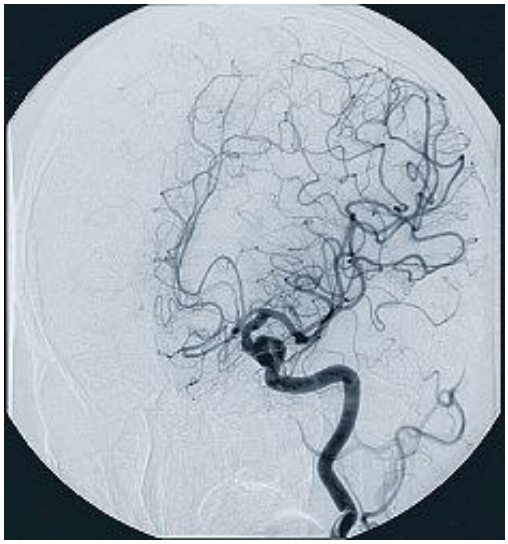


Rysunek 4.4 Na podstawie takiego wykresu można odrzucić te wartości własne, które nie przekraczają pewnego progu. Dla obrazu tekstury z albumu Brodatz'a o rozmiarach 128 × 128, dla którego wykonano ten wykres, można ograniczyć rozmiar do (na przykład) 80 próbek bez ryzyka utraty istotnych informacji.

Po wyznaczeniu macierzy **A**, z jej elementów tworzony jest wektor cech dla danego obrazu. W procesie klasyfikacji wektory cech różnych obrazów są porównywane z wykorzystaniem różnych typów klasyfikatorów.

5. Obrazy angiograficzne

Angiografia to metoda diagnostyki medycznej, która pozawala na wizualizację naczyń krwionośnych (w tym ubytków) poprzez podanie środka kontrastowego. Badania tego typu wykorzystywane są w na przykład w kardiologii w celu określenia ryzyka zawału lub zmian chorobowych (np. retinopatii cukrzycowej [70]).

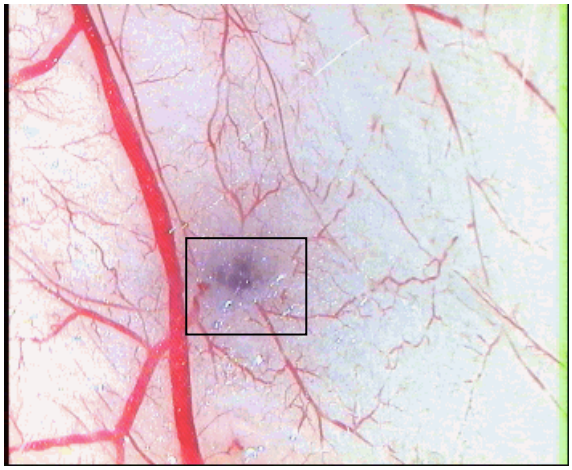


Rysunek 5.1 Obraz naczyń krwionośnych w mózgu (angiogram).

Typową procedurą w przypadku obrazów angiograficznych jest substrakcja. Zapamiętywany jest obraz przed podaniem środka kontrastowego, który następnie odejmowany jest od kolejnych obrazów wykonywanych w trakcie badania [24]. Najczęściej spotykane aplikacje

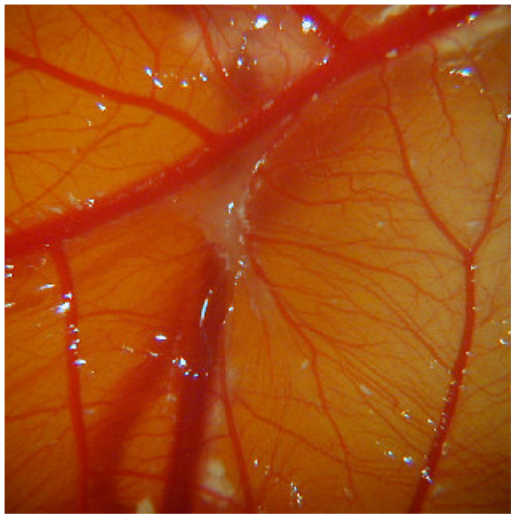
dotyczą wyznaczania konturów w obrazach, przy czym prezentowane są różne podejścia ([69], [111]).

W testach przeprowadzonych dla potrzeb niniejszej pracy wykorzystano obrazy przedstawiające proces angiogenezy. Angiogeneza, czyli tworzenie się nowych naczyń, stanowi ważny składnik wielu procesów fizjologicznych, takich jak organogeneza (wykształcanie się narządów) czy gojenie się ran. Nadmierna, patologiczna angiogeneza stanowi podłoże wielu chorób, np. przewlekłych chorób zapalnych tkanki łącznej, retinopatii cukrzycowej[1] i nowotworów [93]. Angiogeneza nowotworowa, czyli tworzenie się nowych naczyń w obrębie guza, jest procesem niezbędnym dla jego wzrostu i progresji przerzutów. Część obrazów wykorzystanych w omawianych testach pochodzi z badań nad inhibitorami (czynnikami hamującymi wzrost) procesu angiogenezy [4].



Rysunek 5.2 Proces angiogenezy w tkance skórnej myszy. Zaznaczony obszar to miejsce wstrzyknięcia preparatu.

[1] Retinopatia cukrzycowa – powikłanie cukrzycy, najczęstsze schorzenie oka, polegające na uszkodzeniu małych naczyń krwionośnych siatkówki, prowadzące do uszkodzenia wzroku oraz ślepoty.



Rysunek 5.3 Proces angiogenezy w zarodku kurzym.

Analizując tego typu obrazy angiograficzne można brać pod uwagę zarówno długość naczyń, jak też ich średnicę. W przypadku angiogenezy istotna też jest ocena ilości naczyń w danym obszarze. Wykorzystując metody analizy tekstur w obrazach cyfrowych można przeprowadzić analizę porównawczą obrazów badanych wobec kontrolnych (stwierdzić, czy nastąpił przyrost naczyń, czy też ich ubytek).

6. Klasyfikatory cech

6.1. Problem klasyfikacji

W dziedzinie rozpoznawania wzorców proces klasyfikacji polega na przypisaniu wektora cech **c**, reprezentującego badany wzorzec, do pewnej dyskretnej klasy Ω_i ze zbioru klas Ω. Wymagany wektor cech jest wynikiem przekształcenia całego wzorca (w tym przypadku obrazu) w wyniku transformacji przestrzeni reprezentacji lub segmentacji obrazu. Dobry proces detekcji cech powinien doprowadzić do sytuacji, gdy cechy wzorca tej samej klasy zajmują „zwarty" obszar w przestrzeni cech, a obszary cech wzorców różnych klas mogą być od siebie odseparowane. Klasyfikacja wzorca wymaga podjęcia decyzji. Decyzja podejmowana jest zarówno na podstawie obserwacji danego wektora cech, w oparciu o wiedzę nabytą na podstawie zbioru uczącego.

Kryteria oceny jakości (optymalnej) klasyfikatora można definiować na różne sposoby. Pod uwagę można brać takie aspekty jak:

- wartość prawdopodobieństwa błędnej klasyfikacji, koszt ryzyka,
- złożoność obliczeniowa,
- zdolność do uczenia reguł,
- zdolność do uogólnienia dla nieznanych próbek.

Statystyczna ocena klasyfikatora obejmuje trzy zagadnienia. Pierwszy stanowi opis statystycznych własności analizowanych wzorców. Po wielu decyzjach możliwe jest oszacowanie prawdopodobieństwa błędów przekształceń. W drugim etapie obliczane jest ryzyko (koszty) podjęcia decyzji. Koszty powstają, gdy wzorzec zostanie zaklasyfikowany błędnie do klasy Ω_i zamiast Ω_j. Trzecie zagadnienie wiąże się z tym, czy poziom błędu ma charakter reprezentacyjny dla całości obszaru cech, czy też ogranicza się do obszaru pokrytego przez próbki uczące. W przypadku, gdy wybierana jest dokładnie jedna spośród K możliwych klas, to najczęściej stosuje się zero-jedynkową funkcję kosztu:

$$\begin{gathered} r_{kk} = 0, \\ r_{ik} = 1 \text{ dla } i \neq k \text{ oraz } i,k = 1, \ldots, K \end{gathered} \tag{6.1}$$

W takim przypadku optymalny jest klasyfikator Bayesa jako osiągający najmniejsze prawdopodobieństwo błędu [112], jeśli taki błąd może zostać oszacowany. W praktyce jest to prawda, gdy zbiór próbek uczących ma charakter reprezentatywny dla całej dziedziny cech. W przeciwnym razie możemy wiarygodnie oszacować jedynie tzw. błąd empiryczny. Błąd empiryczny jest minimalizowany najlepiej przez klasyfikator SVM.

Podstawowe założenia konstrukcyjne klasyfikatora są następujące:

- jeśli wektory cech należą do tej samej klasy, odległość między nimi jest mała,
- jeśli wektory cech należą do różnych klas, odległość między nimi jest duża.

Odległości między wektorami cech można wyznaczyć korzystając z funkcji gęstości prawdopodobieństwa, odległości Euklidesowej lub metryki modułowej.

6.1.1 Podstawowe typy klasyfikatorów numerycznych

Przykłady klasyfikatorów numerycznych najczęściej występujące w literaturze [103]:

1. Klasyfikator statystyczny

   Rozkład prawdopodobieństwa klas nad przestrzenią cech ma znaną postać, a zadaniem procesu uczenia jest wyznaczenie parametrów tego rozkładu dla każdej z rozpatrywanych klas. Zakłada się, że próbki uczące mają charakter reprezentatywny dla całości rozkładu. Klasyfikator statystyczny może dysponować parametryczną lub nie-parametryczną funkcją gęstości. Do tej grupy zalicza się klasyfikator Bayesa i jego uproszczone formy –

klasyfikator największej wiarygodności i klasyfikator geometryczny według najmniejszej odległości.

2. Klasyfikator SVM (maszyna wektorów nośnych)
   Zakłada się, że próbki uczące mają skończony charakter, a więc nie są reprezentatywne dla całości rozkładu. Klasyfikacja *K* klas ma charakter wielokrotnej klasyfikacji binarnej.

3. Klasyfikator według funkcji potencjału
   Dla danego zbioru próbek określana jest rodzina funkcji potencjału zgodna z jego charakterem (np. funkcja liniowa, wielomian). Pojedyncza funkcja jest związana z klasą poprzez odpowiedni zestaw parametrów, a klasyfikacja odbywa się zgodnie z maksymalną wartością funkcji potencjału dla danego wektora cech. Klasyfikator ten został jednak wykorzystany w testach, ponieważ zdecydowano się na bardziej optymalną SVM.

4. Klasyfikator neuronowy
   Sieci neuronowe wykorzystywane są do aproksymacji funkcji rozdzielających lub reguł decyzyjnych. W celu klasyfikacji wektora cech najczęściej używa się sieci typu „wielowarstwowy perceptron".

5. Klasyfikator k-sąsiadów
   W procesie uczenia określa się podział przestrzeni cech, a każda próbka ucząca jest pojedynczym reprezentantem danej klasy. W procesie klasyfikacji 1 lub więcej reprezentantów sąsiadujących z aktualnie klasyfikowaną próbką rywalizuje w głosowaniu za swoją klasą.

6.2. Klasyfikator statystyczny Bayesa

Niech

- **x** – zmierzony (wykryty) wektor cech, który powinien zostać przypisany do pewnej klasy
- C_k – klasa cech tekstury, k=1, …, K.

Prawdopodobieństwo warunkowe *a priori* przynależności wektora **x** do klasy C_k określa się wzorem:

$$P(x_i \mid C_k) = \frac{P(C_k \cap x_i)}{P(C_k)}. \tag{6.2}$$

Przekształcając powyższy wzór i korzystając z własności prawdopodobieństwa warunkowego otrzymujemy następującą równość:

$$P(C_k \cap x_i) = P(C_k) \cdot P(x_i \mid C_k). \tag{6.3}$$

Te dwa prawdopodobieństwa mogą i powinny być określone na etapie uczenia (projektowania) klasyfikatora. Mamy więc:

- $P(C_k)$ – prawdopodobieństwo *a priori* występowania klasy *k*-tej, $k=1, \ldots, K$
- $P(x_i|C_k)$ – prawdopodobieństwo warunkowe *a priori* wektora cech x_i dla klasy C_k.

6.2.1 Uczenie klasyfikatora

W przypadku uczenia rozkładów prawdopodobieństwa mówimy o dwóch modelach funkcji gęstości prawdopodobieństwa $p(X|C_k)$:

- nieparametryczny, dla którego stosuje się dyskretne prawdopodobieństwa (np. histogramy),
- parametryczny, polegający na założeniu rodziny funkcji gęstości i szacowaniu parametrów na podstawie próbek uczących, zwykle przy założeniu statystycznej niezależności elementów wektora cech.

W przypadku parametryzacji gęstości rozkładu prawdopodobieństwa dla *d*-wymiarowego wektora cech o rozkładzie normalnym stosuje się *d*-wymiarową funkcję gęstości:

$$p(x \mid C_k) = \frac{1}{\sqrt{\det(2\pi\Sigma_k)}} \cdot e^{-\frac{(x-\mu_k)^T \cdot \Sigma_k^{-1} \cdot (x-\mu_k)}{2}}. \tag{6.4}$$

W procesie uczenia klasyfikatora wartości parametrów μ_k i Σ_k, dla każdej klasy C_k, gdzie $x_i^{(k)}$ jest próbką klasy C_k, mogą być oszacowane według metody największego prawdopodobieństwa:

$$\mu_k \cong \frac{1}{N_k}\sum_{i=1}^{N_k} x_i^{(k)} , \tag{6.5}$$

$$\Sigma_k \cong \frac{1}{N_k}\sum_{i=1}^{N_k}\left(x_i^{(k)} - \mu_k\right)^T\left(x_i^{(k)} - \mu_k\right). \tag{6.6}$$

6.2.2 Reguła decyzyjna

Poszukujemy wartości P(C_k|x_i) - prawdopodobieństwo warunkowe a posteriori klasy C_k dla wektora x_i takie, że

$$\underset{K}{\mathbf{arg\,max}}\, P(C_k \mid x_i) . \tag{6.7}$$

Dla oznaczeń jak powyżej (x_i – wektor cechy, C_k – klasa) twierdzenie Bayesa jest postaci:

$$P(C_k \mid x_i) = \frac{P(x_i \mid C_k)\cdot P(C_k)}{P(x_i)} . \tag{6.8}$$

Oznaczenie: C_k|x_i – zdarzenie „obiekt należy do klasy C_k, jeśli zmierzona cecha ma wartość x_i". W liczniku wyrażenia (6.5) występują oba prawdopodobieństwa *a priori*.

Prawdopodobieństwo błędnej klasyfikacji jest najmniejsze, jeśli wybrana zostanie klasa C_k, dla której prawdopodobieństwo warunkowe *a posteriori* (6.5) jest największe.

Ponieważ mianownik wyrażenia (6.5) ma taką samą postać przy wyznaczaniu tego prawdopodobieństwa dla każdej klasy, można pominąć jego wartości w funkcji decyzyjnej.

Reguła decyzyjna klasyfikatora Bayesa to:

$$\varsigma(X) = \underset{k}{\operatorname{argmax}}\; p(C_k \mid X) = \underset{k}{\operatorname{argmax}}\; p(C_k)p(X \mid C_k) \tag{6.9}$$

Dla jednorodnego rozkładu klas (równe prawdopodobieństwa dla wszystkich par klas) reguła decyzyjna Bayesa sprowadza się do reguły decyzyjnej największej wiarygodności:

$$\varsigma(X) = \underset{k}{\operatorname{argmax}}\, p(C_k)p(X \mid C_k) = \underset{k}{\operatorname{argmax}}\, p(X \mid C_k) \tag{6.10}$$

Jeśli prawdopodobieństwa *a priori* ustalone są w sposób reprezentacyjny dla całego wektora klas, to prawdopodobieństwo błędu klasyfikatora Bayesa p_B stanowi dolną granicę błędu dowolnego klasyfikatora [67].

W części eksperymentalnej zastosowany zostanie klasyfikator Bayesa, dla którego w procesie uczenia zastosowano funkcję gęstości dla rozkładu normalnego – wzór (6.4).

6.2.3 Klasyfikator według minimalnej odległości

Podstawą klasyfikacji minimalnoodległościowej jest założenie, że zbiór wektorów cech tworzy obszary skupień w swojej przestrzeni [85]. Dla tego typu klasyfikatora oblicza się dla każdej klasy C_k odległość wektora cech od środka danej klasy. Dokładność i czas rozpoznawania zależą od wybranej metryki przestrzeni cech. Jedną z popularniejszych i stosowanych w tej pracy jest metryka Euklidesowa, dla której miara odległości wyraża się wzorem

$$d = \sqrt{\sum_{j=0}^{N-1} \left(x_j - x_j^{'}\right)^2} \tag{6.11}$$

gdzie:

- N – liczba wyznaczonych cech
- x, x' – wartości charakteryzujące cechy dla wektora cech i środka klasy.

Reguła podejmowania decyzji w tym przypadku oznacza, że nieznany obiekt zostanie zakwalifikowany do tego obszaru, do którego środka jest najbliżej w przestrzeni cech. Jest to specyficzny przypadek klasyfikatora Bayesa. Na przykład dla dwóch klas reguła decyzyjna porównuje kwadratowe miary odległości Euklidesowej aktualnego wektora cech od środków rozkładów klas (reguła decyzyjna według minimalnej odległości):

$$\log p(C_i) - \frac{1}{2}\left[(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right] > \log p(C_j) - \frac{1}{2}\left[(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right]. \quad (6.12)$$

Po uproszczeniu tej reguły otrzymujemy regułę decyzyjną według minimalnej odległości:

$$\begin{aligned} (x-\mu_i)^T(x-\mu_i) &< (x-\mu_j)^T(x-\mu_j), \\ |x-\mu_i|^2 &< |x-\mu_j|^2. \end{aligned} \quad (6.13)$$

Reguła (6.12) powstaje z reguły klasyfikatora Bayesa przy założeniu, że $p(C_i) = p(C_j)$ dla $i,j=1,\ldots,K$ oraz normalizacji przestrzeni cech $\Sigma_i = 1$, $i=1,\ldots,K$.

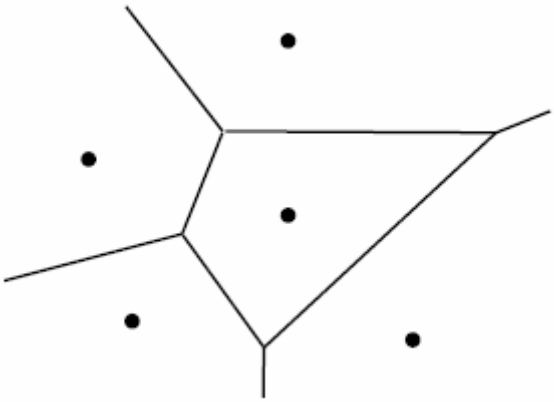
6.3. Klasyfikator „*k* najbliższych sąsiadów”

W przypadku klasyfikatora k-NN („*k* najbliższych sąsiadów”) wymagany jest zbiór próbek wektorów cech x_i należących do klas C_i. Reguła decyzyjna polega na przypisaniu nowego wektora cech do tej klasy, do której należy większość spośród *k* jego najbliższych sąsiadów [80].

W przypadku, gdy rozważamy tylko jednego sąsiada (a więc klasyfikator według najbliższego sąsiada – klasyfikator NN), reguła decyzyjna ma postać:

$$\underset{C_k}{\arg\min}\left\{|x-x_i|, i=1,2,...,K\right\}, \quad (6.14)$$

gdzie *x* oznacza nowy wektor. Zbiór punktów, dla których najbliższym sąsiadem jest x_i, nazywany jest komórką Voronoi’a dla x_i. W ten sposób dzielona jest cała przestrzeń cech.



Rysunek 6.1 Przykładowy zbiór Voronoi'a**.**

Jeśli przyjmiemy jako p_B prawdopodobieństwo błędu klasyfikatora Bayesa (zob. rozdział 6.2), to prawdopodobieństwo błędnej klasyfikacji dla klasyfikatora NN dla *K* klas można oszacować jako:

$$p_B \le p_{NN} \le p_B\left(2 - p_B \frac{K}{K-1}\right) \tag{6.15}$$

W przypadku klasyfikatora k-NN reguła decyzyjna polega na wybraniu klasy najlepszej wśród *k* najbliższych sąsiadów. Prawdopodobieństwo błędnej klasyfikacji dla klasyfikatora k-NN zbiega do prawdopodobieństwa błędu klasyfikatora Bayesa wraz ze wzrostem próbek uczących i wartości *k*.

6.4. Klasyfikator SVM

Algorytm klasyfikacji SVM (*Support Vector Machine*) przedstawiony został m. in. w pracach [109] i [110]. W algorytmie tym przyjmuje się skończony charakter próbek uczących. Problem sprowadza się do wielokrotnej decyzji pomiędzy dwiema klasami (lub grupami klas) (oznaczanymi zwykle jako „+1”, „-1”). Podczas uczenia maszyny poszukuje się optymalnej hiper-płaszczyzny (dla przypadku liniowego) lub hiper-powierzchni (dla „nieliniowej” maszyny) [67]. Algorytm SVM z powodzeniem stosuje się w dziedzinie rozpoznawania twarzy czy znaków [21].

Dla binarnej maszyny SVM można wybrać jedną ze strategii klasyfikacji *k* klas:

- *jedna przeciw wszystkim pozostałym*: klasyfikator decyduje o przynależności próbki do danej klasy lub do pozostałych (*k*-1) klas,
- *jedna przeciw jednej*: dla każdego binarnego klasyfikatora wybrana klasa uzyskuje punkt, a decyzja *k*-klasowa polega na wyborze klasy o największej liczbie punktów,
- *jedna przeciw pozostałym*: po uporządkowaniu klas w szereg, realizuje się (*k*-1) klasyfikatorów binarnych dla odróżnienia pojedynczej klasy od pozostałych, następnych w szeregu.

Do testów wybrano podejście „*jedna przeciw jednej*".

6.4.1 Funkcje kosztu empirycznego

Załóżmy, że zadaniem maszyny jest przypisanie pewnych danych $\mathbf{x}_i$ (reprezentowanych jako wektor) do pewnej klasy określanej za pomocą y_i. Dla danych uczących istnieje też pewna nieznana funkcja prawdopodobieństwa P(**x**, *y*). Próbki rozdzielane są pomiędzy klasami za pomocą funkcji rozdzielającej *f*(**x**, **α**), gdzie **α** jest parametryzatorem. Średni koszt (ryzyko) takiej klasyfikacji określamy jako:

$$R(\alpha)=\int \frac{1}{2}\left|y-f(\mathbf{x},\alpha)\right| dP(\mathbf{x},y). \tag{6.16}$$

Funkcja P(**x**, *y*) jest nieznana, więc powyższy wzór należy zastąpić oszacowaniem ryzyka empirycznego. Dla *N* próbek uczących oszacowanie takie przyjmuje postać:

$$R_e(\alpha)=\frac{1}{2N}\sum_{i=1}^{N}\left|y_i-f(x_i,\alpha)\right|. \tag{6.17}$$

Wielkość $\frac{1}{2}\left|y_i-f(x_i,\alpha)\right|$ nazywana jest stratą [8].

Dla każdej funkcji rozdzielającej *f* i dla każdego $N > h$ zachodzi z prawdopodobieństwem 1-η zależność [109]:

$$R(\alpha) \le R_e(\alpha) + \sqrt{\frac{h\left[\log(2N/h)+1\right] - \log(\eta/4)}{N}} \,. \tag{6.18}$$

Wartość η można określić następująco: załóżmy, ze zadaniem maszyny jest przypisanie obrazu do jednej z dwóch klas. Wówczas strata może przyjąć tylko dwie wartości – 0 albo 1. W takim przypadku $0 \le \eta \le 1$. Z kolei parametr h nazywany jest rozmiarem Vapnika-Chervonenkisa lub krótko – rozmiarem VC.

Rozmiar VC, we wzorze (6.27) oznaczony jako h, jest miarą zbioru funkcji rozdzielających. Dla problemu dwóch klas h wyznacza maksymalną liczbę wzorców, które mogą być rozdzielone we wszystkie możliwe sposoby – liczba takich podziałów wynosi 2^h. Wystarczy przy tym, aby dla każdego sposobu podziału istniał przynajmniej jeden zbiór wzorców o mocy h, tzn. istnieje wtedy przynajmniej jedna prawidłowa funkcja rozdzielająca.

Specyficznym zbiorem funkcji rozdzielających jest zbiór zorientowanych hiperpłaszczyzn

$$\begin{aligned} f(\mathbf{x}, \tilde{\boldsymbol{\alpha}}) &= \mathbf{x}^T \boldsymbol{\alpha} + \alpha_0 \\ \tilde{\boldsymbol{\alpha}} &= \begin{pmatrix} \boldsymbol{\alpha} \\ \alpha_0 \end{pmatrix} \end{aligned} \,. \tag{6.19}$$

Za pomocą tych płaszczyzn określamy, czy wynik leży po „dodatniej" czy po „ujemnej" stronie płaszczyzny, lub też na płaszczyźnie rozdzielającej. Rozmiar VC zbioru zorientowanych hiper-płaszczyzn w przestrzeni $\Re^n$ wynosi $k = n + 1$ [8].

Celem SVM jest minimalizacja oszacowania ryzyka empirycznego.

6.4.2 Próbki separowalne liniowo

Zakładamy, że dwie klasy można dokładnie rozdzielić za pomocą hiperpłaszczyzny $\boldsymbol{\alpha}$, zdefiniowanej jak (6.19). Wówczas dla każdej próbki możemy otrzymać $y_i = 1$ lub $y_i = -1$. Zachodzi:

$$\mathbf{x}^T \boldsymbol{\alpha} + \alpha_0 \ge +1 \tag{6.20}$$

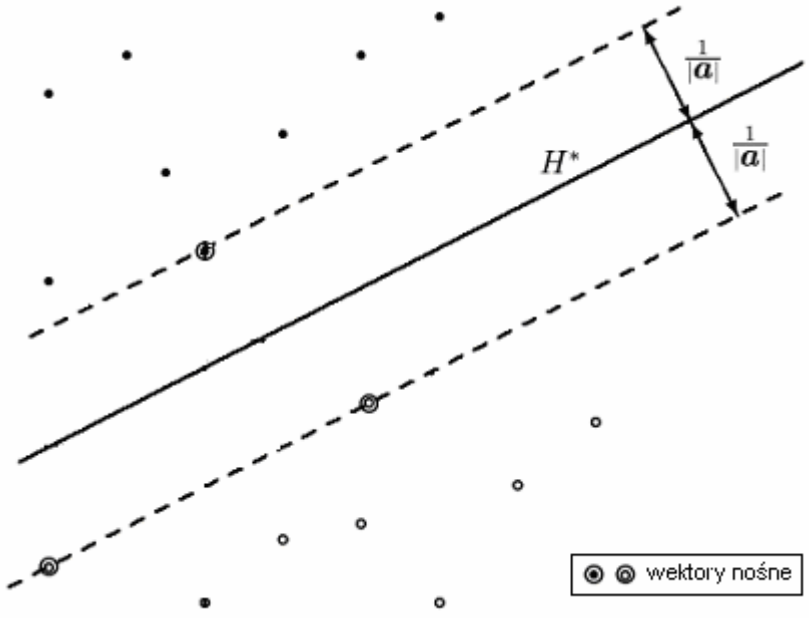
gdy $y_i = 1$ oraz

$$\mathbf{x}^T \boldsymbol{\alpha} + \alpha_0 \leq -1 \tag{6.21}$$

gdy $y_i = -1$.

„Pozytywna" odległość danego punktu od hiperpłaszczyzny zachodzi, gdy punkt leży w kierunku wektora normalnego do powierzchni. Dla przypadku, gdy y_i może przyjmować tylko dwie wartości (próbki klasyfikowane „pozytywnie": $y_i = 1$ i „negatywnie": $y_i = -1$) otrzymamy dwie hiperpłaszczyzny odległe od siebie o $2/|\alpha|$ (tzw. margines), pomiędzy którymi nie leżą żadne próbki uczące. Sensownym kryterium podlegającym optymalizacji jest maksymalizacja odstępu lub minimalizacja $|\boldsymbol{a}|^2$.

Wektorami nośnymi nazywamy te z próbek uczących, które leżą dokładnie na hiperpłaszczyznach. Obecność wektorów nośnych jest warunkiem wystarczającym i koniecznym do wyznaczenia hiperpłaszczyzny optymalnej H*.



Rysunek 6.2 Optymalna hiperpłaszczyzna rozdzielająca H* dla dwóch klas próbek.

Optymalny liniowy klasyfikator wyznaczany jest przez hiperpłaszczyznę H* zdefiniowaną jako:

$$H^*: f(\mathbf{x}, \tilde{\boldsymbol{\alpha}}) = \mathbf{x}_n^T \boldsymbol{\alpha} + \alpha_0 = 0 . \tag{6.22}$$

Hiperpłaszczyznę (6.22) uzyskuje się dzięki minimalizacji kryterium $|\boldsymbol{\alpha}|^2$ oraz przy spełnionych warunkach

$$y_n \left(\mathbf{x}_n^T \boldsymbol{\alpha} + \alpha_0 \right) - 1 \geq 0 \tag{6.23}$$

dla każdego z *N* wektorów **x**. Wówczas reguła decyzyjna ma postać:

$$\mathbf{x} \in \begin{cases} \Omega_1 & : \quad y \geq 0 \\ \Omega_2 & : \quad y < 0 \end{cases} \tag{6.24}$$

Definicja (6.24) odpowiada problemowi optymalizacji metodą wypukłego oprogramowania kwadratowego z nierównościowymi ograniczeniami liniowymi.

Aby zredukować warunki dodatkowe (6.23), wprowadza się mnożniki Lagrange'a $\vartheta = (\vartheta_1, ..., \vartheta_N)^T$. Otrzymujemy w ten sposób funkcjonał

$$L(\tilde{\boldsymbol{\alpha}}, \vartheta) = \frac{|\alpha|^2}{2} - \sum_{n=1}^{N} \vartheta_n \left[y_n \left(\mathbf{x}_n^T \boldsymbol{\alpha} + \alpha_0 \right) - 1 \right], \tag{6.25}$$

który należy minimalizować względem $\boldsymbol{\alpha}$ i α_0 do momentu, gdy nie zanikną pochodne względem ϑ. Dla optymalizacji funkcjonału ***L*** określone są warunki konieczne i wystarczające (Karush-Kuhn-Tucker):

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \boldsymbol{\alpha} - \sum_{n=1}^{N} \vartheta_n y_n \mathbf{x}_n = 0 \tag{6.26}$$

$$\frac{\partial L}{\partial \alpha_0} = -\sum_{n=1}^{N} \vartheta_n y_n = 0 \tag{6.27}$$

$$\begin{gathered} 0 \leq \vartheta_n \\ 0 \leq y_n \left(\mathbf{x}_n^T \boldsymbol{\alpha} + \alpha_0 \right) - 1 \\ 0 \leq \vartheta_n \left[y_n \left(\mathbf{x}_n^T \boldsymbol{\alpha} + \alpha_0 \right) - 1 \right] \end{gathered} \tag{6.28}$$

dla $n = 1, \ldots, N$.

W przypadku maszyn SVM przekształca się problem (6.25) w dualny problem maksymalizacji funkcjonału z warunkiem dodatkowym o zanikaniu pochodnych $\boldsymbol{\alpha}$ i α_0. Jeśli do wzoru (6.25) wstawimy pochodne (6.26) i (6.27), otrzymamy dualny problem postaci Wolfa:

$$L_d(\vartheta) = \sum_{n=1}^{N} \vartheta_n - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \vartheta_i \vartheta_j y_i y_j \left(\mathbf{c}_i^T \mathbf{c}_j\right) \quad , \quad 0 \le \vartheta_n \quad , \quad 0 = \sum_{n=1}^{N} \vartheta_n y_n \tag{6.29}$$

Zadaniem uczenia maszyny SVM jest maksymalizacja funkcjonału $\boldsymbol{L_d}$ względem ϑ przy podanych warunkach dodatkowych, które są niezależne od $\boldsymbol{\alpha}$ i α_0. Po wyznaczeniu ϑ wartości uzyskujemy korzystając ze wzorów (6.26) i (6.27).

Korzystając z definicji (6.25) można przekształcić definicję (6.22) do postaci:

$$H^*: f(\mathbf{x}, \tilde{\boldsymbol{\alpha}}) = \sum_{n=1}^{N} \vartheta_n y_n \left(\mathbf{x}_n^T \mathbf{x}_n\right) + \alpha_0 = 0 \,. \tag{6.30}$$

Na podstawie (6.30) można stwierdzić, że do wyznaczenia optymalnej hiperpłaszczyzny wystarczą wektory cech, dla których mnożnik Lagrange'a wynosi $\vartheta_n > 0$ (wektory nośne).

6.4.3 Próbki nie separowalne liniowo

Jeśli przypadek omawiany w rozdziale 6.4.2 uogólnimy na próbki, których nie da się liniowo odseparować, wówczas w równaniach określających przypisanie próbki do klasy trzeba uwzględnić pewną zmienną $\delta_i>0$, która określa wzajemne „przebicie" obszarów próbek.

Zachodzi:

$$\mathbf{x}^T \boldsymbol{\alpha} + \alpha_0 \ge +1 - \delta \tag{6.31}$$

gdy $y_i = 1$ oraz

$$\mathbf{x}^T \boldsymbol{\alpha} + \alpha_0 \leq -1 + \delta \tag{6.32}$$

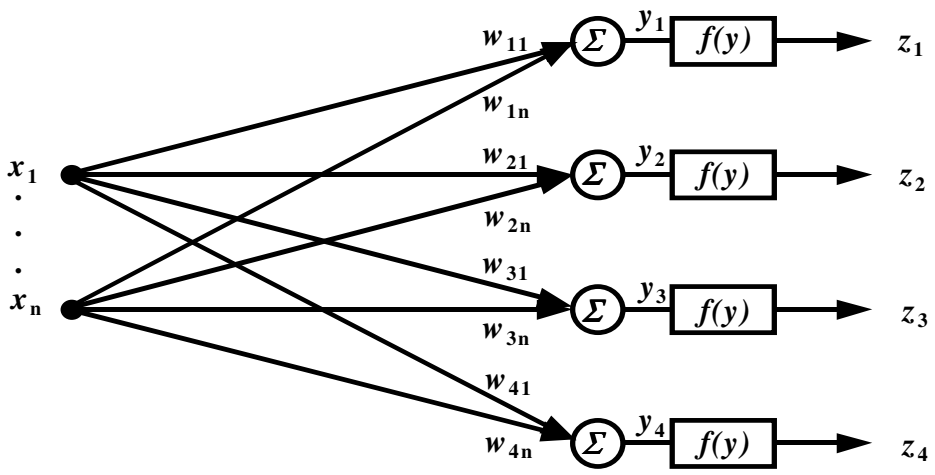
gdy $y_i = -1$.

Błąd klasyfikacji próbki *i*-tej za pomocą hiperpłaszczyzny powstaje wówczas, gdy $\delta_i > 1$. Oszacowaniem takich błędów będzie wówczas suma $\sum_i \delta_i$, która również powinna być minimalizowana.

6.5. Klasyfikator neuronowy

Jeżeli nie zakładamy określonej (zadanej) postaci funkcji rozdzielających, funkcji potencjału lub funkcji rozkładu prawdopodobieństwa, to możemy zastosować klasyfikator zbudowany w oparciu o sieci neuronowe. W przypadku używania sieci neuronowej jako klasyfikatora, sygnałami wejściowymi są wykryte wcześniej cechy obrazu. Liczba neuronów wejściowych jest równa liczbie analizowanych cech, natomiast wyjść jest tyle, ile klas obrazów ([20], [101], [102]). Popularnym klasyfikatorem jest wielowarstwowa sieć jednokierunkowa MLP (perceptron).

W sieci jednowarstwowej neurony ułożone są w jednej warstwie [73]. Połączenie wektora wejściowego neuronami warstwy wyjściowej jest zwykle pełne, to znaczy każdy węzeł wejściowy jest połączony z każdym neuronem wyjściowym. Połączenia reprezentowane są przez macierz wag połączeń **W**.



Rysunek 6.3 Sieć neuronowa jednokierunkowa jednowarstwowa.

W sieciach jednokierunkowych sygnały wejściowe są przesyłane od wejścia do warstwy wyjściowej poprzez połączenia pobudzające. Funkcja pobudzania neuronu wyjściowego ma postać:

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} \tag{6.33}$$

Istnieją również sieci rekurencyjne, które posiadają jedynie połączenia hamujące między neuronami warstwy, zebrane w postaci macierzy wag H. Dla takiej sieci funkcja pobudzenia neuronów wygląda następująco:

$$\mathbf{y} = \mathbf{x} - \mathbf{H} \cdot \mathbf{y} \tag{6.34}$$

W stanie ustalonym liniowa sieć rekurencyjna jest równoważna liniowej sieci jednokierunkowej o postaci:

$$\mathbf{y} = (\mathbf{I} + \mathbf{H})^{-1} \cdot \mathbf{x} \tag{6.35}$$

gdzie **I** oznacza macierz jednostkową.

Typową stosowaną funkcją aktywacji $f(y_i)$ w sieci MLP jest funkcja sigmoidalna unipolarna

$$f(y_i) = \frac{1}{1 + \mathbf{exp}(-\beta y_i)}, \tag{6.36}$$

gdzie β jest zadanym parametrem, a sygnał wyjściowy neuronu przyjmuje wartości z przedziału [0, 1], lub funkcja bipolarna

$$f(y_i) = tgh\left(\frac{\alpha y_i}{2}\right) \tag{6.37}$$

gdzie α jest zadanym parametrem, a sygnał wyjściowy neuronu przyjmuje wartości z przedziału [-1, 1].

W przypadku sieci wielowarstwowych występuje jedna lub więcej warstw ukrytych, oprócz warstwy wejściowej i neuronów wyjściowych ([65], [96]).

Dla każdej warstwy $l = 0, 1, 2, \ldots$ przyjmujemy funkcję:

$$\mathbf{y}_i^{(l)} = \psi\left[W^{(l)}\mathbf{y}_i^{(l-1)} - \Theta^{(l)}\right], \tag{6.38}$$

przy czym $\boldsymbol{y}^0=\boldsymbol{x}$. Indeks i oznacza i-te wyjście w ramach warstwy, natomiast indeks l jest indeksem warstwy. W powyższym wzorze $\psi[]$ oznacza nieliniową funkcję aktywacji, zazwyczaj sigmoidalną funkcję postaci:

$$\psi(y) = \frac{1}{1 + \mathbf{exp}(-y)}, \tag{6.39}$$

o ciągłej pochodnej wynoszącej:

$$\frac{\partial \psi}{\partial y} = \psi[1 - \psi]. \tag{6.40}$$

Oznaczenie Θ_i to wektor dodatkowych wejść, który wyznacza wartość progową dla „szumów pomiarowych”.

W przypadku klasyfikatorów opartych na sieciach neuronowych mamy do czynienia z uczeniem z lub bez nauczyciela [102]. Dla uczenia z nadzorem stosuje się regułę Widrofa-Hoffa („delta”), która mówi, że waga wejściowa i-tego neuronu połączenia z j-tym neuronem wyjściowym jest wzmacniana proporcjonalnie do różnicy pożądanej i rzeczywistej aktywacji.

Dla sieci MLP stosuje się uogólnioną regułę „delta", czyli regułę „wstecznej propagacji błędu" w perceptronie:

$$\Delta \mathbf{W}^{(l)} = \eta \mathbf{d}^{(l)} \left(\mathbf{y}_i^{(l-1)} \right)^T \tag{6.41}$$

gdzie $\boldsymbol{y}^0 = \boldsymbol{x}$, a $\mathbf{d}^{(l)}$ jest wektorem korekcji wag dla neuronów l-tej warstwy.

Dla uczenia bez nadzoru mamy regułę Hebba względnie anty-Hebba [66], która mówi, że waga łuku pobudzającego względnie hamującego jest wzmacniana, gdy aktywacje obu połączonych neuronów są skorelowane (np. PCA lub ICA dla sieci rekurencyjnej). Inaczej mówiąc, jeśli aktywny neuron A jest cyklicznie pobudzany (lub hamowany) przez neuron B, to staje się on jeszcze bardziej czuły na pobudzenie tego neuronu.

6.5.1 Klasyfikator zastosowany w testach

Na potrzeby testów zastosowano neuronową sieć konkurencyjną (ang. *competitive*). Podstawowa sieć konkurencyjna składa się z dwóch warstw – warstwy obliczającej podobieństwo i warstwy konkurencyjnej.

Warstwa wyznaczająca podobieństwo zawiera macierz wag **W**, o rozmiarach $m \times p$. Każdy wiersz macierzy **W** powiązany jest z jednym neuronem. Zadaniem warstwy jest wygenerowanie sygnałów $\mathbf{d}(n)$, które określają podobieństwo (odległość) między wektorem wejściowym $\mathbf{x}(n)$ a każdym z wektorów $\mathbf{w}_i(n)$. Z kolei warstwa konkurencyjna generuje m sygnałów binarnych y_i. Sygnał taki przyjmuje wartość „1" dla neuronu, dla którego sygnał odległości $\mathbf{d}_i$ jest najmniejszy. Innymi słowy $y_i=1$ oznacza, że i-ty wektor $\mathbf{w}_i(n)$ jest najbardziej podobny do sygnału wejściowego $\mathbf{x}(n)$.

W przypadku sieci konkurencyjnej wagi modyfikowane są w odpowiedzi na sygnały wejściowe – jest to uczenie bez nadzoru. Sama warstwa konkurencyjna jest siecią rekurencyjną ze zdefiniowaną macierzą połączeń **M**:

$$\mathbf{M} = \begin{bmatrix} 1 & & \\ & \ddots & -\alpha \\ & -\alpha & \ddots \\ & & 1 \end{bmatrix},$$

gdzie α<1 jest pewną wartością stałą. Macierz **M** określa odpowiedź do pozostałych neuronów o sile – α.

Uczenie w przypadku sieci konkurencyjnej zakłada, że dane wejściowe są podzielone na klastry (w przypadku testowanych obrazów angiograficznych będą to różne klasy obserwowanych naczyń). Zatem każdy z *m* wektorów wagowych **w** reprezentuje środek klastra. Zadaniem uczenia konkurencyjnego jest kwantyzacja przestrzeni wektorów wejściowych.

W testach zastosowano metody dostępne w module sieci neuronowych programu MATLAB. Schemat działania jest następujący:

- Zainicjuj wektory wagowe (np. losowo wybranymi wektorami wejściowymi).

- Dla każdego wektora wejściowego **x**
  - na podstawie odległości między wektorem wagowym **w** a wektorem wejściowym **x** określ neuron najbliższy wektorowi wejściowemu
  - przekształć zwycięski wektor w, by był jak najbliższy wektorowi wejściowemu **x** (nie modyfikuj pozostałych wektorów w):

$$\Delta \mathbf{w}_j(n) = \eta(n) y_j(n) \left[\mathbf{x}^T(n) - \mathbf{w}_j(n) \right] \tag{6.42}$$

- W celu osiągnięcia statycznego wyniku krok uczenia powinien być zmniejszany o pewną stałą wartość, np. $\eta(n) = 0.1\left(1 - \frac{n}{N}\right)$.

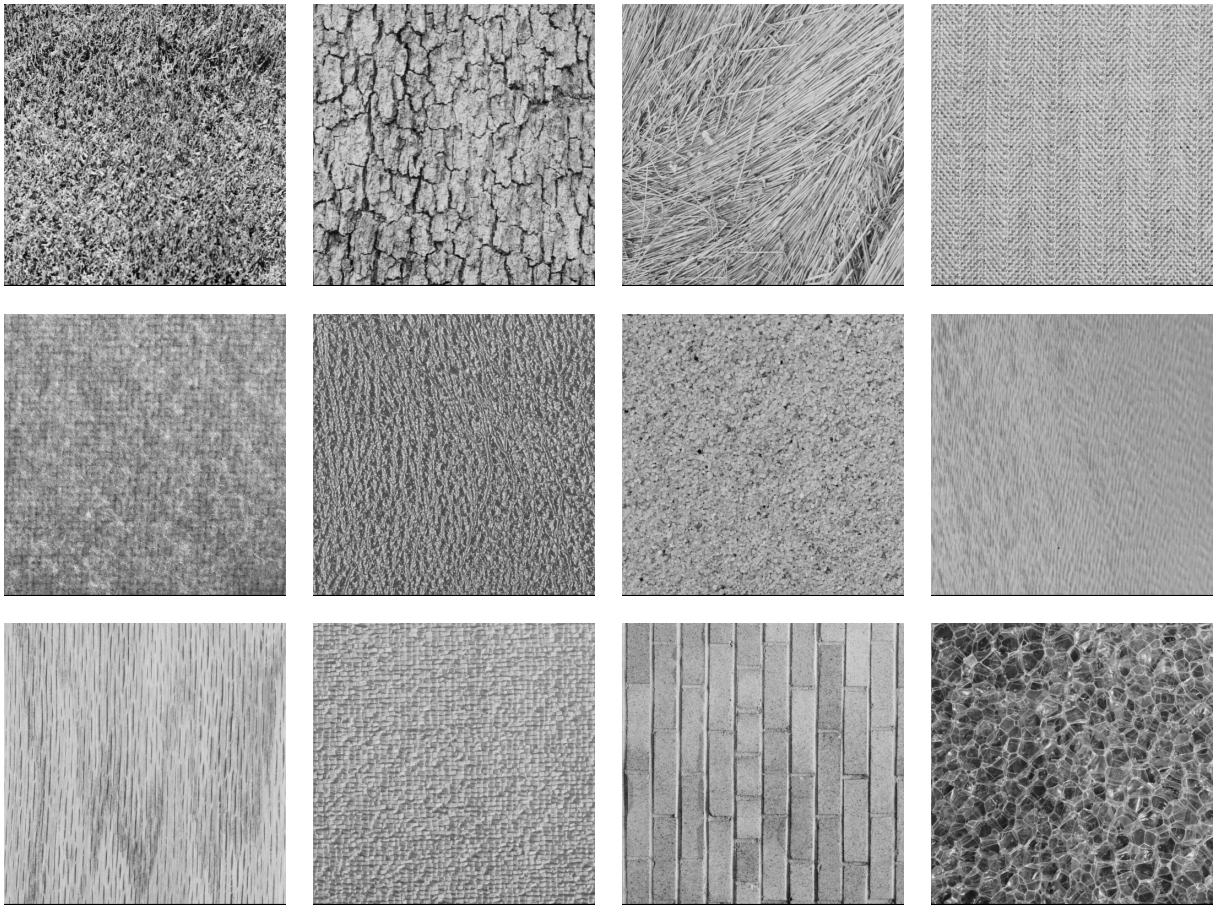
7. Eksperymenty dotyczące detekcji cech tekstur

7.1. Obrazy testowe

Do testów omawianych metod analizy tekstury wybrano dwie grupy obrazów. Pierwszą stanowił tradycyjny zbiór tekstur z albumu Brodatz'a [7], natomiast drugą – obrazy angiograficzne.

7.1.1 Tekstury z albumu Brodatz'a

Album Brodatz'a jest podstawowym źródłem obrazów tekstur dla wielu aplikacji związanych z tą tematyką. Zawiera zbiór obrazów przedstawiających tekstury różnych typów: o różnym stopniu porowatości, regularności wzorca itp. Przykłady tekstur z albumu Brodatz'a przedstawia poniższy Rysunek 7.1.

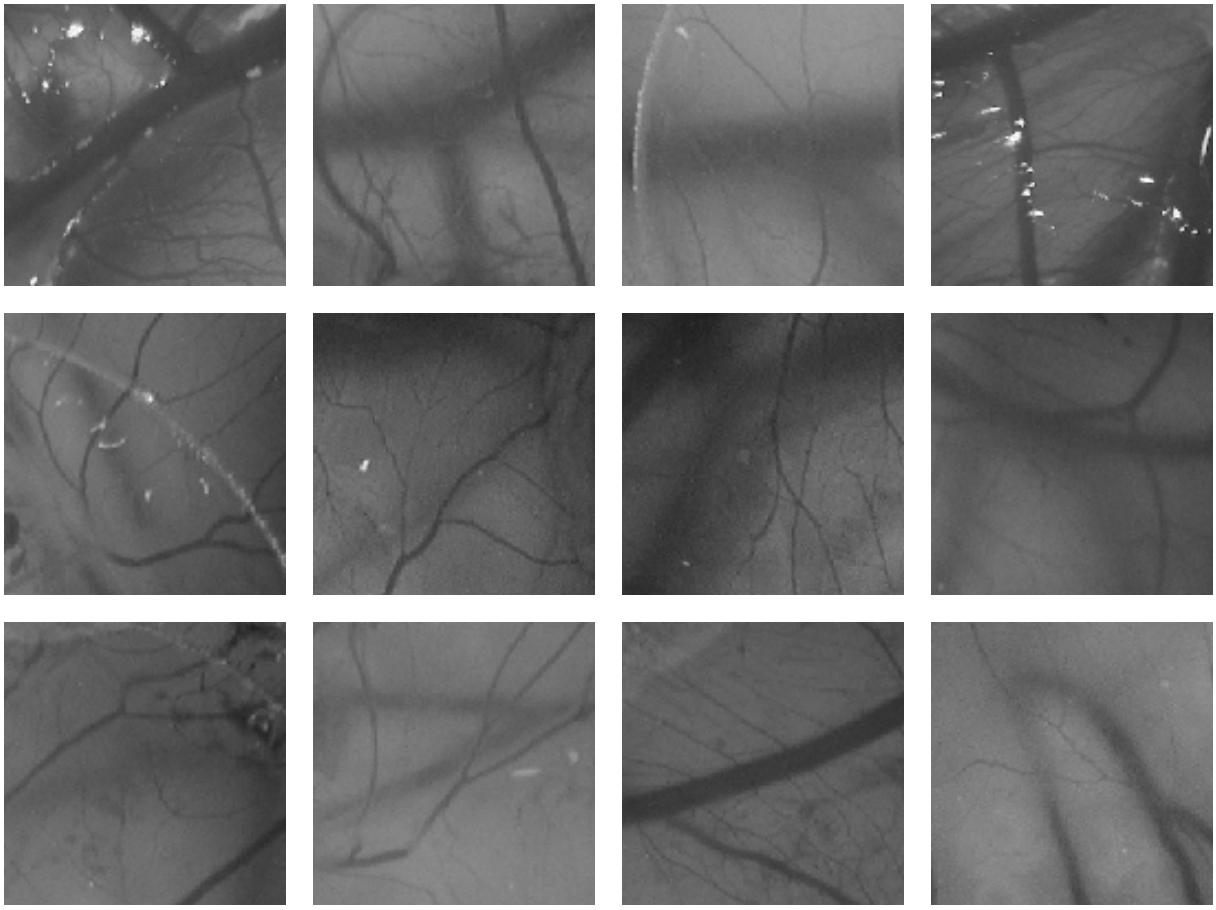


Rysunek 7.1 Przykłady tekstur z albumu Brodatz'a

Tekstury te posłużyły za punkt odniesienia dla badań nad obrazami angiograficznymi i do testów poprawności implementowanych algorytmów.

7.1.2 Obrazy angiograficzne

Aplikacja wyszukuje w badanym obrazie obszary zawierające określone informacje. Dla potrzeb testów odpowiednie obszary zostały oznaczone i opisane przez specjalistów lekarzy biorących udział w grancie 7T08A05016 realizowanym przez autorkę ([4], [93]). Obszary te wykorzystano w niniejszej pracy w procesach uczenia klasyfikatorów. Obszary te zawierają nowo powstałe naczynia krwionośne będące odgałęzieniami już istniejących „głównych" naczyń. O ile za pomocą deskryptora tekstury trudno określić długość i krętość naczynia, o tyle deskryptor ten może przenosić wystarczająco dużo informacji, by wyróżnić klasy obszarów zawierające także naczynia. Takie podejście pozwala na wyróżnienie obrazów zawierających określone obszary lub też na porównanie obrazów jednej serii (np. zdjęcia tkanki pacjenta wykonane w pewnych odstępach czasu).



Rysunek 7.2 Przykłady obrazów angiograficznych (zarodek kurzy)

7.2. Wyniki detekcji cech

7.2.1 Wektory cech oparte o filtry Gabora

Jako metodę wyznaczania wektorów cech wykorzystano w tym przypadku deskryptory tekstury opisane w standardzie MPEG-7 (zob. 2.1.5). Po testach wstępnych zadecydowano, że zastosowany zostanie deskryptor połączony, składający się z deskryptora tekstury jednorodnej HTD i deskryptora przeglądania tekstury TBD. Takie podejście zostało podyktowane faktem, że sam deskryptor przeglądania tekstury dawał zbyt podobne wyniki, by można było przeprowadzić klasyfikację. Można to łatwo wytłumaczyć tym, że TBD jest bardziej ogólny, ponieważ operuje na bardziej percepcyjnych właściwościach obrazu. Wykorzystana seria zawierała bardzo podobne wizualnie obrazy – zdjęcia zarodka kurzego, stąd tak zbliżone wyniki. Natomiast w połączeniu z deskryptorem tekstury jednorodnej otrzymano bardziej zróżnicowane wyniki. Dlatego też TBD może być wykorzystany do rozróżnienia obrazów różnych typów (np. obrazy przedstawiające angiogenezę w tkance skórnej i angiogenezę w

zarodku kurzym), natomiast HTD służy do rozróżniania obrazów należących do tego samego typu.

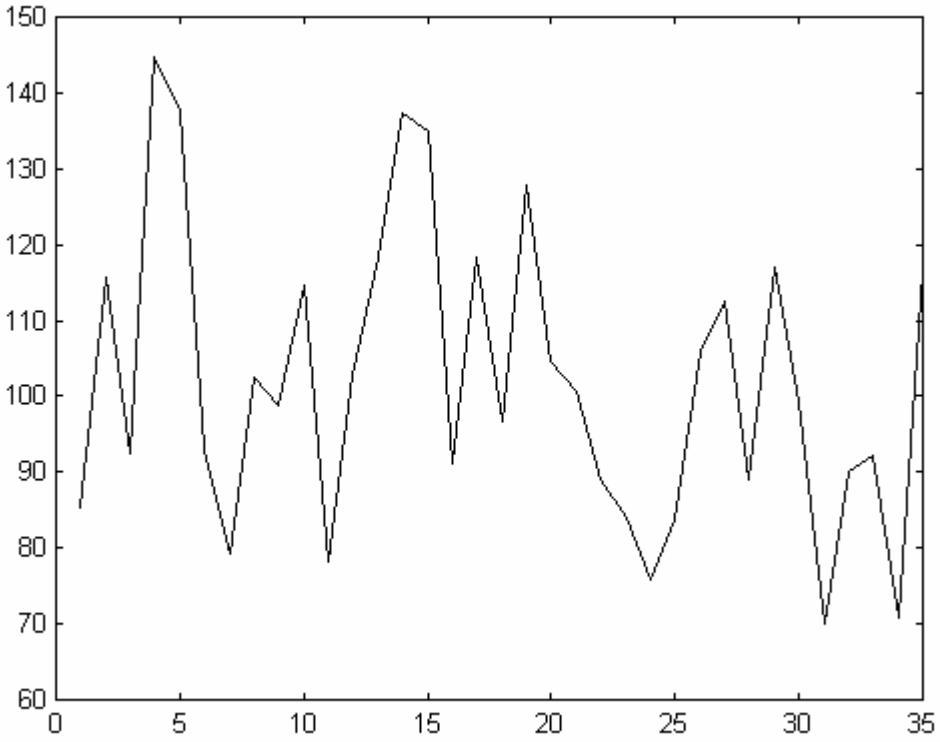
Deskryptor tekstury jednorodnej może składać się z 32 lub 62 elementów i zawiera średnią wartość pikseli i odchylenie standardowe dla nich oraz tabelę wartości energii dla 30 kanałów wynikających z podziału obszaru częstotliwości obrazu. W wersji 62-elementowej zawiera dodatkowo tablicę odchyleń standardowych energii dla 30 kanałów częstotliwości. Tabela 7.1 przedstawia średnie wyniki dla deskryptora tekstury jednorodnej. Testy wykonano dla obrazów w skali szarości, o rozmiarach 64 × 64 pikseli. Badania przeprowadzono dla serii zdjęć przedstawiających rozwój naczyń krwionośnych w zarodku kurzym (obrazy angiograficzne), dla tych samych obrazów przekształconych do postaci krawędziowej oraz tekstur z albumu Brodatz'a.

Tabela 7.1 Średnie wyniki dla deskryptora tekstury jednorodnej. Oznaczenia: μ – średnia intensywność pikseli, σ - odchylenie standardowe, |e| - energia (moduł), $|\sigma_e|$ - odchylenie standardowe energii (moduł)

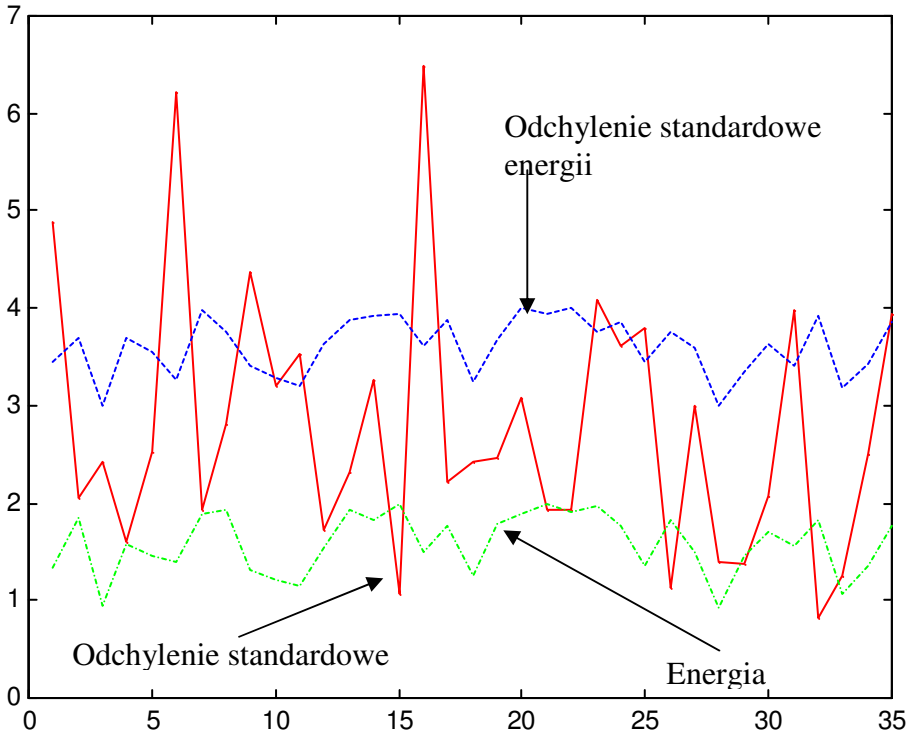
Średnie wartości	**μ**	**σ**	**|e|**	**$\|\sigma_e\|$**
Obrazy angiograficzne	102.0058	2.7855	1.7383	3.6106
Obrazy angiograficzne krawędziowe	28.5703	8.6174	2.1531	4.0427
Tekstury Brodatz'a	151.9115	4.4093	2.2922	4.1044

Tabela 7.1 przedstawia średnie wyniki dla deskryptora tekstury jednorodnej. W trakcie testów zastosowano dwa podejścia dla wyznaczania energii – z uwzględnieniem wartości stałej dla części rzeczywistej współczynników funkcji Gabora i bez uwzględnienia tej wartości. Współczynniki te pozwalają na wyznaczenie energii i jej odchylenia standardowego dla deskryptora HTD oraz elementów deskryptora TPD. W dalszych rozważaniach pominięto wartości wyznaczone dla TBD jako mniej istotne. W trakcie testów okazało się, że wartość stała części rzeczywistej współczynników funkcji Gabora ma znikomy wpływ na wartości elementów deskryptora (wartości są sobie równe do 5-6 miejsca po przecinku).

Na rysunkach poniżej (Rysunek 7.3 i Rysunek 7.4) widoczny jest rozrzut między wartościami średniej intensywności pikseli i odchylenia standardowego oraz płaski przebieg wykresów dla wartości energii i jej odchylenia standardowego.



Rysunek 7.3 Średnia intensywność pikseli dla deskryptora HTD dla serii 35 obrazów angiograficznych.



Rysunek 7.4 Przykładowe wyniki HTD dla serii 35 obrazów angiograficznych: odchylenie standardowe (czerwona linia), energia (moduł, zielona linia) oraz odchylenie standardowe energii (moduł, niebieska linia)

W trakcie testów badany był również rozbudowany deskryptor, powstały w wyniku połączenia deskryptora tekstury jednorodnej HTD oraz deskryptora przeglądania tekstury TBD. Przypomnijmy, że deskryptor TPD składa się z 3 lub 5 elementów: regularności wzorca oraz jednego lub dwóch kierunków dominujących i przypisanych do nich skal (zob. 2.2.3). Testy wykazały nieistotny w praktyce wpływ wartości elementów tego deskryptora dla obrazów angiograficznych, będących głównym obiektem badań. Wynika to z tego, że TPD służy głównie do zgrubnej klasyfikacji ([1]) i nie przenosi w sobie zbyt szczegółowych informacji. Dlatego też w przypadku obrazów bardzo do siebie podobnych wizualnie, jak ma to miejsce w przypadku obrazów angiograficznych, jego wpływ jest niewielki lub żaden. Znaczenie miał w tym przypadku również fakt, że w testach wykorzystano obrazy jednego typu, czyli przedstawiające angiogenezę w zarodku kurzym. W przypadku, gdyby roważane były obrazy przedstawiające różne typy angiogenezy, wartości elementów TPD miałyby większe znaczenie.

7.2.2 Wektory cech oparte o PCA

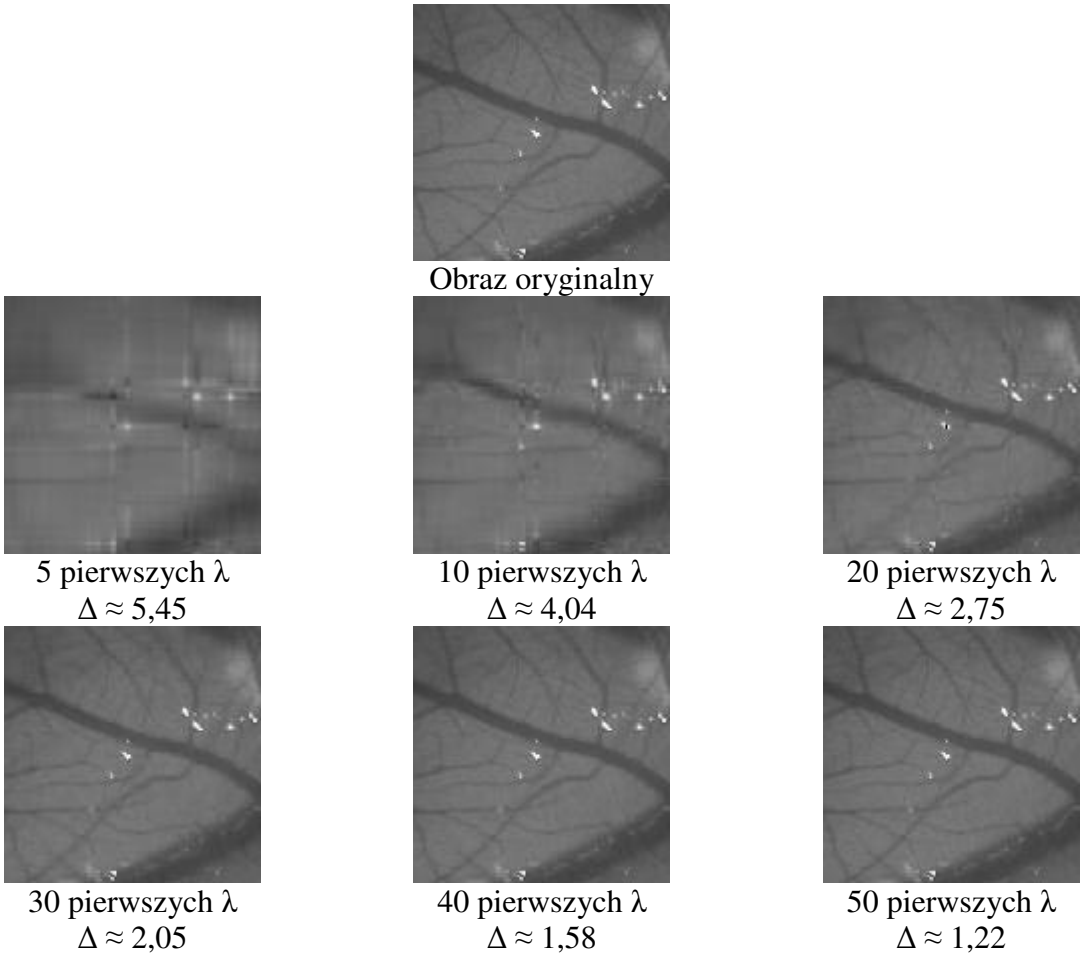
Do wyznaczania cech tekstury można również zastosować przekształcenie PCA (zobacz rozdział 3.1). W tym przypadku jako elementy wektora cech traktowane są składowe główne dla przekształconego obrazu.

Do testów wykorzystano obrazy w skali szarości, o rozmiarach 64 × 64 pikseli. Do wyznaczenia wektora cech wybrano algorytm SVD (ang. *singular value decomposition*), który przekształca dane wejściowe **X** (macierz o rozmiarach $m \times n$) do postaci:

$$\mathbf{X} = \mathbf{U} * \mathbf{S} * \mathbf{V}^T \tag{7.1}$$

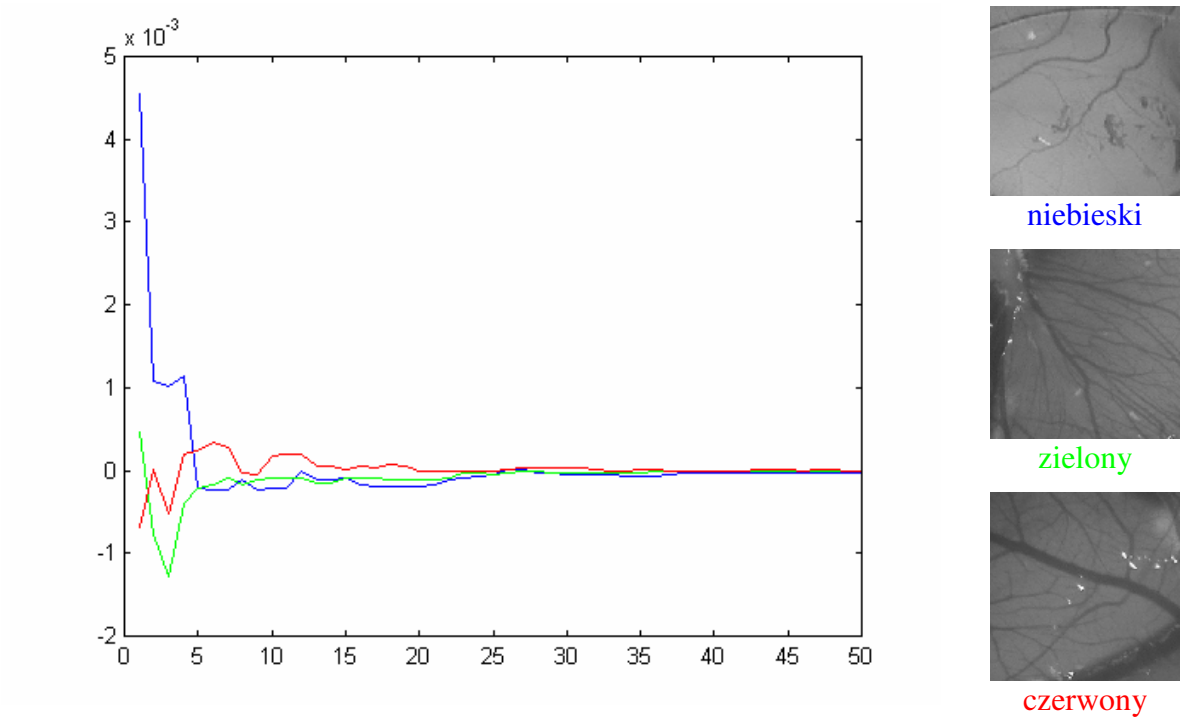
Macierz U ma rozmiary $m \times n$, pozostałe dwie - $n \times n$. Macierz **V** zawiera składowe główne, natomiast macierz **S** jest diagonalna a jej niezerowe wartości są wartościami własnymi danych wejściowych. Przyjmuje się, że wartości własne sortowane są w porządku malejącym – pozwala to na wybranie tylko k największych wartości, a więc ograniczenie rozmiaru wektora wyjściowego.

W przypadku obrazów testowanych na potrzeby niniejszej pracy maksymalny rozmiar wektora testowego wynosił 64 (tyle, ile wymiary danych wejściowych). Poniższa tabela przedstawia przykłady obrazów uzyskanych po ograniczeniu liczby składowych:



Rysunek 7.5 PCA: Przykładowe obrazy uzyskane dla określonej liczby najwyższych wartości własnych λ.

Rysunek 7.5 zawiera również informacje o błędach dla obrazów po przekształceniach. Jako błąd Δ przyjęto wartość oczekiwaną różnicy między obrazem oryginalnym i obrazem po przekształceniach. Mniejsza wartość błędu oznacza lepszą jakość odtworzonego obrazu.



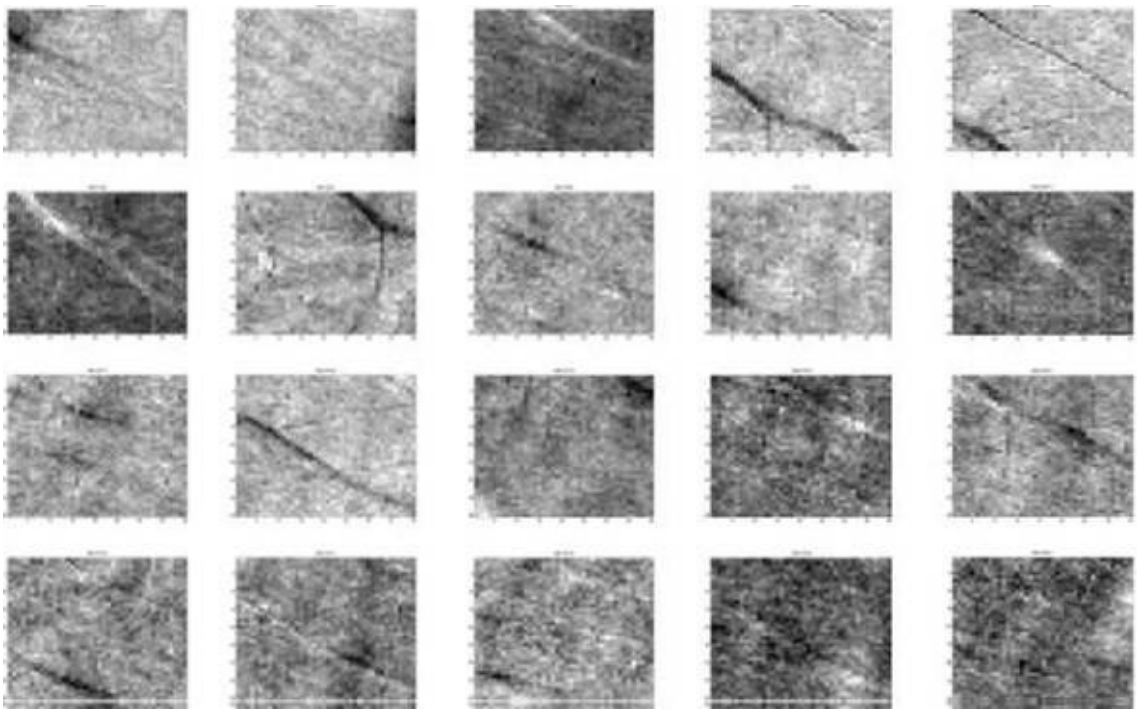
Rysunek 7.6 Wartość błędu dla obrazów uzyskanych w przekształceniu PCA dla pierwszych 50 największych wartości własnych. Wyznaczone dla przykładowych obrazów angiograficznych.

Przyjmując jako kryterium minimalizację wartości oczekiwanej różnicy między obrazem oryginalnym a obrazem po przekształceniach (rozumianej jako błąd), można zauważyć, że nie jest konieczne uwzględnianie wszystkich składowych w dalszym przetwarzaniu.

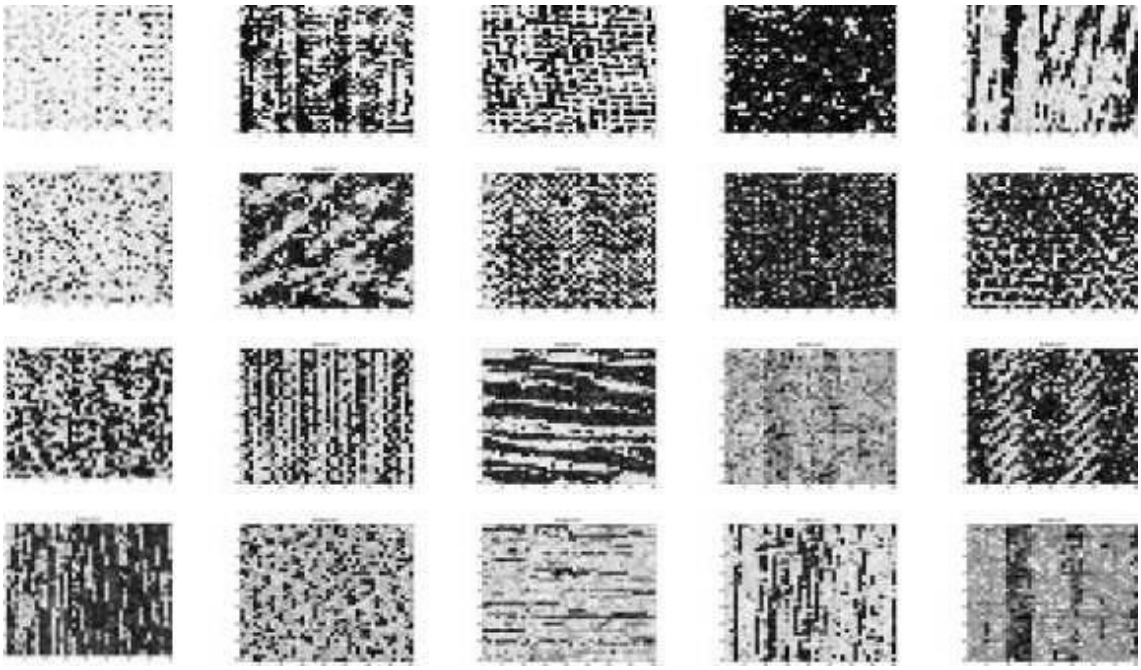
7.2.3 Wektory cech oparte o ICA

Do uzyskania wartości elementów tego wektora cech (współczynników w przekształceniu ICA) wykorzystano autorskie podejście przedstawione w rozdziale 4.2.2 [95].

Podobnie jak dla omawianych do tej pory metod, materiałem testowym były obrazy angiograficzne, przedstawiające proces angiogenezy w zarodku kurzym. Dla porównania wykonano również testy przykładowych tekstur z albumu Brodatza. Wyniki te zostaną wykorzystane w wyznaczeniu i porównianiu wartości dyskryminanu Fishera dla wektorów cech wyznaczonych różnymi metodami (zobacz następny rozdział).



Rysunek 7.7 Wektory bazowe dekompozycji ICA, przedstawione w postaci obrazów dla próbki uczącej (obrazy angiograficzne).



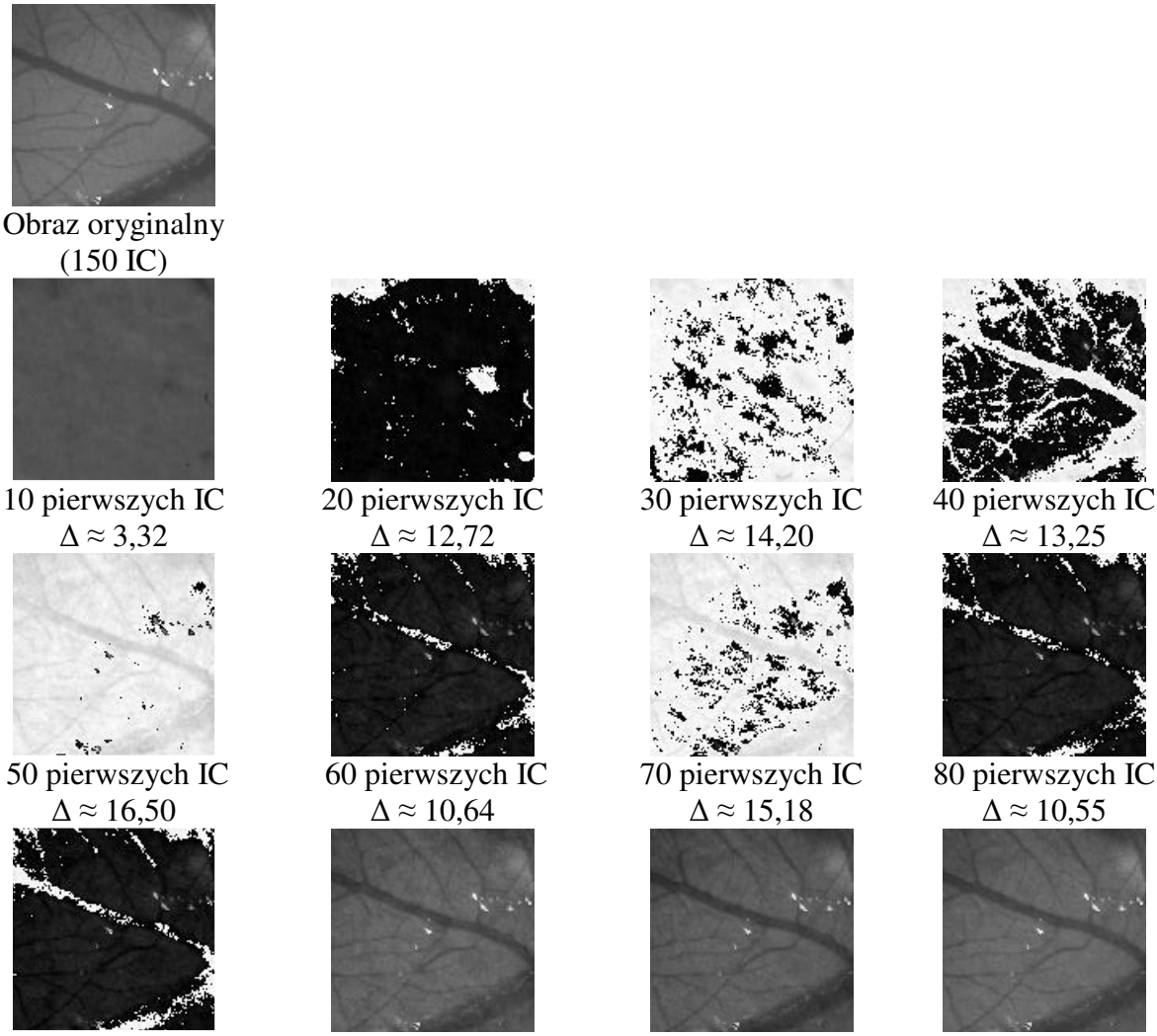
Rysunek 7.8 Wektory bazowe dekompozycji ICA, przedstawione w postaci obrazów, dla wybranych przykładów tekstur z albumu Brodatz'a

Przy zastosowaniu metody ICA wektor cech wyznaczono jako zbiór elementów macierzy wymieszania – zostało to omówione w rozdziale 4.3. Przypomnijmy, że dane wejściowe traktowane są jako mieszanina różnych sygnałów. ICA wyznacza macierz współczynników niezależnych oraz macierz wymieszania. Podobnie jak w przypadku metody PCA, można

również ograniczyć rozmiary danych (np. wyznaczając tylko współczynniki dla *n* największych wartości własnych).

Tak jak oczekiwano, algorytm ICA daje bardziej zróżnicowane wyniki (zob. rozdział 7.3), co ułatwia odpowiednią klasyfikację obrazów jednego typu. W przypadku wektora cech opartego o filtry Gabora nie dało się w pełni wykorzystać jego możliwości właśnie ze względu na duże podobieństwo między analizowanymi obrazami.

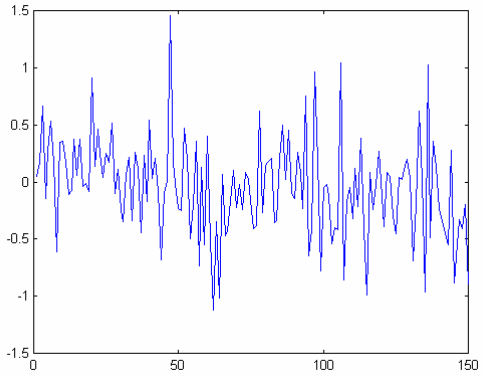
Poniższy rysunek przedstawia przykładowe wyniki obrazy odtworzone na podstawie danych wyznaczonych za pomocą ICA dla ograniczonej liczby składowych niezależnych.



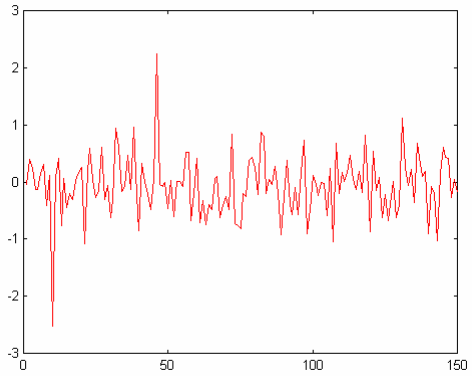
Rysunek 7.9 ICA: Przykładowe obrazy uzyskane dla ograniczonej liczby najwyższych składowych niezależnych.

Jako kryterium ograniczania liczby składowych niezależnych wybrano minimalizację średniego błędu przekształcenia, traktowanego jako różnicę między obrazem oryginalnym i obrazem odtworzonym. Jak wynika z powyższego rysunku, można ograniczyć liczbę składowych niezależnych, niemniej jednak wymaga to wcześniejszych testów – zbyt ograniczona liczba składowych niezależnych może doprowadzić do powstania błędów w późniejszym przetwarzaniu danych.

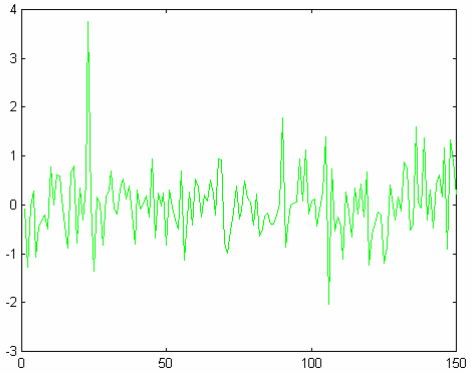
Testy omawiane w rozdziale 8 zakładają przeprowadzenie klasyfikacji dla wyznaczonych wektorów cech. Obrazy powinny być rozdzielone pomiędzy cztery klasy. Poniższy wykres przedstawia współczynniki wymieszania dla obrazów różnych klas (dane obrazów przekształcono do postaci jednowymiarowej).



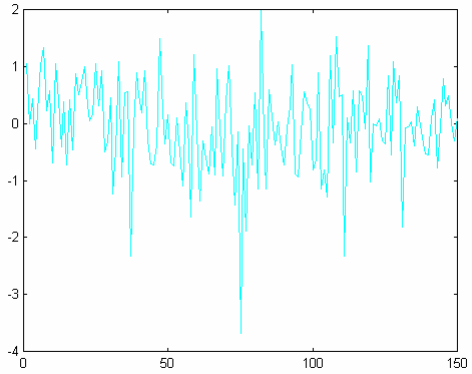
Rysunek 7.10 ICA: wykres wartości współczynników wymieszania dla obrazu klasy A.



Rysunek 7.11 ICA: wykres wartości współczynników wymieszania dla obrazu klasy B.

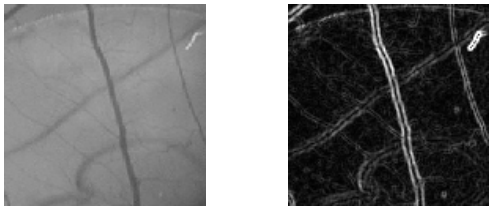


Rysunek 7.12 ICA: wykres wartości współczynników wymieszania dla obrazu klasy C.



Rysunek 7.13 ICA: wykres wartości współczynników wymieszania dla obrazu klasy D.

Dla każdego typu wektora cech wykonano testy dla obrazów oryginalnych (tzn. bez dodatkowej obróbki) oraz dla obrazów krawędziowych.



Rysunek 7.14 Przykładowy obraz angiograficzny i odpowiadający mu obraz krawędziowy.

7.3. Porównanie wektorów cech

7.3.1 Kryterium porównywania obszaru cech

Liniowa analiza dyskryminacyjna (DA) jest to takie przekształcenie przestrzeni, które pozwala na optymalną separację podobszarów cech tworzących różne klasy – w tym sensie jest to więc klasyfikator. Kryterium optymalizacyjnym podlegającym minimalizacji podczas wyznaczania DA jest tzw. informacja Fishera (lub dyskryminant Fishera).

Analiza dyskryminacyjna należy do rodzaju kryteriów, dla których następuje maksymalna koncentracja cech jednej klasy i/lub maksymalne rozdzielenie cech różnych klas. Jeśli przyjmiemy, że **c** oznacza wektor cech wyznaczony za pomocą opisanych wcześniej metod:

$$c_j = \mathbf{\Phi} \cdot f_j, \tag{7.2}$$

(gdzie f oznacza informacje poddane analizie, j=1, …, N) stosowane są następujące kryteria:

a) średni odstęp kwadratowy jednej wartości cechy od każdej innej

$$s_1 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\mathbf{c}_i - \mathbf{c}_j\right)^T \left(\mathbf{c}_i - \mathbf{c}_j\right) \tag{7.3}$$

b) średni odstęp kwadratowy cech jednej klasy od cech innych klas (k – liczba klas)

$$s_2 = \frac{2}{k(k-1)} \sum_{\kappa=2}^{k} \sum_{\lambda=1}^{\kappa-l} \frac{1}{N_\kappa N_\lambda} \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\lambda} ({}^{\kappa}\mathbf{c}_i - {}^{\lambda}\mathbf{c}_j)^T ({}^{\kappa}\mathbf{c}_i - {}^{\lambda}\mathbf{c}_j) \tag{7.4}$$

c) średni odstęp kwadratowy cech jednej klasy

$$s_3 = \frac{1}{k} \sum_{\kappa=2}^{k} \frac{1}{N_\kappa^2} \sum_{i=1}^{N_\kappa} \sum_{j=1}^{N_\lambda} ({}^{\kappa}\mathbf{c}_i - {}^{\kappa}\mathbf{c}_j)^T ({}^{\kappa}\mathbf{c}_i - {}^{\kappa}\mathbf{c}_j) \tag{7.5}$$

d) kombinacja s_2 i s_3 (np. wspomniany dyskryminant Fishera)

Każde z kryteriów s_i zależy od macierzy przekształcenia $\mathbf{\Phi}$, czyli $s_i=s_i(\mathbf{\Phi})$. Niech $\mathbf{\Phi}^{(i)}$ oznacza macierz optymalizującą kryterium s_i. Macierz optymalizującą uzyskamy w wyniku obliczenia wektorów własnych $\boldsymbol{\varphi}_\upsilon{}^{(i)}$ odpowiedniej macierzy $\mathbf{Q}^{(i)}$. Macierz ta wyznaczana jest na podstawie równania:

$$\mathbf{Q}^{(i)} \varphi_\upsilon^{(i)} = \lambda_\upsilon^{(i)} \varphi_\upsilon^{(i)} \tag{7.6}$$

gdzie $\lambda_\nu{}^{(i)}$ są wartościami własnymi $\mathbf{Q}^{(i)}$.

Dla maksymalizacji kryteriów s_1 lub s_2 należy obliczyć n wektorów własnych $\boldsymbol{\varphi}_\upsilon$, odpowiadających n największym wartościom własnym λ_ν macierzy $\mathbf{Q}^{(1)}$ lub $\mathbf{Q}^{(2)}$ odpowiednio. Natomiast dla minimalizacji s_3 należy obliczyć wektory własne odpowiadające najmniejszym wartościom własnym macierzy $\mathbf{Q}^{(3)}$.

Uzyskane w ten sposób wektory $\boldsymbol{\varphi}_\upsilon{}^{(i)}$ tworzą n wierszy macierz $\mathbf{\Phi}^{(i)}$:

$$\mathbf{\Phi}^{(i)} = \begin{pmatrix} \varphi_1^{(i)T} \\ \varphi_2^{(i)T} \\ \vdots \\ \varphi_n^{(i)T} \end{pmatrix} \tag{7.7}$$

Macierze $\mathbf{Q}^{(i)}$ zdefiniowane są następująco:

a) dla kryterium s_1:

$$\begin{aligned}\mathbf{Q}^{(1)} &= \mathbf{R} - \mathbf{m}\mathbf{m}^T \\ \mathbf{R} &= \frac{1}{N}\sum_{j=1}^{N} f_j f_j^T \\ \mathbf{m} &= \frac{1}{N}\sum_{j=1}^{N} f_j\end{aligned} \tag{7.8}$$

b) dla kryterium s_2:

$$\begin{aligned}\mathbf{Q}^{(1)} &= \frac{1}{k}\sum_{\kappa}^{k}\mathbf{R}_\kappa - \frac{1}{k(k-1)}\sum_{\kappa=2}^{N_\kappa}\sum_{\lambda=1}^{\kappa-1}\left(\mathbf{m}_\kappa\mathbf{m}_\lambda^T + \mathbf{m}_\lambda\mathbf{m}_\kappa^T\right) \\ \mathbf{R}_\kappa &= \frac{1}{N_\kappa}\sum_{j=1}^{N_\kappa} {}^{\kappa}f_j\, {}^{\kappa}f_j^T \\ \mathbf{m}_\kappa &= \frac{1}{N_\kappa}\sum_{j=1}^{N_\kappa} {}^{\kappa}f_j\end{aligned} \tag{7.9}$$

c) dla kryterium s_3:

$$\mathbf{Q}^{(3)} = \frac{1}{k}\sum_{\kappa=1}^{k}\left(\mathbf{R}_\kappa - \mathbf{m}_\kappa\mathbf{m}_\kappa^T\right) \tag{7.10}$$

gdzie f oznacza informację poddaną analizie.

Załóżmy, że istnieje przekształcenie $\mathbf{y}=f(\boldsymbol{x})=\mathbf{w}^T\boldsymbol{x}$, które odwzorowuje próbki $\boldsymbol{x}_i$, i=1, …, d, należące do dwóch różnych klas $\boldsymbol{\Omega}_k$, k≥2. Należy znaleźć taki wektor $\mathbf{w}$, aby punkty dla jednej klasy były dobrze odseparowane od punktów należących do innej klasy. Miarą separacji jest odległość między średnimi wartościami punktów klas. Jeśli przyjmiemy wartość średnią dla danej klasy (centrum obszaru) $\boldsymbol{\Omega}_k$ jako

$$\mathbf{m}_k = \frac{1}{d_k}\sum_{x\in\Omega_k}\mathbf{x} \tag{7.11}$$

oraz

$$\tilde{\mathbf{m}}_k = \frac{1}{d_k}\sum_{x\in\Omega_k}\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{m}_k \tag{7.12}$$

to wówczas miara separacji między dwiema klasami może zostać wyliczona ze wzoru:

$$\left|\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}}_j\right| = \left|\mathbf{w}^T\left(\mathbf{m}_i - \mathbf{m}_j\right)\right|. \tag{7.13}$$

Miarę rozrzutu wartości klas określamy jako

$$\tilde{s}_i^2 = \sum_{y \in Y_i} \left(y - \tilde{m}_i\right)^2 . \tag{7.14}$$

Stąd $\frac{1}{d}\sum_k \tilde{s}_k^2$ jest oszacowaniem wariancji wektorów cech dla każdej klasy, zaś $\sum_k \tilde{s}_k^2$ jest łącznym rozrzutem w ramach klas.

Liniowy dyskryminant Fishera oznacza taką funkcję liniową $f(\boldsymbol{x})=\mathbf{w}_{\mathrm{m}}{}^{\mathrm{T}}\boldsymbol{x}$, dla której funkcja jakości $J(\mathbf{w})$ osiąga swoje maksimum. Definiujemy macierze $\mathbf{S}_{\mathrm{i}}$ rozrzutu dla każdej klasy $\boldsymbol{\Omega}_{\mathrm{k}}$:

$$\mathbf{S}_i = \sum_{x \in \Omega_k} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T . \tag{7.15}$$

Macierz rozrzutów w ramach klasy zdefiniowana jest jako suma macierzy rozrzutu wszystkich klas:

$$S_W = \sum_{i=1}^{k} S_i = \sum_{i=1}^{k} \sum_{j=1}^{d_i} (\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)^T , \tag{7.16}$$

natomiast macierz rozrzutów pomiędzy klasami wyznaczana jest według wzoru:

$$S_B = \sum_{i=1}^{k} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T , \tag{7.17}$$

gdzie **m** jest centralnym punktem zbioru wszystkich próbek.

Funkcja jakości wynosi:

$$J(w) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \tag{7.18}$$

Poszukiwana jest maksymalizacja stosunku wyznacznika macierzy rozrzutów pomiędzy klasami do wyznacznika macierzy rozrzutów w ramach klas. Relacja ta jest maksymalna, gdy wektory kolumnowe $\mathbf{w}_m$ macierzy $\mathbf{W}$ są wektorami własnymi macierzy $\mathbf{S}_W^{-1}\,\mathbf{S}_B$.

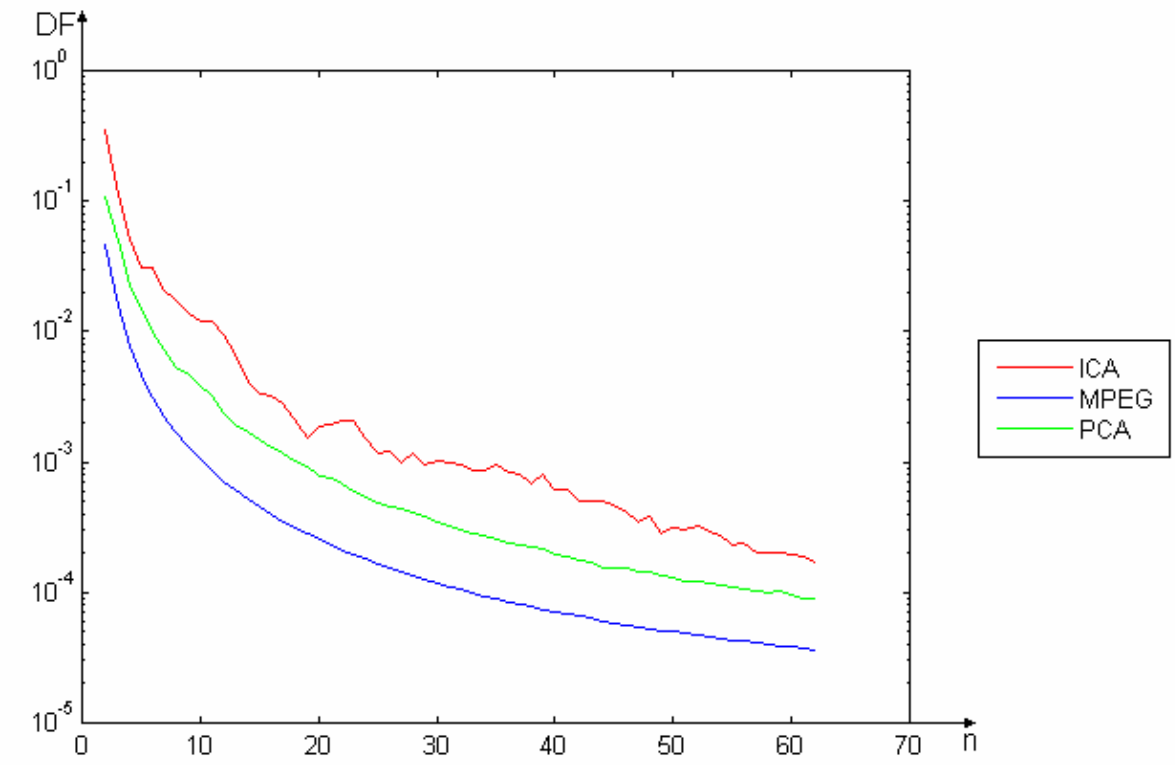
7.3.2 Wyniki porównania obszarów cech

Poniższa tabela i wykres przedstawiają wyniki zastosowania dyskryminantu Fishera do porównania jakości wektorów cech wyznaczonych za pomocą różnych algorytmów. Można zaobserwować istotną zmianę stosunku wartości dyskryminatora w zależności od typu obrazów testowych. Szczególnie duże różnice widać przy zastosowaniu wektora cech opartego o algorytmy ICA. Dla wektora cech tekstury uzyskanego metodą PCA wartość funkcji jakości jest wyższa niż dla wektora opartego o algorytmy MPEG, ale wciąż pozostaje niższa niż dla wektora cech ICA.

Tabela 7.2 Dyskryminant Fishera: wartość funkcji jakości dla wektorów cech (ograniczonych do pierwszych 62 elementów) dla obrazów angiograficznych.

Metoda detekcji cech	Wartość
ICA	$1{,}676 \cdot 10^{-4}$
MPEG	$3{,}645 \cdot 10^{-5}$
PCA	$9{,}03 \cdot 10^{-5}$

Rysunek 7.15 przedstawia zależność wartości dyskryminantu Fishera dla różnych metod wyznaczania wektora cech dla obrazów angiograficznych. Liczbę uwzględnionych elementów wektora cech ograniczono do 62 (wektory cech wyznaczone różnymi metodami mogą mieć różne długości). Celem lepszego uwidocznienia różnic między metodami oś Y wykresu, odpowiadającą wartościom dyskryminantu, przedstawiono w skali logarytmicznej.



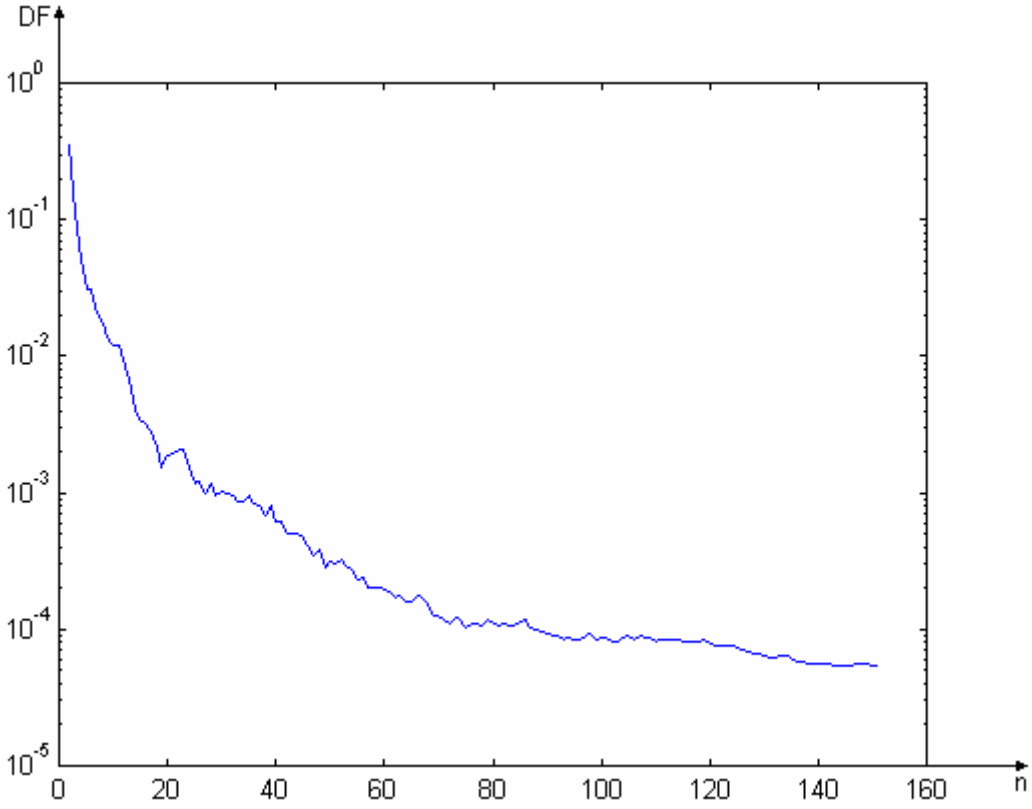
Rysunek 7.15 Dyskryminant Fishera: przebieg wartości funkcji jakości dla pierwszych 62 elementów wektorów cech wyznaczonych różnymi metodami dla obrazów angiograficznych.

Przypomnijmy tutaj, że funkcja jakości zależy od stosunku wyznacznika rozrzutów pomiędzy klasami do wyznacznika rozrzutów wewnątrz klas (wzór 7.18), a więc należy dążyć do maksymalizacji tego stosunku. Tabela 7.2 pokazuje, że dla wektora cech zbudowanego z współczynników przekształcenia ICA funkcja jakości jest najwyższa w porównaniu do wektorów opartych o współczynniki PCA czy filtrów Gabora. Jeśli porównamy wyniki dla obrazów angiograficznych i tekstur Brodatz'a, funkcja jakości zbioru cech opartego o współczynniki ICA jest wyższa dla obrazów angiograficznych (Tabela 7.3). Można więc stwierdzić, że opis cech w oparciu o algorytm ICA charakteryzuje się najwyższą funkcją jakości spośród badanych deskryptorów.

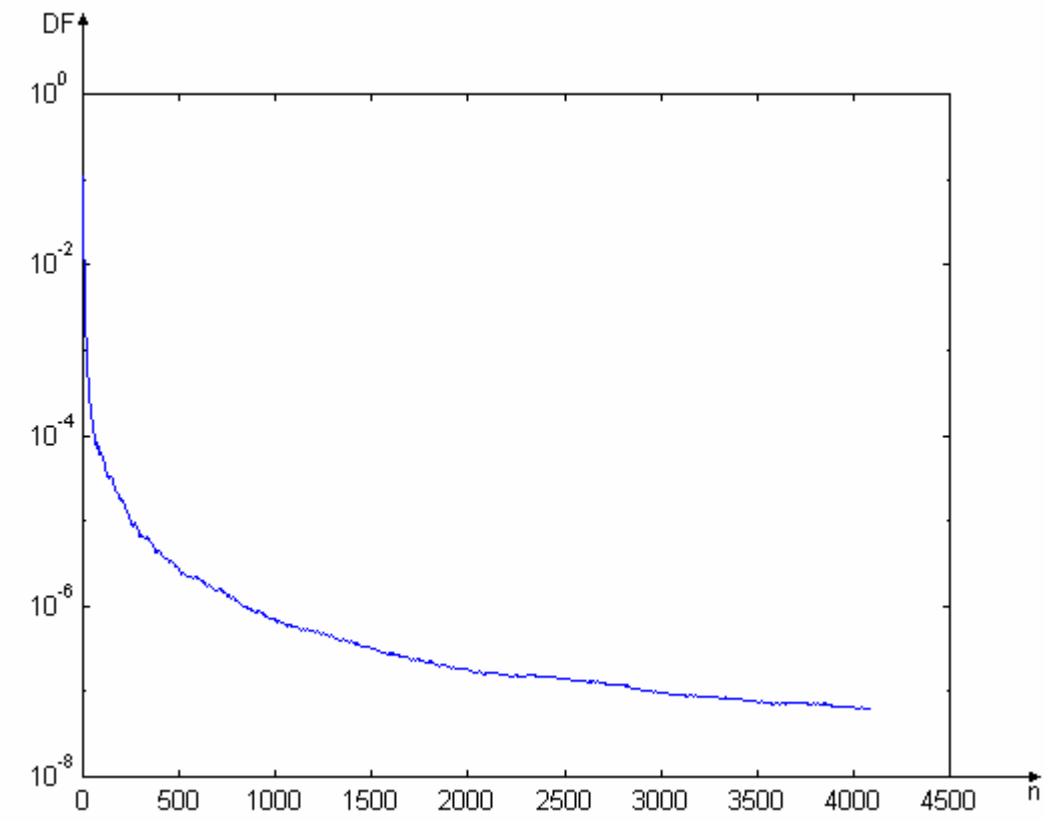
Tabela 7.3 Dyskryminant Fishera: wartość funkcji jakości dla wektorów cech (ograniczonych do pierwszych 62 elementów) dla obrazów tekstur z albumu Brodatza.

Metoda detekcji cech	**Wartość**
MPEG	$7{,}419 \cdot 10^{-6}$
PCA	$1{,}601 \cdot 10^{-4}$
ICA	$8{,}15 \cdot 10^{-4}$

Rysunek 7.15 ma taką samą wymowę, jak Tabela 7.2. Wartość dyskryminantu, a więc funkcji jakości, pozostaje najwyższa dla wektora cech wyznaczonego algorytmem ICA. Wszystkie wektory cech ograniczono do 62 elementów, aby porównanie miało sens. Wartość funkcji jakości maleje wraz ze wzrostem liczby uwzględnianych współczynników, ale po przekroczeniu pewnej liczby współczynników zaczyna utrzymywać się na stałym poziomie. Można to zaobserwować na poniższych wykresach. Rysunek 7.16 przedstawia przebieg wartości dyskrymiantu Fishera dla wektora cech wyznaczonego algorytmem ICA w skali logarytmicznej). Uwzględniono 150 pierwszych współczynników. Z kolei rys. 2 przedstawia przebieg wartości funkcji jakości w skali logarytmicznej dla wektora cech opartego o PCA – uwzględniono ponad 4000 współczynników.



Rysunek 7.16 Dyskryminant Fishera: przebieg wartości funkcji jakości dla 150 współczynników wektora cech wyznaczonego algorytmem ICA dla obrazów angiograficznych.



Rysunek 7.17 Dyskryminant Fishera: przebieg wartości funkcji jakości dla 4096 współczynników wektora cech wyznaczonego algorytmem PCA dla obrazów angiograficznych.

Badane wektory cech różniły się liczbą elementów – dla wektora opartego o standardy MPEG ilość elementów jest ściśle określona i zależy od rodzaju użytego deskryptora – na przykład dla HTD (ang. *Homogeneous Texture Descriptor*) wynosi 62.

Dla wektora cech wyznaczanego algorytmem PCA ilość elementów zależy od rozmiaru obrazu (obrazy testowe miały rozmiar 64 × 64 pikseli). Jak wspomniano wcześniej, można tę ilość ograniczyć, bazując na wybranym kryterium poprawności przekształconego obrazu. W testach nie ograniczano rozmiaru wektora ze względu na dokładność wyników.

Liczba elementów wektora cech wyznaczonego metodą ICA zależy od ilości danych (liczby przetwarzanych sygnałów). Również w tym przypadku nie ograniczono rozmiaru wektora, podobnie jak w przypadku PCA kierując się dokładnością wyników.

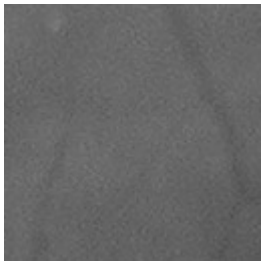
8. Eksperymenty dotyczące klasyfikacji tekstur

8.1. Założenia

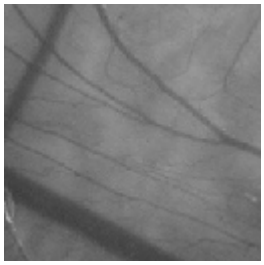
Opisane we wcześniejszych rozdziałach sposoby wyznaczania cech tekstury wykorzystano do klasyfikacji obrazów testowych. Zastosowano cztery typy klasyfikatora: według najmniejszej odległości, klasyfikator Bayesa, neuronowy i SVM. Celem testów było sprawdzenie skuteczności wybranych algorytmów wyznaczania wektorów cech.

Do testów wykorzystano obrazy testowe angiograficzne. Obrazy te dzielone są na cztery grupy w zależności od charakteru przedstawionej na nich siatki naczyń krwionośnych. Testy wykonane zostały dla obrazów według skali szarości oraz dla tych samych obrazów przekształconych do postaci krawędziowej. Wyznaczono następujące typy obrazów:

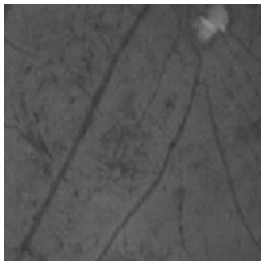
- obrazy z bardzo rzadką lub słabo widoczną siecią naczyń krwionośnych (mogą być traktowane jako tło)



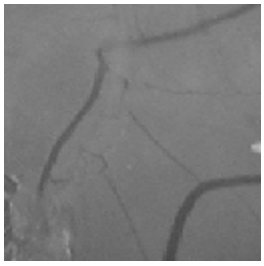
- obrazy z fragmentami grubych naczyń krwionośnych



- obrazy przedstawiające regularny układ naczyń



- obrazy przedstawiające naczynia kręte



W grupie testowej znajdowały się obrazy wszystkich typów.

8.2. Klasyfikator Bayesa

W pierwszej kolejności wykorzystano klasyfikator oparty o twierdzenie Bayesa do sprawdzenia poprawności klasyfikowania tekstur w obrazach angiograficznych na podstawie wektorów cech wyznaczonych różnymi metodami. Jest to optymalny klasyfikator cech, gdy funkcja kosztu jest zero-jedynkowa i gdy próbka ucząca ma charakter reprezentatywny dla całej dziedziny cech.

Tabela 8.1 Klasyfikator Bayes'a – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody ekstrakcji cech.

Metoda detekcji cech	Odsetek prawidłowo sklasyfikowanych obrazów	
	Obrazy angiograficzne	Obrazy angiograficzne krawędziowe
ICA	65,064 %	73,076 %
MPEG	64,615 %	60,256 %
PCA	62,820 %	71,154 %

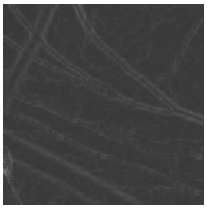
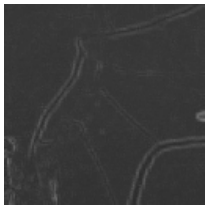
Zarówno dla zwykłych obrazów angiograficznych, jak i do obrazów przekształconych do postaci krawędziowej, wektor cech tekstury oparty o przekształcenie ICA uzyskał najlepsze wyniki. Warto zwrócić uwagę na zwiększenie skuteczności klasyfikatora w przypadku obrazów krawędziowych.

Istotną informacją jest również skuteczność klasyfikatora dla każdej klasy bazowej. Przypomnijmy, że w testach określono cztery klasy bazowe, przedstawiające różne układy naczyń krwionośnych.

Tabela 8.2 Klasyfikator Bayes'a - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy.

Klasa	Metoda	Odsetek prawidłowo sklasyfikowanych obrazów	
		Obrazy angiograficzne	Obrazy angiograficzne krawędziowe
1	ICA	90,909 %	97,727 %
	MPEG	15,909 %	18,182 %
	PCA	88,636 %	79,545 %
2	ICA	41,463 %	56,098 %
	MPEG	53,659 %	21,951 %
	PCA	26,829 %	53,659 %
3	ICA	54,717 %	37,736 %
	MPEG	71,698 %	90,566 %
	PCA	47,170 %	39,623 %
4	ICA	25,000 %	58,333 %
	MPEG	91,667 %	75,000 %
	PCA	16,667 %	41,667 %

Dla klas 2 i 4 zaobserwowano wzrost skuteczności deskryptora dla obrazów krawędziowych. Wynika to tego, że obrazy tych klas mają wyraźnie zaznaczone krawędzie naczyń, a więc po przekształceniu do postaci krawędziowej informacja nie jest tracona.

Rysunek 8.1 Przykłady obrazów krawędziowych różnych klas.

8.3. Klasyfikator NN

W przypadku klasyfikatora NN („najbliższego sąsiada") klasyfikacja obrazu przebiega na wyznaczeniu odległości (w tym przypadku Euklidesowej) do każdego z obrazów grupie obrazów bazowych i wybranie najmniejszej.

Tabela 8.3 Klasyfikator NN – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody ekstrakcji cech.

Metoda detekcji cech	**Odsetek prawidłowo sklasyfikowanych obrazów**	
	Obrazy angiograficzne	**Obrazy angiograficzne krawędziowe**
ICA	60,897 %	65,705 %
MPEG	59,295 %	64,102 %
PCA	58,230 %	62,500 %

Tabela 8.3 pokazuje skuteczność klasyfikatora najbliższego sąsiada dla wektora cech zwykłego obrazu angiograficznego, wyznaczonego za pomocą funkcji Gabora oraz z wykorzystaniem algorytmów PCA i ICA. Klasyfikator ten dał bardzo zrównoważone wyniki dla wszystkich testowanych metod. Również przekształcenie obrazów do postaci krawędziowej nie spowodowało dużego zróżnicowania wyników, jednak tak jak w poprzednim przypadku (klasyfikator Bayesa) również i klasyfikator najbliższego sąsiada lepiej radził sobie z obrazami krawędziowymi.

Tabela 8.4 przedstawia odsetek prawidłowo sklasyfikowanych obrazów pod względem podziału na klasy.

Tabela 8.4 Klasyfikator NN - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy.

Klasa	Metoda	Odsetek prawidłowo sklasyfikowanych obrazów	
		Obrazy angiograficzne	Obrazy angiograficzne krawędziowe
1	ICA	56,818 %	79,545 %
	MPEG	47,727 %	43,182 %
	PCA	52,273 %	72,727 %
2	ICA	87,805 %	14,634 %
	MPEG	90,244 %	95,122 %
	PCA	75,610 %	12,195 %
3	ICA	43,396 %	73,585 %
	MPEG	47,170 %	47,170 %
	PCA	56,604 %	71,698 %
4	ICA	41,667 %	91,667 %
	MPEG	50,000 %	16,667 %
	PCA	33,333 %	91,667 %

Można zauważyć wzrost skuteczności klasyfikatora dla obrazów klasy 1 i 4 po przekształceniu do postaci krawędziowej. Przypomnijmy, że klasa 1 zawiera obrazy mogące stanowić tło (czyli z nielicznymi lub słabo widocznymi naczyniami), natomiast klasa 4 zawiera wyraźną sieć naczyń. Po przekształceniu do postaci krawędziowej informacja o przebiegu naczyń zostaje uwypuklona (zob. Rysunek 8.1), w związku z czym różnice między obrazami tych dwóch klas stają się bardziej widoczne.

Klasyfikator najbliższego sąsiada dobrze radzi sobie również z obrazami klasy 2 (czyli zawierającymi fragmenty naczyń głównych).

8.4. Klasyfikator neuronowy

Klasyfikator neuronowy ustala przynależność obrazu testowego poprzez przetworzenie wektora cech przez sieć neuronową (omówiona w rozdziale 6.5.1). Liczba wektorów wejściowych równa się liczbie elementów wektora cech, natomiast liczba wektorów wyjściowych równa jest liczbie klas obrazów (w tym przypadku 4).

Tabela 8.5 Klasyfikator neuronowy – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody wyznaczania cech tekstury.

Metoda detekcji cech	Odsetek prawidłowo sklasyfikowanych obrazów	
	Obrazy angiograficzne	Obrazy angiograficzne krawędziowe
ICA	73,333 %	72,115 %
MPEG	70,513 %	73,718 %
PCA	64,102 %	65,705 %

Różnice w skuteczności klasyfikatora neuronowego dla różnych metod ekstrakcji cech tekstury nie są zbyt duże. Warto jednak zauważyć, że po przekształceniu obrazów testowych do postaci krawędziowej wzrasta skuteczność klasyfikatora. Najlepsze wyniki w przypadku zwykłych obrazów angiograficznych uzyskano dla wektora cech opartego o ICA, natomiast po przekształceniu do postaci krawędziowej – dla wektora cech opartego o algorytmy MPEG.

Tabela 8.6 Klasyfikator neuronowy - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy.

Klasa	Metoda	Odsetek prawidłowo sklasyfikowanych obrazów	
		Obrazy angiograficzne	Obrazy angiograficzne krawędziowe
1	ICA	40,909 %	77,273 %
	MPEG	22,727 %	72,727 %
	PCA	34,091 %	65,909 %
2	ICA	51,220 %	17,073 %
	MPEG	95,122 %	92,683 %
	PCA	46,341 %	7,317 %
3	ICA	77,358 %	96,226 %
	MPEG	75,472 %	35,849 %
	PCA	67,925 %	92,453 %
4	ICA	91,667 %	66,667 %
	MPEG	25,000 %	66,333 %
	PCA	83,333 %	64,667 %

Podobnie jak w przypadku poprzednich klasyfikatorów, również klasyfikator neuronowy okazuje się skuteczniejszy dla wybranych klas obrazów krawędziowych. Za wyjątkiem danych klasy 2 klasyfikator ten jest najskuteczniejszy przy zastosowaniu wektora cech wyznaczonego metodą ICA.

8.5. Klasyfikator SVM

W przypadku klasyfikatora SVM (maszyny wektorów nośnych) proces klasyfikacji przebiega dwustopniowo. Zastosowany został algorytm „*jedna przeciw jednej*" (opis metody zawarty jest w rozdziale 6.4). W pierwszym kroku algorytm porównuje obraz testowy do dwóch wybranych klas i przyznaje punkt tej klasie, która jest najbliżej obrazu testowego. Operacja ta jest powtarzana dla każdego obrazu dla każdego układu dwóch klas bazowych. Punkty dla każdej klasy są sumowane (odrębnie dla każdego obrazu testowego). W drugim kroku algorytm sprawdza, która klasa otrzymała największą ilość punktów – do tej klasy ostatecznie zostanie przypisany obraz.

Poniższe tabele przedstawiają wyniki uzyskane dla takiego klasyfikatora. Zastosowano dwie grupy obrazów testowych: zwykłe obrazy angiograficzne oraz te same obrazy przekształcone do postaci krawędziowej.

Tabela 8.7 Klasyfikator SVM – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody ekstrakcji cech.

Metoda detekcji cech	**Odsetek prawidłowo sklasyfikowanych obrazów**	
	Obrazy angiograficzne	**Obrazy angiograficzne krawędziowe**
ICA	76,923 %	57,692 %
MPEG	75,320 %	54,487 %
PCA	68,910 %	56,089 %

W przypadku klasyfikatora opartego o maszynę wektorów nośnych najlepsze wyniki dla zwykłych obrazów angiograficznych uzyskano dla deskryptora ICA dla zwykłych obrazów angiograficznych i dla obrazów krawędziowych.

W przypadku obrazów angiograficznych klasyfikator SVM uzyskał najlepsze wyniki spośród wszystkich omawianych klasyfikatorów. Porównując jednak z wynikami dla klasyfikatora neuronowego (Tabela 8.5) charakteryzuje się on słabą skutecznością dla obrazów krawędziowych.

Tabela 8.8 Klasyfikator SVM - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy.

Klasa	Metoda	Odsetek prawidłowo sklasyfikowanych obrazów	
		Obrazy angiograficzne	Obrazy angiograficzne krawędziowe
1	ICA	95,455 %	95,455 %
	MPEG	81,818 %	50,000 %
	PCA	47,727 %	54,545 %
2	ICA	56,098 %	63,415 %
	MPEG	29,268 %	19,512 %
	PCA	19,512 %	17,073 %
3	ICA	84,906 %	81,132 %
	MPEG	71,698 %	50,943 %
	PCA	50,943 %	67,925 %
4	ICA	75,000 %	83,333 %
	MPEG	91,667 %	75,000 %
	PCA	75,000 %	50,000 %

Klasyfikator SVM wykazał dużą skuteczność dla wszystkich omawianych metod wyznaczania cech tekstury. W przypadku klasy 1 wysoka skuteczność klasyfikatora ma znaczenie praktyczne. Można stworzyć aplikację, której zadaniem będzie wsparcie lekarza-specjalisty. Działanie aplikacji będzie polegało na odrzucaniu (oczywiście do zatwierdzenia przez specjalistę) obrazów, które nie zawierają istotnych informacji. Jednak w przypadku aplikacji mających zastosowanie w medycynie nie można całkowicie (zdaniem autorki) zrezygnować z opinii specjalisty i powinna ona stanowić jeden z etapów końcowych przetwarzania danych.

9. Zakończenie

W niniejszej pracy przedstawiono podstawowe metody analizy tekstur w obrazach cyfrowych oraz sposoby klasyfikacji. Szczególnym aspektem tej pracy było zastosowanie badanych technik do analizy obrazów angiograficznych.

Praca składa się z dwóch części. W części pierwszej przetestowane zostały metody ekstrakcji cech tekstury. Zostały wybrane trzy metody pozwalające na utworzenie skutecznych wektorów cech tekstury. Pierwsza z nich bazuje na filtracji Gabora, a wektor cech został zbudowany w oparciu o modele tzw. deskryptorów w standardzie MPEG-7. Druga metoda pozwala na utworzenie wektora cech poprzez przekształcenie PCA danych. Ostatnia metoda korzysta z przekształcenia ICA.

Praca przedstawia sposób wykorzystania procesu analizy składowych niezależnych ICA (ang. *Independent Component Analysis*) do ekstrakcji cech tekstury obrazów cyfrowych – jest to nowe podejście na gruncie polskim. Za pomocą tego algorytmu wyznaczane są wektory bazowe i odpowiadająca im przestrzeń cech. Wektory bazowe znalezione w procesie ICA są statystycznie niezależne i spełniają warunek optymalności, a przestrzeń cech jest dopasowana do danych uczących. Inną metodą, optymalizującą pewne zależne od danych kryterium, jest PCA. Z kolei omawiana filtracja Gabora, wykorzystywana w algorytmach MPEG-7, należy do metod o heurystycznie ustalonych wektorach bazowych, niezależnie od danych testowych.

Jako kryterium jakości omawianych metoda detekcji cech tekstury wykorzystano tzw. dyskryminant (informacja) Fishera. Wartość informacji Fishera wyraża stosunek rozdzielenia cech pomiędzy klasami do zwartości cech próbek należących do jednej klasy – im większa wartość, tym lepiej rozdzielone cechy różnych klas i bardziej zwarte cechy tej samej klasy.

Praca zwiera również omówienie wybranych typów klasyfikatorów. W trakcie testów sprawdzana była skuteczność stosowanych klasyfikatorów dla każdego omawianego typu wektora cech. Przeanalizowane zostały cztery typy klasyfikatorów: klasyfikator Bayesa, najbliższego sąsiada, opary o maszynę wektorów nośnych i neuronowy. Klasyfikatory te zastosowano do uzyskanych wektorów cech dla obrazów angiograficznych. Analiza wyników testów obejmowała dwa etapy. W pierwszym etapie porównano wyniki działania wybranego klasyfikatora dla wektorów cech opartych o różne algorytmy. W drugim etapie dla wybranego wektora cech, opartego o przekształcenie ICA, poszukiwano klasyfikatora o największej skuteczności.

Pierwszą tezą tej pracy było stwierdzenie, że odpowiednio dobierając reprezentację cech tekstury do rzeczywistych danych i klasyfikator, dla którego określono koszty ryzyka błędnej klasyfikacji, można stworzyć narzędzie skuteczne i pomocne w analizie obrazów. W celu udowodnienia tej tezy przetestowane zostały trzy typy wektorów cech i cztery klasyfikatory. Dla każdego wektora cech testowane były parametry przekształcenia, będącego jego podstawą. W trakcie testów wybrano najlepsze parametry, a wyniki sprawdzono posługując się metodami statystycznymi (zob. rozdział 7). Z kolei dla klasyfikatorów miarą przydatności była ich skuteczność w klasyfikowaniu obrazów (zob. rozdział 8). Wyniki testów wyraźnie wskazują, że można dobrać taką parę wektor cech - klasyfikator, by dla danej grupy obrazów uzyskać najlepsze wyniki. W przypadku ekstrakcji cech tekstury dla obrazów angiograficznych w skali szarości, dokonanej algorytmem ICA najskuteczniejszym klasyfikatorem okazał się klasyfikator oparty o maszynę wektorów nośnych.

Druga teza pracy opiera się na stwierdzeniu, że wektor cech tekstury określony poprzez przekształcenie ICA odznacza się lepszą jakością i ułatwi poprawną klasyfikację. Jakość wektora cech opartego o ICA udowodniono poprzez wyznaczenie jego funkcji jakości (dyskryminant Fisher'a – zob. rozdział 7.3.2). Przekształcenie ICA daje lepsze wyniki niż pozostałe omawiane metody. Porównując wyniki dla obrazów angiograficznych i standardowych obrazów tekstur rysunków albumu Brodatz'a również można stwierdzić

wyższość przekształcenia ICA w zastosowaniu do obrazów angiograficznych. Dla potrzeb klasyfikacji wybrano cztery różne klasyfikatory i testowano ich skuteczność dla dwóch grup obrazów angiograficznych – w skali szarości i krawędziowych. Można stwierdzić, że zastosowanie klasyfikatora neuronowego lub SVM podwyższa jakość klasyfikacji dla tekstur opisanych za pomocą wektora cech powstałego w wyniku przekształcenia ICA. Ponadto zastosowanie obrazów krawędziowych jako materiału badawczego poprawiało wyniki klasyfikacji w większości omawianych przypadków.

Tekstura jest źródłem ważnych informacji o obrazie. Analiza cyfrowych obrazów angiograficznych może być pomocnym narzędziem w diagnostyce medycznej, a omawiane algorytmy można zastosować do innych typów obrazów. Oczywiście nie można tu mówić o pełnej automatyzacji stawiania diagnozy – zbyt wiele znaczy doświadczenie lekarza, by je pomijać. Można jednak skorzystać z wyników takiej analizy do ograniczenia zbioru danych i wstępnej klasyfikacji.

Podziękowania

Autorka pracy dziękuje Panu Doktorowi Habilitowanemu Włodzimierzowi Kasprzakowi za trud promowania niniejszej rozprawy i cenne wskazówki.

Autorka chciałaby również podziękować wszystkim osobom, których pomoc przyczyniła się do powstania tej pracy.

Spis rysunków

Rysunek 1.1 Przykłady tekstur z albumu Brodatz'a (źródło: [7]) 6
Rysunek 1.2 Przykładowe obrazy angiograficzne: (a) i (b) - tkanka skórna myszy, (c) - zarodek kurzy 6
Rysunek 1.3 Sąsiedztwo punktów (pikseli) oraz kierunki analizy dla macierzy spójności jasności 11
Rysunek 2.1 Podział płaszczyzny spektogramu w przekształceniach: (a) - DFT, (b) - WT.... 19
Rysunek 2.2 Schemat rozkładu częstotliwości dla 30 kanałów. 23
Rysunek 2.3 Przykłady wzorców o różnych kształtach: (a) - oryginał, (b) - wzorzec prostokątny na czarnym tle, (c) - wzorzec prostokątny na tle wypełnionym zawartością tektury wzorca, (d) - wzorzec prostokątny na tle wypełnionym inną strukturą, (e) - wzorzec okrągły, (f) - wzorzec romboidalny. 26
Rysunek 2.4 Przykłady tekstur o różnym stopniu regularności wzorca (od lewej - zmniejszający się stopień regularności) 29
Rysunek 3.1 PCA: Składowa główna próbek w dwuwymiarowej przestrzeni cech. Linia wskazuje kierunek pierwszej składowej głównej, druga składowa główna jest prostopadła (w ogólności - ortogonalna). 34
Rysunek 3.2 Linie przedstawiają powiązania między wybranymi metodami. Opisy obok linii mówią o wymaganiach, które muszą być spełnione, by relacja była prawdziwa. 37
Rysunek 4.1 Ilustracja składania tekstury obrazu z niezależnych składowych: x_i - obserwowany i-ty blok obrazu; a_i współczynniki wymieszania źródeł (tu: wartości cech obserwowanego bloku obrazu); s_1, s_2 ,..., s_m, - źródła w ICA (ich 2-wymiarowa ilustracja) niezależne składowe (wektory bazowe) 39
Rysunek 4.2 Graficzna interpretacja procesu wybielania - po wybieleniu uzyskujemy kwadratowy rozkład próbek, wystarczy więc znalezienie jednego kąta obrotu. 42
Rysunek 4.3 Schemat działania aplikacji testowej 46
Rysunek 4.4 Na podstawie takiego wykresu można odrzucić te wartości własne, które nie przekraczają pewnego progu. Dla obrazu tekstury z albumu Brodatz'a o rozmiarach 128 × 128, dla którego wykonano ten wykres, można ograniczyć rozmiar do (na przykład) 80 próbek bez ryzyka utraty istotnych informacji. 47
Rysunek 5.1 Obraz naczyń krwionośnych w mózgu (angiogram). 48
Rysunek 5.2 Proces angiogenezy w tkance skórnej myszy. Zaznaczony obszar to miejsce wstrzyknięcia preparatu. 49
Rysunek 5.3 Proces angiogenezy w zarodku kurzym. 50
Rysunek 6.1 Przykładowy zbiór Voronoi'a. 58
Rysunek 6.2 Optymalna hiperpłaszczyzna rozdzielająca H* dla dwóch klas próbek. 61
Rysunek 6.3 Sieć neuronowa jednokierunkowa jednowarstwowa. 65

Rysunek 7.1 Przykłady tekstur z albumu Brodatz'a 70
Rysunek 7.2 Przykłady obrazów angiograficznych (zarodek kurzy) 71
Rysunek 7.3 Średnia intensywność pikseli dla deskryptora HTD dla serii 35 obrazów angiograficznych. 73
Rysunek 7.4 Przykładowe wyniki HTD dla serii 35 obrazów angiograficznych: odchylenie standardowe (czerwona linia), energia (moduł, zielona linia) oraz odchylenie standardowe energii (moduł, niebieska linia) 73
Rysunek 7.5 PCA: Przykładowe obrazy uzyskane dla określonej liczby najwyższych wartości własnych λ. 75
Rysunek 7.6 Wartość błędu dla obrazów uzyskanych w przekształceniu PCA dla pierwszych 50 największych wartości własnych. Wyznaczone dla przykładowych obrazów angiograficznych. 76
Rysunek 7.7 Wektory bazowe dekompozycji ICA, przedstawione w postaci obrazów dla próbki uczącej (obrazy angiograficzne). 77
Rysunek 7.8 Wektory bazowe dekompozycji ICA, przedstawione w postaci obrazów, dla wybranych przykładów tekstur z albumu Brodatz'a 77
Rysunek 7.9 ICA: Przykładowe obrazy uzyskane dla ograniczonej liczby najwyższych składowych niezależnych. 78
Rysunek 7.10 ICA: wykres wartości współczynników wymieszania dla obrazu klasy A. 79
Rysunek 7.11 ICA: wykres wartości współczynników wymieszania dla obrazu klasy B. 79
Rysunek 7.12 ICA: wykres wartości współczynników wymieszania dla obrazu klasy C. 79
Rysunek 7.13 ICA: wykres wartości współczynników wymieszania dla obrazu klasy D. 79
Rysunek 7.14 Przykładowy obraz angiograficzny i odpowiadający mu obraz krawędziowy. 80
Rysunek 7.15 Dyskryminant Fishera: przebieg wartości funkcji jakości dla pierwszych 62 elementów wektorów cech wyznaczonych różnymi metodami dla obrazów angiograficznych. 85
Rysunek 7.16 Dyskryminant Fishera: przebieg wartości funkcji jakości dla 150 współczynników wektora cech wyznaczonego algorytmem ICA dla obrazów angiograficznych. 86
Rysunek 7.17 Dyskryminant Fishera: przebieg wartości funkcji jakości dla 4096 współczynników wektora cech wyznaczonego algorytmem PCA dla obrazów angiograficznych. 87
Rysunek 8.1 Przykłady obrazów krawędziowych różnych klas. 91

Spis tabel

Tabela 2.1 Przykładowe wyniki testów dla wzorców HTD o różnym kształcie (wskaźnik ARR) 26
Tabela 7.1 Średnie wyniki dla deskryptora tekstury jednorodnej. Oznaczenia: μ – średnia intensywność pikseli, σ - odchylenie standardowe, |e| - energia (moduł), $|\sigma_e|$ - odchylenie standardowe energii (moduł) 72
Tabela 7.2 Dyskryminant Fishera: wartość funkcji jakości dla wektorów cech (ograniczonych do pierwszych 62 elementów) dla obrazów angiograficznych. 84
Tabela 7.3 Dyskryminant Fishera: wartość funkcji jakości dla wektorów cech (ograniczonych do pierwszych 62 elementów) dla obrazów tekstur z albumu Brodatza. 85
Tabela 8.1 Klasyfikator Bayes'a – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody ekstrakcji cech. 90
Tabela 8.2 Klasyfikator Bayes'a - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy. 90
Tabela 8.3 Klasyfikator NN – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody ekstrakcji cech. 91
Tabela 8.4 Klasyfikator NN - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy. 92
Tabela 8.5 Klasyfikator neuronowy – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody wyznaczania cech tekstury. 93
Tabela 8.6 Klasyfikator neuronowy - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy. 93
Tabela 8.7 Klasyfikator SVM – odsetek prawidłowo sklasyfikowanych obrazów w zależności od zastosowanej metody ekstrakcji cech. 94
Tabela 8.8 Klasyfikator SVM - odsetek prawidłowo sklasyfikowanych obrazów pod względem przynależności do klasy. 95

Bibliografia

[1] (-), *Core experiment description for binary and dynamic MPEG-7 data representation*, ISO/IEC JTC1/SC29/WG11/N3577, Beijing, lipiec 2000.

[2] Amari S., *Neural learning in structured parameter spaces – natural Riemannian gradient*. Advances in Neural Information Processing 9 NIPS'96, pp. 127-133, MIT Press, Cambridge 1997.

[3] Amari S., Cichocki A., Yang H.H., *A new learning algorithm for blind source separation*. Advances in Neural Information Processing 8 NIPS'95, pp. 757-763, MIT Press, Cambridge 1996.

[4] Bałan B. J., Skopińska-Różewska E., Snitkowska E., Sommer E., Domański J., Skurzak H. *Ocena aktywności anty-angiogennej wieloskładnikowego preparatu ziołowego „Pervivo" z zastosowaniem komputerowej analizy obrazów*, Terapia, rok IX, Nr 3, z. 2/105, marzec 2001.

[5] Bell A.J., Sejnowski T.J., *An information–maximization approach to blind separation and blind decolvolution*. Neural Computation, 7, pp. 1129-1159, 1995.

[6] Bell A.J., Sejnowski T.J., *Edges are the independent components of natural scenes*. Advances in Neural Information Processing 9 NIPS'96, pp. 831-837, MIT Press, Cambridge 1997.

[7] Brodatz P., *Textures. A photographic album for artists and designers*. Dover Publ., New York, 1966.

[8] Burges C.J.C.: *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, Vol. 2, No. 2, pp. 121-167, 1998.

[9] Cardoso J.-F., Comon P., *Independent component analysis, a survey of some algebraic methods*. Proc. ISCAS'96, Vol. 2, pp. 93-96, 1996.

[10] Cardoso J.-F., Laheld B.H., *Equivariant adaptive source separation*. IEEE Trans. On Signal Processing, 44(12), pp. 3017-3030, 1996.

[11] Chen Y, Dougherty E. R., *Gray-scale morphological granulometric texture classification*. Optical Engineering, Vol. 33, No. 8, pp. 2713-2722, August 1994.

[12] Cichy P., *Analiza tekstury obrazów cyfrowych – zastosowanie do wybranej klasy obrazów biomedycznych*. Rozprawa doktorska, Politechnika Łódzka, Wydział Elektrotechniki i Elektroniki, Instytut Elektroniki, Łódź 2001.

[13] Cichocki A., Amari S., *Adaptive Blind Signal and Image Processing*. John Wiley, Chichester, UK, 2002.

[14] Cichocki A., Amari S.I., Thawonmas R., *Blind signal extraction using self-adaptive non-linear hebbian learning rule*. Proc. NOLTA'96, pp. 377-380, 1996.

[15] Cichocki A., Kasprzak W., *Nonlinear learning algorithms for blind separation of natural images*. Neural Network World, 4, pp. 515-523, 1996.

[16] Cichocki A., Kasprzak W., Skarbek W., *Adaptive learning algorithm for principal component analysis with partial data*. Cybernetics and Systems'96, Austrian Society for Cybernetics Studies, Vienna, 1996, pp. 1014-1019.

[17] Cichocki A., Unbehauen R., Moszczynski L., Rummert E., *A new on-line adaptive algorithm for blind separation of source signals*. Proc. Int. Symposium on Artifical Neural Networks ISANN-94, pp. 406-411, Tainan, Taiwan 1994.

[18] Cieplinski L., Munchurl K. Ohm J.-R. Pickering M., Yamada A.,: *Study of CD 15938-3 MPEG-7 Multimedia Content Description Interface – Part 3 Visual*. ISO/IEC JTC1/SC29/WG11/M6809 Pisa, styczeń 2001.

[19] Comon P., *Independent component analysis – a new concept?*.Signal Processing, 36, pp.287-314, 1994.

[20] Cordón O., del Jesus M. J., Herrera F., *Genetic Learning Of Fuzzy Rule-Based Classification Systems Cooperating With Fuzzy Reasoning Methods*. Technical Report #DECSAI-970130, Department of Computer Science and Artificial Intelligence, Granada University, Nov. 1997.

[21] Cortes C., Vapnik V., *Support-Vector Networks*. Machine Learning, Vol. 20, No. 3, pp. 273-297, 1995.

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

[23] Faugeras O. D., Pratt W. K., *Decorrelation Methods of Texture Feature Extraction*. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 4, July 1980, pp. 323-332.

[24] de Figueiredo M.T., Leitão J.M.N., *Bayesian Estimation of Ventricular Contours in Angiographic Images*. IEEE Transactions on Medical Imaging, Vol. 11, No. 3, pp. 416-429, September 1992.

[25] Friedman J.H., *Exploratory projection pursuit*. J. of the American Statistical Association, 82(397), pp. 249-266, 1987.

[26] Galiński G., Skarbek W., *Object Tracing Using SVD and Oja Algorithm*. Proceedings of the 11th Portuguese Conference on Pattern Recognition, RECPAD 2000, Porto, May 2000, pp.219-226.

[27] Giannakopoulos X., Karhunen J., Oja E., *An Experimental Comparison Of Neural ICA Algorithms*. Proc. Int. Conf. on Artificial Neural Networks ICANN'98, pp. 651-656, Sweden, 1998.

[28] Giardina C. R., Daugherty E. R., *Morphological Methods in Image and Signal Processing*. Prentice-Hall Inc., Englewood Cliffs, New Jersey 1988.

[29] Haralick R. M., Shanmugam K., Dinstein I., *Textural Features For Image Classification*. IEEE Trans. On Systems, Man and Cybernetics, Vol. SMC-3, No. 6., pp. 610-621, November 1973.

[30] Harman H.H., *Modern Factor Analysis*. University of Chicago Press, 2nd Edition, 1967.

[31] Hoyer P. O., Hyvärinen A., *Independent Component Analysis Applied To Feature Extraction From Colour And Stereo Images*. Computation In Neural Systems 11, pp. 191-210, August 2000.

[32] Huber P.J., *Projection pursuit*, The Annals of Statistics, 13(2), pp. 435-475, 1985.

[33] Hyvärinen A., *A Family Of Fixed-Point Algorithms For Independent Component Analysis*. Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing ICASSP'97, pp. 3917-3920, Munich, Germany 1997.

[34] Hyvärinen A., *Independent Component Analysis By Minimization Of Mutual Information*. Report A46, Helsinki University Of Technology, Department Of Computer Science And Engineering, Laboratory Of Computer And Information Science, 1997.

[35] Hyvärinen A., *Survey On Independent Component Analysis*. Neural Computing Surveys, 2, pp.94-128, 1999.

[36] Hyvärinen A., *Independent Component Analysis: A neural Network Approach*., Ph.D. Thesis, Helsinki University Of Technology, Finland, 1997.

[37] Hyvarinen A., Karhunen J., Oja E., *Independent Component Analysis*. John Wiley & Sons, New York etc., 2001.

[38] Hyvärinen A., Oja E., *One-unit Learning Rules For Independent Component Analysis*. Advances in Neural Information Processing Systems 9 (proc. NIPS'96), pp. 480-486, MIT Press, 1997.

[39] Hyvärinen A., Oja E., *A Fast Fixed-Point Algorithm For Independent Component Analysis*. Neural Computation, 9(7), pp. 1483-1492, 1997.

[40] Hyvärinen A., Oja E., *Simple Neuron Models For Independent Component Analysis*. Int. J. Of Neural Systems, 7(6), pp. 671-687, 1995.

[41] Hyvärinen A., Pajunen P., *Nonlinear Independent Component Analysis: Existence And Uniqueness Results*. Neural Networks, 12(3), pp. 429-439, 1999.

[42] Jain A. K., Farrokhnia F., *Unsupervised texture segmentation using Gabor Filters*. Pattern Recognition, vol. 24 (1991), No. 12, pp. 1167-1186.

[43] Jenssen R., Eltoft T., *ICA Filter Bank For Segmentation Of Textured Images*. Proc. Int. Workshop on Independent Component Analysis and Blind Source Separation ICA2003, pp. 827-832, Nara, Japan, April 2003.

[44] Jutten C., Herault J., *Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture*. Signal Processing, 24, pp.1-10, 1991.

[45] Karklin Y., Lewicki M.S., *Learning higher-order structure in natural images*. Network: Computation in Neural Systems, 14, pp. 483-499, 2003.

[46] Kasprzak W., *Adaptive Computation Methods In Digital Image Sequence Analysis*. Prace Naukowe Elektronika Z. 127, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000.

[47] Kasprzak W. *Rozpoznawanie obrazów i sygnałów mowy*. Materiały wykładowe.

[48] Kasprzak W., Cichocki A., *Hidden Image Separation From Incomplete Image Mixtures By Independent Component Analysis*. 13th. Int. Conf. On Pattern Recognition ICPR'96, Proceedings Vol. II, pp. 394-398.

[49] Kasprzak W., Cichocki A., Amari S., *Blind Source Separation with Convolutive Noice Cancellation*. Neural Computing and Applications, Springer, London, vol. 6 (1997), pp. 127-141.

[50] Kassyk-Rokicka H., *Mierniki statystyczne*. Państwowe Wydawnictwo Ekonomiczne, Warszawa 1986.

[51] Kendall M., *Multivariate Analysis*. Charles Griffin & Co., 1975.

[52] Klukowski J., *Algebra w zadaniach*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1995.

[53] Krysicki W., Bartos J., Dyczka W., Królikowska K., Wasilewski M., *Elementy probabilistyki w zadaniach*. Wydawnictwo Politechniki Łódzkiej, Łódź 1986.

[54] Kulikowski J.L., *Cybernetyczne układy rozpoznające*. PWN, Warszawa 1972.

[55] Lappalainen H., *Ensemble Learning For Independent Component Analysis*. Proc. 1st Int. Workshop on Independent Component Analysis and Signal Separation, Aussois, France, pp. 7-12, 1999.

[56] Lee T.W. Lewicki M.S., *Unsupervised Imane Classification, Segmentation and Enhancement Using ICA Mixture Models*. IEEE Transactions on Image Processing, Vol. 11, No. 3, pp.270-279, March 2002.

[57] Lygeros J., *A Formal Approach to Fuzzy Modelling*. Proceedings of ACC, pp. 3740-3744, 1995.

[58] Manjunath B.S., Shin H., *Texture core experiment results: browsing*. ISO/IEC JTC1/WG11/M5047, Melbourne, October 1999.

[59] Manjunath B.S., Shin H.: *Texture core experiment results: similarity retrieval*. ISO/IEC JTC1/SC29/WG11/M5092, Melbourne, October 1999.

[60] Manjunath, B.S., Wo, P., Newsam, S., Shin, H.D., *A texture descriptor for browsing and similarity retrieval*. Journal of Signal Processing: Image Communication, **16**, Issue 1-2, pp. 33-43, September 2000.

[61] Manjunath B.S., Wu P., Newsam S.D., Shin H., Choi Y., *MPEG-7 Core Experiment: Homogeneous Texture Descriptor (P77)*. ISO/IEC JTC1/SC29/WG11/M4721, Vancouver, July 1999.

[62] Materka A., *Elementy cyfrowego przetwarzania i analizy obrazów*. Państwowe Wydawnictwo Naukowe, Warszawa 1991.

[63] Materka A., *Metody diagnostyki obrazowej w medycynie i technice*. II Krajowa Konferencja Elektroniki, pp. 23-28, Kołobrzeg 2003.

[64] Matheron G., *Random sets and integraf geometry*. Wiley, New York 1975.

[65] Nawa N. E., Hashiyama T., Furuhashi T., Uchikawa Y., *A Study on Fuzzy Rules Discovering Using Pseudo-Bacterial Genethic Algorithm with Adaptive Operator*. Proceedings of IEEE International Conference on Evolutionary Computation, ICEC 1997.

[66] Niemann H., *Pattern Analysis and Understanding*. Springer, Berlin etc., 1990.

[67] Niemann H., *Klassifikation von Mustern*. Springer, Berlin, 2003, (2nd edition).

[68] Nieniewski M., Serneels R., *Extraction of the Shape of Small Defects on the Surface of Ferrite Cores*. Machine Graphics and Vision, vol. 9, No 1/2, pp. 453-462, 2000.

[69] O'Brien J.F., Ezquerra N.F., *Automated Segmentation of Coronary Vessels in Angiographic Image Sequences Utilizing Temporal, Spatial and Structural Constraints*. Proc. of the 3rd Conference on Visualization in Biomedical Computing SPIE, 1994.

[70] Øien G.E., Osnes P., *Diabetic retinopathy: Automatic detection of early symptoms from retinal images*. Proc. NORSIG'95, Norvegian Signal Processing Symposium, pp. 135-140, Stavanger, Norway 1995.

[71] Oja E., Karhunen J., Hyvärinen A., *From neural principal components to neural independent component*. Proc. Int. Conf., on Artificial Neural Networks, Lausanne, Switzerland, 1997.

[72] Oja E., Karhunen J., Hyvärinen A., Vigario R., Hurri J., *Neural independent component analysis – approaches and applications*. Proc. Int. Conf. on Neural Information Processing, Dunedin, New Zealand, 1997.

[73] Osowski S., *Sieci neuronowe w ujęciu algorytmicznym*. WNT Warszawa 1996.

[74] Park N., Prasanna V.K., *Cache Conscious Walsh-Hadamard Transform*. Proc. ICASSP, Vol. 2, pp. 1205-1208, 2001.

[75] Randen T., Husøy J. H., *Filtering For Texture Classification: A Comparative Study*. IEEE Trans. On Pattern Analysis And Machine Intelligence, vol. 21, no. 4, pp.291-310, April 1999.

[76] Ro Y.M., Yoo K., Kim M., Kim J., *Preliminary experimental results on boundary shape effects in homogeneous texture descriptor*. ISO/IEC JTC1/SC29/WG11/M5449, Maui, December 1999.

[77] Ro Y.M., You K.W., Kim M., Kim J., *Texture description using Radon transform*. ISO/IEC JTC1/SC29/WG11/M4703, Vancouver 1999.

[78] Ro Y.M., You K.W., Kim M., Kim J., *Texture description using Radon transform and experimental results on CT-5 core experiment*. ISO/IEC JTC1/SC29/WG11/M5152, Vancouver 1999.

[79] Ro Y.M. (ed.), *A unified texture descriptor*. ISO/IEC JTC1/SC29/WG11/M5490, Maui, grudzień 1999.

[80] Russel S., Norvig P., *Artificial Intelligence: A Modern Approach*. Person Education, 2nd Ed., 2003

[81] Scheunders P., Livens S., Van de Wouwer G., Vautrot P., Van Dyck D., *Wavelet-based Texture Analysis*. Journal on Computer Science And Information Management, Vol. 1(1998), Nr 2, pp. 22-34.

[82] Schurmann J., *Pattern classification. A unified view of statistical and neural approaches*. John Wiley & Sons, New York 1996.

[83] Sikora R., Skarbek W.: *On Stability of Oja Algorithm in Rough Sets and Current Trends in Computing*. (eds.: L. Polkowski, A. Skowron) Lecture Notes in Computer Science 1424, pp. 354-360, 1998.

[84] Singer B., Veloso M.M., *Automating the Modeling and Optimization of the Performance of Signal Transforms*. IEEE Trans. on Signal Processing, 50(8), 2003.

[85] Skarbek W., *Metody reprezentacji obrazów cyfrowych*. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993.

[86] Skarbek W. (ed.), *Multimedia – Algorytmy i standardy kompresji*. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998.

[87] Skarbek W., *Local Principal Component Analysis in Transform Coding*. International Symposium on Nonlinear Theory and Its Applications, NOLTA 1996.

[88] Skarbek S., *Rough sets for enhancements of local subspace classifier*. Neurocomputing, Vol. 36, pp. 67-83, 2001.

[89] Skarbek W., Sikora R., *Stability Analysis of Oja-RLS Learning Rule*. Fundamenta Informaticae 34, pp.1-13, 1998, IOS Press.

[90] Smith J.R., Chang S.F., *Automated Binary Texture Feature Sets for Image Retrieval*. Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP'96), Atlanta, May 1996.

[91] Snitkowska E., *Deskryptory tekstury w standardzie MPEG-7*. VIII Sympozjum Nowości w Technice Audio i Wideo, Warszawa 2001.

[92] Snitkowska E., *Deskryptory tekstury Snitkowska analizie obrazów angiograficznych*. IX Sympozjum Nowości w Technice Audio i Wideo, Warszawa 2002.

[93] Snitkowska E., Bałan B. J., Sommer E., Domanski J., Skopińska-Różewska E. *Program komputerowej analizy obrazów MOUSE OBSERVER i jego zastosowanie do oceny wyników uzyskanych w teście skórnej angiogenezy u myszy*. Terapia, rok IX, Nr 3, z. 2/105, marzec 2001.

[94] Snitkowska E., Kasprzak W., *Procedura analizy składowych niezależnych i jej wykorzystanie do opisu tekstur w obrazach cyfrowych*. VIII Krajowa Konferencja Robotyki, KKR 2004, Polanica Zdrój, czerwiec 2004) (w druku).

[95] Snitkowska E., Kasprzak W., *Independent Komponent Analysis of Textures In Angiography Images*. ICCVG 2004, Warszawa, wrzesień 2004 (w druku)

[96] Strzelecki M., *Segmentacja tekstur obrazów biomedycznych z wykorzystaniem wielowarstwowych sieci percepronowych*. Elektronika, Prace Naukowe, Zeszyt 2, Łódź 1997, str. 115-135.

[97] Sumengen B., Newsam S., Manjunath B.S., Shin H., Choi Y., *Texture core experiments results: similarity retrieval*. ISO/IEC JTC1/SC29/WG11/M5563, Maui, December 1999.

[98] Sun J., *Some practical aspects of exploratory projection pursuit*. SIAM J. of. Sci. Comput., 14, pp. 68-80, 1993.

[99] Swets D.L., Weng J.J., *Using discriminant eigenfeatures for image retrieval*. IEEE Trans. Patt. Anal., Mach. Intell., PAMI-18 (1996), No. 8, pp. 831-836.

[100] Szabatin J., *Podstawy teorii sygnałów, Wydawnictwa Komunikacji i Łączności*,.Warszawa 1990.

[101] Tadeusiewicz R., *Problemy biocybernetyki*. Państwowe Wydawnictwo Naukowe, Warszawa 1991.

[102] Tadeusiewicz R., *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, Warszawa 1993.

[103] Tadeusiewicz R., Flasiński M., *Rozpoznawanie obrazów*. PWN, Warszawa 1991.

[104] Tadeusiewicz R., Korohoda P., *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków, 1997.

[105] Tamura H., Mori S., Yamawaki T., *Textural Features Corresponding To Visual Perception*. IEEE Trans. On Systems, Man And Cybernetics, Vol. SMC-8, No. 6, pp. 460-473, June 1978.

[106] Turk M., Pentland A., *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, vol. 3 (1991), pp. 71-86.

[107] Turner M. R., *Texture Discrimination by Gabor Functions*. Biological Cybernetics, vol. 55 (1986), pp. 71-82.

[108] Unser M., Eden M., *Nonlinear Operators for Improving Texture Segmentation Based on Features Extracted by Spatial Filtering*. IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 4, July/August 1991, pp. 804-815.

[109] Vapnik V., *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[110] Vapnik V., *Statistical Learning Theory*. John Wiley and Sons Inc., New York 1998.

[111] Wang L., Bhalerao A., *Detecting Branching Structures using Local Gaussian Model*. Proc. of IEEE International Symposium on Biomedical Imaging ISBI, Washington DC, July 2002.

[112] Watkins D., Sadun A., Marenka S., *Nowoczesne metody przetwarzania obrazu*. WNT, Warszawa 1995.

[113] Weingessel A., Natter M., Hornik K., *Using Independent Component Analysis for Feature Extraction and Multivariate Data Projection*. SFB Adaptive Information Systems And Modelling In Economics And Management Science, Working Paper No. 16, Vienna 1998.

[114] Weldon T., Higgins W., Dunn D., *Efficient Gabor Filter Design For Texture Segmentation*. Pattern Recognition, Vol. 29(12), pp. 2005-2015, 1996.

[115] Wu P., Manjunath B.S., Shin H., *Texture core experiment results: browsing*. ISO/IEC JTC1/SC29/WG11/M5499, Maui, December 1999.

[116] Yamada A., Pickering M., Jeannin S., Cieplinski L., Ohm J.R., Munchurl K.: *MPEG-7 Visual part of eXperimentation Model Version 8.0.* ISO/IEC JTC1/SC29/WG11/N3673, La Baule, październik 2000.